

# Dual Temporal Memory Network for Efficient Video Object Segmentation

Kaihua Zhang<sup>1</sup>, Long Wang<sup>1</sup>, Dong Liu<sup>2</sup>, Bo Liu<sup>3</sup>, Qingshan Liu<sup>1</sup>, Zhu Li<sup>4</sup>

<sup>1</sup>B-DAT and CICAET, Nanjing University of Information Science and Technology, Nanjing, China

<sup>2</sup>Netflix Inc.

<sup>3</sup>JD Digits, Mountain View, CA, USA

<sup>4</sup> Dept. of CSEE, University of Missouri-Kansas City, Missouri 64110, USA.

{zhkhua, kfliubo}@gmail.com

## Abstract

Video Object Segmentation (VOS) is typically formulated in a semi-supervised setting. Given the ground-truth segmentation mask on the first frame, the task of VOS is to track and segment the single or multiple objects of interests in the rest frames of the video at the pixel level. One of the fundamental challenges in VOS is how to make the most use of the temporal information to boost the performance. We present an end-to-end network which stores short- and long-term video sequence information preceding the current frame as the temporal memories to address the temporal modeling in VOS. Our network consists of two temporal sub-networks including a short-term memory sub-network and a long-term memory sub-network. The short-term memory sub-network models the fine-grained spatial-temporal interactions between local regions across neighboring frames in video via a graph-based learning framework, which can well preserve the visual consistency of local regions over time. The long-term memory sub-network models the long-range evolution of object via a Simplified-Gated Recurrent Unit (S-GRU), making the segmentation be robust against occlusions and drift errors. In our experiments, we show that our proposed method achieves a favorable and competitive performance on three frequently-used VOS datasets, including DAVIS 2016, DAVIS 2017 and Youtube-VOS in terms of both speed and accuracy.

## 1. Introduction

Video Object Segmentation (VOS) aims to separate the foreground objects from the backgrounds in all frames of a video sequence. The common approach casts the problem into a semi-supervised learning task, *i.e.*, the segmentation ground truth of the target object in the first frame is provided and the goal is to infer the segmentation masks of the object in all other frames [2, 40, 33, 49, 50, 45, 39, 38, 19]. Fast

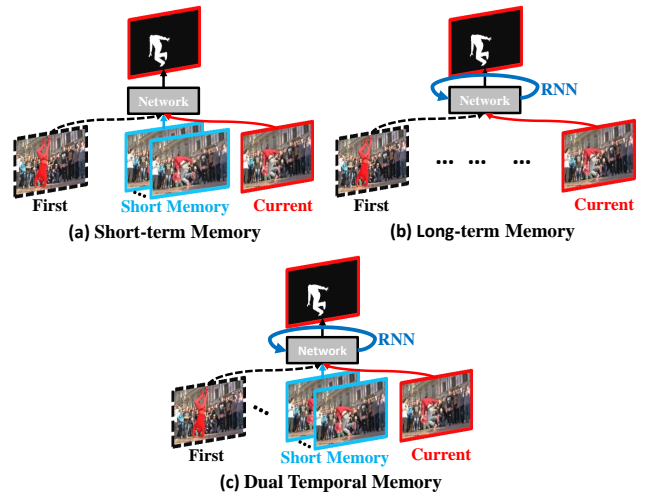


Figure 1. Previous methods [33, 21, 5, 38, 47, 37] capture temporal dependencies in a short-term or long-term video sequence for VOS (a,b). Our proposed method leverages both short- and long-term temporal information (c). RNN is short for Recurrent Neural Network.

and accurate VOS methods are beneficial to many applications such as video editing [26, 42], object tracking [41, 51] and activity recognition [9].

Modelling the inter-frame temporal correlation is one of the essential challenges in VOS. Some existing methods [33, 21, 5] model the *short-term* consistency of the object appearance across neighboring frames in the video (Figure 1 (a)). The predicted mask of the previous frame is propagated to the current frame either by feature map aggregation or by optical-flow-guided pixel matching. The major issue with these methods is that they ignore the fine-grained interactions between local regions over time. In other words, the spatial regions in the predicted mask of the previous frames are integrated into the corresponding spatial regions in the current frame *individually* without ex-

ploring their spatial-temporal correlations. As a result, local prediction errors may easily be propagated and amplified during temporal modeling, especially at the border of object regions. Ideally, we need a mechanism to model the fine-grained spatial-temporal interactions on the local frame regions, so that the consistency of object can be preserved.

Other works [38, 47, 37] apply *Convolutional Recurrent Unit* to capture the evolution of the frame’s convolutional feature map over a *long-term time range*, and map the output of the recurrent units into a segmentation map of the current frame (Figure 1 (b)). These methods can get a long view of the video sequence preceding the current frame so that the long-range dynamics of the object can be captured, making the network be robust against occlusions and drift errors. Nevertheless, the major issue with these methods is that the feature maps fed into the recurrent units describe the holistic frame, which not only unnecessarily involves the background region into the learning process, but also dramatically increases the computational complexity. In fact, only the object mask regions are needed to model the evolution of object in time.

Motivated by the above issues in VOS, we propose an end-to-end *Dual Temporal Memory Network* (DTMNet) that stores both short-term and long-term video sequence information as memories to assist the segmentation of a current frame (Figure 1 (c)). In our network, the *short-term memory sub-network* is designed as a spatial-temporal feature correlation module to capture the fine-grained inter-frame object appearance consistency. Given a current frame, we collect a small window of the preceding frames as its short-term memory. The frame and its memory frames are respectively encoded into a feature map in which each spatial location denotes one local region in the frame and the same feature location across different frames naturally encodes the evolution of a region across time. Then a spatial-temporal graph is built over all local regions in which each region is a node and the edges are established between regions within a local spatial-temporal window. The *Graph Convolution* [18] operation is performed to update each region feature on the node according to its relations to others. By doing this, we model the spatial-temporal consistency of local regions across frames, leading to an improved segmentation performance.

The *long-term sub-network* models the evolution of object across a long-time range. Given a current frame, we collect all preceding frames from the beginning of the video as its long-term memory. Instead of using the convolutional features of frames in the memory to model the dynamics of object over time, we propose to pool an object-orientated feature vector from the object mask on each frame, and apply the *Simplified-Gated Recurrent Unit* (S-GRU) to learn a hidden-state vector to characterize the evolution of the object over a long-time range in the memory. This relieves

the distractions of the background regions and significantly reduces the computational complexity.

The outputs from the short-term and the long-term sub-networks are sent to the *segmentation sub-network* as supportive information to perform object segmentation. Extensive evaluations on three benchmark VOS datasets demonstrate that our DTMNet yields state-of-the-art performance in terms of both speed and accuracy. Our main contributions include:

(1) DTMNet for VOS, through which both the short-term spatial-temporal local region consistency and the long-term object evolution can be exploited.

(2) A graph-based learning framework to model the short-term spatial-temporal interactions of the local regions from neighboring frames in the video.

(3) An object-orientated feature based S-GRU module to model object evolution over a long-time range.

## 2. Related Work

**Video Object Segmentation.** There is a line of research on unsupervised VOS which leverages visual saliency [14, 17], point trajectory [3] and motion [32] to segment objects from the background. Many semi-supervised VOS methods heavily rely on online fine-tuning on the first-frame mask to predict the masks on other frames during testing. OSVOS [2] and its extensions [40] ignore the temporal dimension and fine-tune a pre-trained fully convolutional network on the first frame to remember the object appearance. MHP-VOS [49] proposes a novel method called Multiple Hypotheses Propagation to defer the decision until a global view can be established. Other methods take *temporal information* into consideration. MSK [33] and LucidTracker [21] use the predicted mask of the last frame as additional input of the current frame. PReMVOS [28] combines four different sub-networks to achieve impressive performance. Although online fine-tuning boosts test accuracy, it badly sacrifices running efficiency for practical applications.

A growing line of research attempts to avoid the time-consuming online fine-tuning at the expense of a little accuracy reduction. VideoMatch [15] explores pixel-level embedding matching. OSMN [50] uses two novel modulators to capture visual and spatial information of the target object and injects them into the segmentation branch. FAVOS [4] utilizes tracking to obtain object bounding boxes and performs segmentation within the boxes. AGAM-VOS [19] learns a probabilistic generative model to find a representation of the target and background appearance. Our method shares the same spirit of not performing online fine-tuning as these methods, but the key difference is that we design a dedicated dual temporal memory mechanism to make the VOS accuracy even higher than some state-of-the-art online fine-tuning methods (see Table 1).

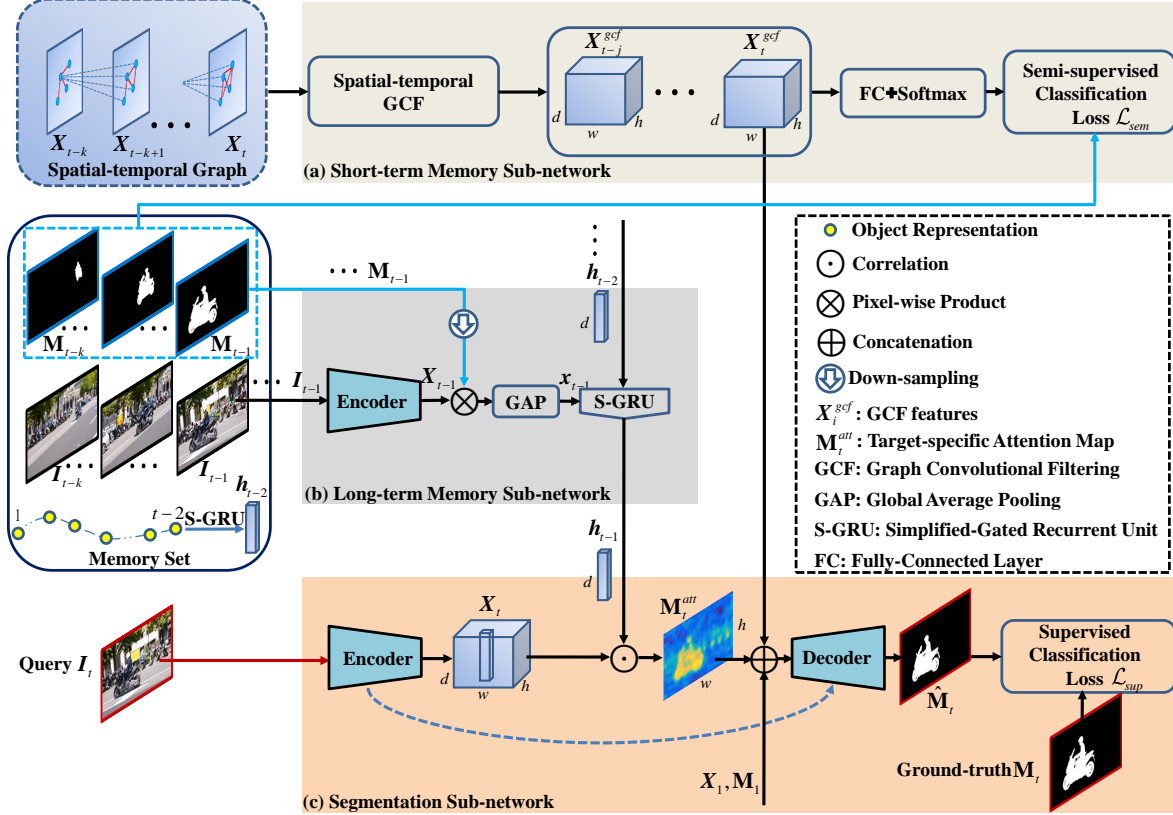


Figure 2. Pipeline of the proposed DTMNet for VOS. The network includes three key components: (a) A short-term memory sub-network to capture the spatial-temporal consistency of local regions over time; (b) A long-term memory sub-network to model the evolution of object over a long-time range to ensure robustness against occlusions and drift errors; (c) A segmentation sub-network that seamlessly fuses the short- and long-term memory information and the ground-truth information from the first frame to accurately predict the segmentation mask.

**Temporal Modeling in VOS.** Temporal sequence modeling plays an important role in VOS. Some methods try to model the *long-term* object dynamics in the video sequence using Recurrent Neural Networks (RNNs). RNN is designed to sequence modeling by propagating and accumulating a hidden state over time [8, 20]. Gated Recurrent Unit (GRU) [6] and Long Short-Term Memory (LSTM) [12] are the two classic RNN components, both of which have been extended to CNNs. The existing methods introduce ConvLSTM and ConvGRU to model the long-term temporal dynamics in the video sequence. [37] designs a visual memory module based ConvGRU to capture the evolution of object mask over times. RVOS [38] presents a spatial-temporal recurrence module and applies ConvLSTM as the decoder. [47] proposes a sequence-to-sequence network by ConvLSTM. Without any RNN unit, STCNN [46] designs a novel temporal coherence branch inspired by video predict task, and is also able to model the long-term dynamics in video. STMN [31] stores the first, intermediate and previous frame in the memory and

uses them as reference to infer the object mask of the current frame. The advantage of modeling long-term temporal dynamics is to allow the network to get the long view of the video sequence preceding the current frame, making the network be robust against occlusions and drift errors.

The other VOS methods model the short-term visual consistency across neighboring frames in the video. MSK [33] uses the predicted mask of the last frame as additional input of the current frame to help with the mask prediction. RGMP [45] utilizes a Siamese encoder-decoder network to extract the first, previous and current feature to propagate the previous predicted mask to current frame. Optical flow is also commonly used to match the pixel correspondence between successive frames, through which the object mask of sequential frames can be estimated [21, 5, 28]. It turns out that modeling short-term visual consistency is an important prior to enhance the performance of VOS, and is commonly applied in the VOS works [33, 45, 21, 5, 28]. In contrast to these existing methods, we integrate the long- and short-term temporal model-

ing into a unified framework, and each sub-network in our network design is dedicated to resolve the issues of the existing methods.

### 3. Proposed Method

#### 3.1. Framework Overview

Given a video sequence with  $t$  frames  $\mathcal{I}_1^t = \{\mathbf{I}_i\}_{i=1}^t$  and the binary ground-truth mask  $\mathbf{M}_1 \in \{0, 1\}^{w \times h}$  of the first frame  $\mathbf{I}_1$  with width  $w$  and height  $h$ , our task is to predict the segmentation masks of the subsequent frames  $\mathcal{I}_2^t$ , denoted as  $\mathcal{M}_2^t = \{\mathbf{M}_i\}_{i=2}^t$ . To this end, we develop the DTMNet for VOS, as illustrated by Figure 2. Our DTMNet is composed of three seamless components: (a) A short-term memory sub-network; (b) A long-term memory sub-network and (c) A segmentation sub-network. Among them, the segmentation module takes full advantages of the complementary characteristics of the rich supportive information provided by the short- and the long-term memory modules, *i.e.*, good adaptation to appearance changes and robustness against occlusions and drift errors, thereby enabling to predict an accurate segmentation mask.

Specifically, when segmenting video frame  $\mathbf{I}_t$ , we take it as the query frame and the preceding  $k$  frames  $\mathcal{I}_{t-k}^{t-1}$  with their masks  $\mathcal{M}_{t-k}^{t-1}$  as the short-term memories. The frames  $\mathcal{I}_{t-k}^t$  are fed into the backbone network as the encoder to extract features  $\mathcal{X}_{t-k}^t = \{\mathbf{X}_i\}_{i=t-k}^t$ , where the feature map  $\mathbf{X}_i \in \mathbb{R}^{w \times h \times d}$  with  $d$  channels. Afterwards, as shown in Figure 2(a), the features  $\mathcal{X}_{t-k}^t$  are fed into a spatial-temporal graph convolutional filtering (GCF) module, generating the refined features  $\mathbf{X}_t^{gcf} \in \mathbb{R}^{w \times h \times d}$  for the query frame  $\mathbf{I}_t$ . The GCF leverages Laplacian smoothing to compute  $\mathbf{X}_t^{gcf}$  that can be viewed as a low-pass filtering process [24]. The smoothing makes the features in the same cluster similar, facilitating the subsequent classification task in the segmentation sub-network.

Meanwhile, as shown by Figure 2(b), we leverage S-GRU to model the long-term memory that simplifies the GRU proposed by [6] with only one update gate left. The output of S-GRU is a  $d$ -dimensional hidden-state vector  $\mathbf{h}_{t-1} \in \mathbb{R}^d$  that can memorize all object appearances  $\{\mathbf{x}_i\}_{i=1}^{t-1}$  appearing before frame  $\mathbf{I}_t$ , where  $\mathbf{x}_i \in \mathbb{R}^d$  denotes the object representation at frame  $\mathbf{I}_i$ . The S-GRU updates its states  $\mathbf{h}_1 \xrightarrow{x_2} \mathbf{h}_2 \cdots \xrightarrow{x_{t-1}} \mathbf{h}_{t-1}$  incrementally across the video frames, which can effectively capture the long-range dynamics of the object that are robust against occlusions and drifting with less memory overhead.

Finally, as shown in Figure 2(c), the learned hidden-state vector  $\mathbf{h}_{t-1}$  is correlated with the query image features  $\mathbf{X}_t$  to generate a target-specific attention map  $\mathbf{M}_t^{att} \in \mathbb{R}^{w \times h}$ , which highlights the target-specific region while suppressing other distractors. Then, we concatenate the attention map  $\mathbf{M}_t^{att}$  and the GCF features  $\mathbf{X}_t^{gcf}$  to further refine the

target-specific features of the query image. Moreover, we also concatenate features  $\mathbf{X}_1$  and its ground-truth mask  $\mathbf{M}_1$  from the first frame to further strengthen the target-specific feature representation. Finally, the concatenated features  $\mathbf{M}_t^{att} \oplus \mathbf{X}_t^{gcf} \oplus \mathbf{M}_1 \oplus \mathbf{X}_1$  are fed into the decoder module to produce the final segmentation mask  $\hat{\mathbf{M}}_t$ , with skip-connections to fuse multi-scale features of different layers like U-Net [36].

#### 3.2. Short-term Memory Sub-network

As aforementioned, the short-term memory sub-network is to model the inter-frame temporal correlation. Previous works achieve this by propagating the mask from previous frame to current frame, either by directly concatenating the previously predicted mask and the current frame [33, 45, 19, 39] or depending on the optical-flow guided pixel matching between two sequential frames [13, 5, 37, 27]. However, the former is easy to introduce noisy backgrounds into the object regions, especially on the object boundaries, leading to sub-optimal accuracy. Although the latter seems reasonable, there exist two limitations: First, it is very computationally expensive to estimate optical flows. Second, estimating optical flows needs to compute point-to-point mapping between two pixels, which is too restrictive [25]. For the high-level feature maps, they involve both the strength of the responses and their spatial locations [10], where each feature corresponds to a single site in the predicted mask. Hence, mask propagation can be implemented via feature propagation. Moreover, due to the fact that each feature in the high-level feature maps represents a local region inside the receptive field of the CNN filter instead of a single image pixel, a linear combination of these features to implement feature propagation serves well to model the spatial-temporal interactions between the local regions across video frames, thereby enabling to well preserve their spatial-temporal consistency across the frames. Motivated by this analysis, we propose to propagate features by spatial-temporal GCF, that is, using graph convolutions to linearly combine spatial-temporal neighbors.

**Notations.** As shown in Figure 2(a), given the short-term memory set  $\mathcal{X}_{t-k}^t$ , we define the spatial-temporal graph as  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X})$  [44], where  $\mathcal{V} = \{v_i\}_{i=1}^N$  denotes the node set with size  $N = (k+1)wh$ ,  $\mathcal{E} = \{e_{ij}\}_{i,j=1}^N$  is the edge set, where  $e_{ij}$  models the pairwise relations between any two nodes  $i$  and  $j$ ,  $\mathbf{A} \in \mathbb{R}^{N \times N}$  is the adjacency matrix whose entry  $\mathbf{A}(i, j)$  is the weight of edge  $e_{ij}$ ,  $\mathbf{X} = [\mathbf{x}_1^T; \dots; \mathbf{x}_N^T] \in \mathbb{R}^{N \times d}$  is the feature matrix constructed by set  $\mathcal{X}_{t-k}^t$ , where  $\mathbf{x}_i \in \mathbb{R}^d$  denotes the feature representation of node  $v_i$ .

**Sparse Adjacency Matrix  $\mathbf{A}$ .** For each node  $v_i \in \mathcal{V}$ , we construct its edge set  $\{e_{ij}, j \in \mathcal{N}(i) = \mathcal{N}^s(i) \cup \mathcal{N}^t(i)\}$  to capture the spatial-temporal interactions of the pair-wise local regions across video frames, where  $\mathcal{N}^s(i)$  is an  $w_s \times$



Figure 3. Laplacian smoothing effect of GCF features. Top: three query frames selected from sequence *pigs* in DAVIS 2017 val dataset [35]; bottom: the corresponding GCF feature responses, showing that the features of the same object across frames well preserve spatial-temporal consistency.

$h_s$  window centered at node  $v_i$  at the current frame,  $\mathcal{N}^t(i)$  denotes an  $w_t \times h_t$  window centered at node  $v_i$  at the next frame. The number of non-zero edges in  $\mathcal{E}$  is  $N(w_s h_s + w_t h_t) \ll N^2$ , leading to a sparse  $\mathbf{A}$  with less computational cost. To learn task-specific similarity between nodes  $i$  and  $j$  for adaptive graph learning, we define the weight of edge  $e_{ij}$  as

$$\mathbf{A}(i, j) = \sigma((\mathbf{W}_1 \mathbf{x}_i)^\top (\mathbf{W}_2 \mathbf{x}_j)), \quad (1)$$

where  $\sigma(x) = \frac{1}{1+e^{-x}}$ ,  $x \in \mathbb{R}$  is the sigmoid function,  $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$  denote node features,  $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{r \times d}$  are learnable weight matrices.

**Spatial-temporal GCF.** To perform GCF, we apply the graph convolutional networks (GCNs) proposed in [23]. We design one-layer graph convolution in our DTMNet as

$$\mathbf{z} = \mathbf{X}^{gcf} \mathbf{w}, \quad (2)$$

where  $\mathbf{w} \in \mathbb{R}^d$  denotes the weight vector of the FC layer and

$$\mathbf{X}^{gcf} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X}, \quad (3)$$

where  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ ,  $\mathbf{I}$  denotes the identity matrix,  $\tilde{\mathbf{D}}$  is a diagonal matrix with  $\tilde{\mathbf{D}}(i, i) = \sum_j \tilde{\mathbf{A}}(i, j)$ .

After GCF using (3), the  $i$ -th node representation  $\mathbf{x}_i^{gcf}$  can be formulated as

$$\mathbf{x}_i^{gcf} = \sum_{j=1}^{|\mathcal{N}(i)|} \frac{\mathbf{A}(i, j)}{\sqrt{\tilde{\mathbf{D}}(i, i) \tilde{\mathbf{D}}(j, j)}} \mathbf{x}_j + \mathbf{x}_i. \quad (4)$$

It is obvious that  $\mathbf{x}_i^{gcf}$  in (4) is a linear combination of the nodes in its spatial-temporal neighborhood  $\mathcal{N}(i)$ , thereby expressing feature propagation more accurately than existing optical-flow-based methods [13, 5, 37, 27] that are limited by restrictive point-to-point mappings. Moreover, (4) is a Laplacian smoothing process [24] that calculates the new features  $\mathbf{x}_i^{gcf}$  as the weighted average of its neighboring features in  $\mathcal{N}(i)$  and itself  $\mathbf{x}_i$ . The smoothing makes the features in the same cluster similar that favorably preserves the spatial-temporal consistency of the segmented object across frames as illustrated by Figure 3, rendering a great benefit

to the downstream pixel-wise classification task in the segmentation sub-network (§ 3.4).

**Semi-supervised Classification.** When training our model, we assume that in set  $\mathcal{X}_{t-k}^t$ , the ground-truth masks  $\mathcal{M}_{t-k}^{t-1}$  are given that correspond to the labeled nodes  $\mathcal{V}_l \in \mathcal{V}$ , while the query image mask  $\mathbf{M}_t$  is to be propagated from nodes  $\mathcal{V}_l$ . We leverage a one-layer GCN which applies a softmax classifier on the output features  $\mathbf{z}$  in (2)

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{z}). \quad (5)$$

The loss function is defined as the cross-entropy error over all the labeled nodes

$$\mathcal{L}_{sem} = - \sum_{i \in \mathcal{V}_l} \mathbf{y}(i) \log(\hat{\mathbf{y}}(i)), \quad (6)$$

where  $\mathbf{y}(i) \in \{0, 1\}$  denotes the ground-truth label of node  $i \in \mathcal{V}_l$ .

### 3.3. Long-term Memory Sub-network

Using short-term memory for VOS can deal with target appearance changes well. However, it suffers from drifting problem under challenging scenarios such as severe occlusion or fast motion between sequential frames. To address this issue, we further develop the S-GRU module to capture long-term memory information as a complement, as illustrated by Figure 2(b).

For frame  $\mathbf{I}_{t-1}$ , given its features  $\mathbf{X}_{t-1}$  and segmentation mask  $\mathbf{M}_{t-1}$ , we first mask out object features as  $\mathbf{X}_{t-1} \otimes \mathbf{M}_{t-1}$ , where  $\otimes$  denotes pixel-wise product. Then, we feed the object features into a global average pooling (GAP) layer, yielding

$$\mathbf{x}_{t-1} = \text{GAP}(\mathbf{X}_{t-1} \otimes \mathbf{M}_{t-1}), \quad (7)$$

which captures global context information that is robust against object appearance variations. Next, the S-GRU leverages  $\mathbf{x}_{t-1}$  in (7) and the previous state  $\mathbf{h}_{t-2}$  to compute the new state  $\mathbf{h}_{t-1}$ . The state vector  $\mathbf{h}$  plays a key role in S-GRU since it well captures the long-term dynamics of the object across frames. Then, the learning process is formulated as

$$\begin{aligned} \mathbf{z}_{t-1} &= \sigma(\mathbf{W}[\mathbf{x}_{t-1}; \mathbf{h}_{t-2}]), \\ \mathbf{h}_{t-1} &= (1 - \mathbf{z}_{t-1}) \odot \mathbf{h}_{t-2} + \mathbf{z}_{t-1} \odot \mathbf{x}_{t-1}, \end{aligned} \quad (8)$$

where  $\odot$  denotes the element-wise multiplication,  $\sigma$  is the sigmoid function,  $\mathbf{W} \in \mathbb{R}^{d \times 2d}$  is a learnable weight matrix. Different from the ConvGRU [37] that consists of update and reset gates, our S-GRU in (8) only has update gate  $\mathbf{z}_{t-1}$ , which reduces computational complexity significantly. In (8), the new state  $\mathbf{h}_{t-1}$  is a weighted sum of the current object representation  $\mathbf{x}_{t-1}$  and the previous state  $\mathbf{h}_{t-2}$  that memorizes the dynamic object appearances across all previous frames. If the update gate  $\mathbf{z}_{t-1}$  is close to one, the memories encoded in  $\mathbf{h}_{t-2}$  will be forgotten.

Table 1. Comparison of our DTMNet with the state of the arts on DAVIS 2016 val. **Red** and **blue** bold fonts indicate the best, the second-best performance respectively.

Method	OL	$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J}$ Mean $\uparrow$	$\mathcal{J}$ Recall $\uparrow$	$\mathcal{J}$ Decay $\downarrow$	$\mathcal{F}$ Mean $\uparrow$	$\mathcal{F}$ Recall $\uparrow$	$\mathcal{F}$ Decay $\downarrow$	Time (s) $\downarrow$
MSK [33]	✓	77.6	79.7	93.1	8.9	75.4	87.1	9.0	12
LIP [29]	✓	78.5	78.0	88.6	<b>5.0</b>	79.0	86.8	<b>6.0</b>	-
OSVOS [2]	✓	80.2	79.8	93.6	14.9	80.6	92.6	15.0	9
Lucid [21]	✓	83.6	84.8	-	-	82.3	-	-	>30
STCNN [46]	✓	83.8	83.8	96.1	<b>4.9</b>	83.8	91.5	6.4	3.9
CINM [1]	✓	84.2	83.4	94.9	12.3	85.0	92.1	14.7	>30
OnAVOS [40]	✓	85.5	<b>86.1</b>	96.1	5.2	84.9	89.7	<b>5.8</b>	13
OSVOS-S [30]	✓	86.6	85.6	<b>96.8</b>	5.5	87.5	<b>95.9</b>	8.2	4.5
PRemVOS [28]	✓	<b>86.8</b>	84.9	96.1	8.8	<b>88.6</b>	94.7	9.8	>30
MHP-VOS [49]	✓	<b>86.9</b>	<b>85.7</b>	<b>96.6</b>	-	<b>88.1</b>	<b>94.8</b>	-	>14
VPN [16]	✗	67.9	70.2	82.3	12.4	65.5	69.0	14.4	0.63
OSMN [50]	✗	73.5	74.0	87.6	9.0	72.9	84.0	10.6	0.14
VideoMatch [15]	✗	-	81.0	-	-	-	-	-	0.32
FAVOS [4]	✗	81.0	<b>82.4</b>	<b>96.5</b>	<b>4.5</b>	79.5	89.4	<b>5.5</b>	1.8
FEELVOS [39]	✗	81.7	81.1	90.5	13.7	<b>82.2</b>	86.6	14.1	0.45
RGMP [45]	✗	<b>81.8</b>	81.5	91.7	10.9	82.0	<b>90.8</b>	10.1	0.13
AGAM-VOS [19]	✗	<b>81.8</b>	81.4	93.6	9.4	82.1	90.2	9.8	<b>0.07</b>
<b>DTMNet</b>	✗	<b>85.4</b>	<b>85.9</b>	<b>96.0</b>	<b>4.7</b>	<b>84.9</b>	<b>92.0</b>	<b>5.7</b>	<b>0.12</b>

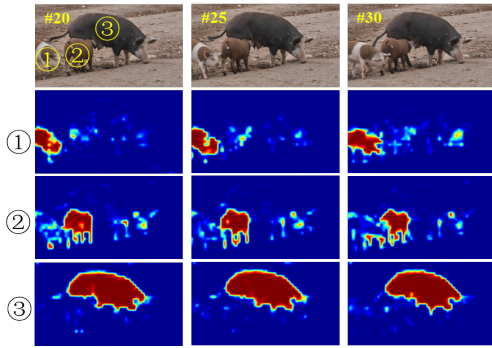


Figure 4. Illustration of the target-specific attention maps. The same query frames are shown in Figure 3, including three pigs (ID numbers: ①, ②, ③) to be segmented.

### 3.4. Segmentation Sub-network

Figure 2(c) illustrates the architecture of our segmentation sub-network. Similar to U-Net [36], our segmentation network uses skip-connections to fuse multi-scale features from the encoder to the decoder modules. The encoder uses the ResNet101 backbone network [11] with dilated convolutions to set the stride of the deepest layer to 16. For query frame  $I_t$ , the deepest layer outputs feature maps  $X_t$ . Then, we correlate  $X_t$  with the hidden-state vector  $h_{t-1}$  learned from the long-term memory sub-network, yielding the target-specific attention map  $\mathbf{M}_t^{att} = X_t \odot h_{t-1}$ . As illustrated by Figure 4, the learned  $\mathbf{M}_t^{att}$  can effectively highlight the target-specific regions while suppressing other distractors such as other objects and backgrounds. Next, af-

ter achieving the GCF features  $X_t^{gcf}$  from the short-term memory sub-network, we feed the concatenated features  $X_t^{gcf} \oplus \mathbf{M}_t^{att} \oplus X_1 \oplus \mathbf{M}_1$  into the decoder. Meanwhile, we leverage skip-connections to concatenate the feature maps from the decoder and its counterparts from the encoder, which are then gradually upsampled by a factor of two at a time. Afterwards, they are concatenated with the following layer features. Finally, the aggregated features are fed into a convolutional layer following a softmax layer to predict the object mask  $\hat{\mathbf{M}}_t$ . As the short-term memory sub-network, the loss function here is also defined as the cross-entropy loss for pixel-wise classification task:

$$\mathcal{L}_{sup} = - \sum_t \sum_{ij} \mathbf{M}_t(i, j) \log \hat{\mathbf{M}}_t(i, j), \quad (9)$$

where  $\mathbf{M}_t \in \{0, 1\}^{w \times h}$  denotes the ground-truth mask of frame  $I_t$ .

Finally, the loss function for the whole network training is defined as

$$\mathcal{L} = \mathcal{L}_{sem} + \lambda \mathcal{L}_{sup}, \quad (10)$$

where  $\mathcal{L}_{sem}$  is defined in (6),  $\lambda > 0$  is a pre-defined trade-off parameter.

## 4. Experimental Results

### 4.1. Implementation Details

Following AGAME-VOS [19], the training process of our DTMNet is divided into two stages:

**Stage 1.** Firstly, we train our DTMNet using the Adam optimizer [22] to minimize the loss  $\mathcal{L}$  in (10) on DAVIS

2017 [35] and YouTube-VOS [48] datasets for 80 epochs, where all training images are resized to  $240 \times 432$  pixels. Each batch contains 4 videos, where 8 frames are randomly selected for training in each video. The hyperparameters in our DTMNet are set empirically as learning rate =  $1e - 4$ , learning rate decay = 0.95 and weight decay =  $1e - 5$ .

**Stage 2.** Next, we fine-tune the trained model at **Stage 1** on the same datasets for 100 epochs but the images are resized to  $480 \times 864$  pixels which is twice the size of the input images at **Stage 1**. Each batch contains 2 videos with randomly selected 5 frames in each video. The parameters are also set empirically as learning rate =  $1e - 5$ , learning rate decay = 0.985 and weight decay =  $1e - 6$ .

The DTMNet is implemented in Pytorch and an Nvidia GTX 2080Ti is used for acceleration. All of the training procedures can be completed within one day.

## 4.2. Datasets and Evaluation Metrics

**Datasets.** We train and evaluate the DTMNet on three VOS benchmark datasets, including DAVIS 2016 [34], DAVIS 2017 [35] and YouTube-VOS [48]. The DAVIS 2016 is a densely-annotated VOS dataset, which contains 30 training and 20 validation video sequences of high-quality with 3,455 highly accurate pixel-wise annotation in total. The DAVIS 2017 enlarges the DAVIS 2016 by introducing more additional videos with multi-objects. The DAVIS 2017 contains a training set with 60 sequences, a validation set with 30 sequences, a test-dev set with 30 sequences and a test-challenge set with 30 sequences. The YouTube-VOS is the first large-scale VOS dataset, which is more than 30 times larger than existing largest dataset at that time. The YouTube-VOS consists of 3,471 videos in the training set, 474 videos in the validation set with 65 seen categories, and 26 unseen categories in the training set.

**Evaluation Metrics.** We use the standard metrics provided by the DAVIS challenge [35], including the region similarity  $\mathcal{J}$ , contour accuracy  $\mathcal{F}$  and the mean of the two metrics  $\mathcal{J}\&\mathcal{F}$ . Given the estimated segmentation mask  $\hat{\mathbf{M}}$  and the ground-truth mask  $\mathbf{M}$ , the region similarity is calculated as  $\mathcal{J} = \frac{|\hat{\mathbf{M}} \cap \mathbf{M}|}{|\hat{\mathbf{M}} \cup \mathbf{M}|}$ . The contour accuracy is measured by the F-measure  $\mathcal{F}$  between the contour-based precision  $\mathcal{P}$  and recall  $\mathcal{R}$  as  $\mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}}$ .

## 4.3. Comparison with the State-of-the-arts

We compare our DTMNet with some state-of-the-art online-learning (OL) VOS methods and some offline ones on the DAVIS 2016, the DAVIS 2017 and the YouTube-VOS benchmark datasets. It is worth noting that the our DTMNet does not resort to any post-processing or OL technique.

**Results on DAVIS 2016.** Table 1 lists the evaluation results on DAVIS 2016 by our DTMNet and 17 state-of-the-

Table 2. Comparison of our DTMNet with the state of the arts on DAVIS 2017 val.

Method	OL	$\mathcal{J}\&\mathcal{F} \uparrow$	$\mathcal{J} \text{ Mean} \uparrow$	$\mathcal{F} \text{ Mean} \uparrow$	Time (s) $\downarrow$
MSK [33]	✓	54.3	51.2	57.3	15
OSVOS [2]	✓	60.3	56.6	63.9	<b>11</b>
LIP [29]	✓	61.1	59.0	63.2	-
STCNN [46]	✓	61.7	58.7	64.6	6
OnAVOS [40]	✓	65.4	61.6	69.1	26
OSVOS-S [30]	✓	68.0	64.7	71.3	<b>8</b>
CINM [1]	✓	<b>70.6</b>	<b>67.2</b>	<b>74.0</b>	50
MHP-VOS [49]	✓	<b>75.3</b>	<b>71.8</b>	<b>78.8</b>	20
OSMN [50]	✗	54.8	52.5	57.1	0.28
SiamMask [41]	✗	56.4	54.3	58.5	0.02
FAVOS [4]	✗	58.2	54.6	61.8	1.2
VideoMatch [15]	✗	62.4	56.6	68.2	0.35
RANet [43]	✗	65.7	63.2	68.2	-
RGMP [45]	✗	66.7	64.8	68.6	0.28
AGSS-VOS [27]	✗	67.4	64.9	69.9	-
AGAM-VOS [19]	✗	70.0	67.2	72.7	-
DMM-Net [52]	✗	70.7	<b>68.1</b>	73.3	<b>0.13</b>
FEELVOS [39]	✗	<b>71.6</b>	<b>69.1</b>	<b>74.0</b>	0.51
<b>DTMNet</b>	✗	<b>71.5</b>	<b>69.1</b>	<b>73.9</b>	<b>0.17</b>

art OL and offline VOS methods in comparison. Among the offline methods, our DTMNet achieves the best performance in terms of  $\mathcal{J}\&\mathcal{F}$  (85.4%),  $\mathcal{J}$  Mean (85.9%),  $\mathcal{F}$  Mean (84.9%) and  $\mathcal{F}$  Recall (92.0%), the second-best performance with  $\mathcal{J}$  Recall of 96.0%,  $\mathcal{J}$  Decay of 4.7% and  $\mathcal{F}$  Decay of 5.7%. Furthermore, the DTMNet is the first runner-up with a fast speed of 0.12 s/frame, closely following the AGAM-VOS that runs at 0.07 s/frame. Even compared with the OL methods, the DTMNet still has a competing  $\mathcal{J}\&\mathcal{F}$  of 85.4%, which is only slightly lower than the best-performing MHP-VOS with  $\mathcal{J}\&\mathcal{F}$  of 86.9% by 1.5%. Besides, the DTMNet runs at 0.12 s/frame, which is much faster than the MHP-VOS at a speed of more than 14 s/frame.

**Results on DAVIS 2017.** The DAVIS 2017 considers multi-object scenarios, making it more challenging than the DAVIS 2016 that is only for single-object segmentation. Table 2 lists the comparison results of our DTMNet with 18 state-of-the-art OL and off-line methods. Among them, we can observe that our DTMNet has the best performance in terms of  $\mathcal{J}$  Mean (69.1%), and the second-best  $\mathcal{J}\&\mathcal{F}$  of 71.5% and  $\mathcal{F}$  Mean of 73.9%, closely following the best-performing FEELVOS in terms of  $\mathcal{J}\&\mathcal{F}$  (71.6%) and  $\mathcal{F}$  Mean (74.0%) with only a small gap of 0.1%, but our DTMNet runs at 0.17 s/frame on DAVIS 2017 val, which is much faster than FEELVOS that is 0.51 s/frame. Furthermore, the DTMNet even outperforms the second best-performing of-fline method CINM in terms of  $\mathcal{J}\&\mathcal{F}$  and  $\mathcal{J}$  Mean by 0.9% and 1.9%, respectively, demonstrating the effectiveness of

Table 3. Comparison of our DTMNet with the state of the arts on YouTube-VOS dataset.

Method	OL	$\mathcal{G} \uparrow$	$\mathcal{J}_s \uparrow$	$\mathcal{F}_s \uparrow$	$\mathcal{J}_u \uparrow$	$\mathcal{F}_u \uparrow$
MSK [33]	✓	53.1	59.9	59.5	45.0	47.9
OnAVOS [40]	✓	55.2	<b>60.1</b>	<b>62.7</b>	46.6	51.4
OSVOS [2]	✓	<b>58.8</b>	59.8	60.5	<b>54.2</b>	<b>60.7</b>
S2S [47]	✓	<b>64.4</b>	<b>71.0</b>	<b>70.0</b>	<b>55.5</b>	<b>61.2</b>
OSMN [50]	✗	51.2	60.0	60.1	40.6	44.0
DMM-Net [52]	✗	51.7	58.3	60.7	41.6	46.3
SiamMask [41]	✗	52.8	60.2	58.2	45.1	47.7
RGMP [45]	✗	53.8	59.5	-	45.2	-
RVOS [38]	✗	56.8	63.6	67.2	45.5	51.0
CapsuleVOS [7]	✗	<b>62.3</b>	<b>67.3</b>	<b>68.1</b>	<b>53.7</b>	<b>59.9</b>
<b>DTMNet</b>	✗	<b>65.6</b>	<b>66.1</b>	<b>68.9</b>	<b>60.5</b>	<b>66.8</b>

the dual temporal memory learning strategy in our DTM-Net.

**Results on YouTube-VOS.** The YouTube-VOS computes  $\mathcal{J}$  and  $\mathcal{F}$  on seen and unseen categories, denoted as  $\mathcal{J}_s$ ,  $\mathcal{F}_s$ ,  $\mathcal{J}_u$ ,  $\mathcal{F}_u$  in Table 3. The seen categories are included in both the training and the validation sets while the unseen categories only exist in the validation set. As listed by Table 3, our DTMNet achieves the best global mean  $\mathcal{G}$  of 65.6%, outperforming the second best-performing CapsuleVOS ( $\mathcal{G} = 62.3\%$ ) by a large margin. Besides, our DTMNet even outperforms the best-performing offline method S2S by 1.2% in terms of  $\mathcal{G}$ . Especially, our DTMNet achieves excellent performance on the unseen categories with  $\mathcal{J}_u = 60.5\%$  and  $\mathcal{F}_u = 66.8\%$ , significantly outperforming the second-best method CapsuleVOS by 6.8% and 6.9% and even outperforming the best OL method S2S by 5.0% and 5.6%, respectively. The experimental results demonstrate the favorable generalization capability of our DTMNet to unseen categories. We argue that this is due to the fact that the short-term memory sub-network learning is guided by the semi-supervised loss  $\mathcal{L}_{sem}$  (6).

#### 4.4. Ablation Study

We compare three variants of our DTMNet, including those without long-term memory sub-net (DTMNet-L), short-term memory sub-net (DTMNet-S) and graph learning model (DTMNet-W denotes removing the weights in (1)). We evaluate them on the DAVIS 2016 val and list their results in Table 4. The DTMNet-S achieves a  $\mathcal{J}$  of 81.5%, which is lower than the DTMNet by 4.4%, which verifies the effectiveness of the short-term temporal information that can help to boost the accuracy of VOS. Moreover, the DTMNet-L only has a  $\mathcal{J}$  of 71%, which is significantly lower than the DTMNet by 14.9%. This shows the key role of the long-term temporal information that makes the model robust against occlusions and drifting, which sig-

Table 4. Ablative experiments of our DTMNet on DAVIS 2016 val. DTMNet-A, A=S, L, W, denotes the DTMNet without short-memory, long-memory and graph learning modules, respectively.

Metric	DTMNet	DTMNet-S	DTMNet-L	DTMNet-W
$\mathcal{J}$	85.9	81.5	71	85.2

nificantly affects the performance of our model. Finally, we can observe that  $\mathcal{J}$  is dropped from 85.9% to 85.2% when removing the weights of the adjacency matrix in (1), which verifies the effectiveness of using graph learning structure that can also help to boost the performance of our model to some extent.

#### 4.5. Qualitative Results

Figure 5 shows some qualitatively visual results on DAVIS 2016, DAVIS 2017 and YouTube-VOS datasets. We select some challenging videos from these three datasets. We can observe that our DTMNet still achieves favorable segmentation results when the targets suffer from various challenges like fast motion (the first column top), large-scale variations (the first column bottom and the second column top) and interacting objects (the second column bottom).

### 5. Conclusions

In this paper, we have proposed an end-to-end DTMNet for VOS which mainly includes a short-term and a long-term memory sub-networks. The former models the fine-grained spatial-temporal interactions between local regions across neighboring frames via a graph-based learning framework, which can well preserve the visual consistency of local regions over time. The latter models the long-range dynamics of object via an S-GRU, making the segmentation robust against occlusions and drift errors. Extensive evaluations on three benchmark datasets including DAVIS 2016, DAVIS 2017 and YouTube-VOS demonstrate favorable performance of our method over state-of-the-art methods in terms of both speed and accuracy.

### References

- [1] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, pages 5977–5986, 2018. 6, 7
- [2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, pages 221–230, 2017. 1, 2, 6, 7, 8
- [3] Lin Chen, Jianbing Shen, Wenguan Wang, and Bingbing Ni. Video object segmentation via dense trajectories. *TMM*, 17(12):2225–2234, 2015. 2





Figure 5. Some qualitative results of our DTMNet on DAVIS 2016 val (the first column), DAVIS 2017 val (top of the second column) and YouTube-VOS (bottom of the second column) respectively. The sequences are *soapbox*, *parkour*, *motocross-jump* and *3e03f623bb*. Best viewed in color.

- [4] 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*, pages 7415–7424, 2018. 2, 6, 7
- [5] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, pages 686–695, 2017. 1, 3, 4, 5
- [6] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. 3, 4
- [7] Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Capsulevos: Semi-supervised video object segmentation using capsule routing. In *ICCV*, pages 8480–8489, 2019. 8
- [8] Jeffrey L Elman. Finding structure in time. *COGS*, 14(2):179–211, 1990. 3
- [9] Jiaming Guo, Zhuwen Li, Loong-Fah Cheong, and Steven Zhiying Zhou. Video co-segmentation for meaningful action extraction. In *ICCV*, pages 2232–2239, 2013. 1
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015. 4
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6
- [12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *NEURAL COMPUT*, 9(8):1735–1780, 1997. 3
- [13] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *NeurIPS*, pages 325–334, 2017. 4, 5
- [14] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *ECCV*, pages 786–802, 2018. 2
- [15] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, pages 54–70, 2018. 2, 6, 7
- [16] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. In *CVPR*, pages 451–461, 2017. 6
- [17] Won-Dong Jang, Chulwoo Lee, and Chang-Su Kim. Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In *CVPR*, pages 696–704, 2016. 2
- [18] Bo Jiang, Ziyang Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *CVPR*, pages 11313–11320, 2019. 2
- [19] 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*, pages 8953–8962, 2019. 1, 2, 4, 6, 7
- [20] Kazuya Kawakami. Supervised sequence labelling with recurrent neural networks. *Ph. D. thesis*, 2008. 3
- [21] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. In *The DAVIS Challenge on Video Object Segmentation*, 2017. 1, 2, 3, 6
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 5
- [24] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, pages 3538–3545, 2018. 4, 5
- [25] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *CVPR*, pages 5997–6005, 2018. 4
- [26] Yin Li, Jian Sun, and Heung-Yeung Shum. Video object cut and paste. In *ToG*, volume 24, pages 595–600, 2005. 1
- [27] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Agss-vos: Attention guided single-shot video object segmentation. In *ICCV*, pages 3949–3957, 2019. 4, 5, 7
- [28] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, pages 565–580, 2018. 2, 3, 6
- [29] Ye Lyu, George Vosselman, Gui-Song Xia, and Michael Ying Yang. Lip: Learning instance propagation for video object segmentation. *arXiv preprint arXiv:1910.00032*, 2019. 6, 7
- [30] 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. *TPAMI*, 41(6):1515–1530, 2018. 6, 7

- [31] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. *arXiv preprint arXiv:1904.00607*, 2019. [3](#)
- [32] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, pages 1777–1784, 2013. [2](#)
- [33] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, pages 2663–2672, 2017. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [34] 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*, pages 724–732, 2016. [7](#)
- [35] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. [5](#), [7](#)
- [36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. [4](#), [6](#)
- [37] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, pages 4481–4490, 2017. [1](#), [2](#), [3](#), [4](#), [5](#)
- [38] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, pages 5277–5286, 2019. [1](#), [2](#), [3](#), [8](#)
- [39] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, pages 9481–9490, 2019. [1](#), [4](#), [6](#), [7](#)
- [40] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017. [1](#), [2](#), [6](#), [7](#), [8](#)
- [41] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, pages 1328–1338, 2019. [1](#), [7](#), [8](#)
- [42] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Selective video object cutout. *TIP*, 26(12):5645–5655, 2017. [1](#)
- [43] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, pages 3978–3987, 2019. [7](#)
- [44] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019. [4](#)
- [45] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, pages 7376–7385, 2018. [1](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [46] Kai Xu, Longyin Wen, Guorong Li, Liefeng Bo, and Qingming Huang. Spatiotemporal cnn for video object segmentation. In *CVPR*, pages 1379–1388, 2019. [3](#), [6](#), [7](#)
- [47] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, pages 585–601, 2018. [1](#), [2](#), [3](#), [8](#)
- [48] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *ECCV*, pages 585–601, 2018. [7](#)
- [49] Shuangjie Xu, Daizong Liu, Linchao Bao, Wei Liu, and Pan Zhou. Mhp-vos: Multiple hypotheses propagation for video object segmentation. In *CVPR*, pages 314–323, 2019. [1](#), [2](#), [6](#), [7](#)
- [50] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, pages 6499–6507, 2018. [1](#), [2](#), [6](#), [7](#), [8](#)
- [51] Donghun Yeo, Jeany Son, Bohyung Han, and Joon Hee Han. Superpixel-based tracking-by-segmentation using markov chains. In *CVPR*, pages 1812–1821, 2017. [1](#)
- [52] Xiaohui Zeng, Renjie Liao, Li Gu, Yuwen Xiong, Sanja Fidler, and Raquel Urtasun. Dmm-net: Differentiable mask-matching network for video object segmentation. *ICCV*, pages 3929–3938, 2019. [7](#), [8](#)