

Joint Inductive and Transductive Learning for Video Object Segmentation

Yunyao Mao¹ Ning Wang¹ Wengang Zhou^{1,2,*} Houqiang Li^{1,2,*}

¹ CAS Key Laboratory of Technology in GIPAS, EEIS Department, University of Science and Technology of China

² Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

{myy2016,wn6149}@mail.ustc.edu.cn, {zhwg,lihq}@ustc.edu.cn

Abstract

Semi-supervised video object segmentation is a task of segmenting the target object in a video sequence given only a mask annotation in the first frame. The limited information available makes it an extremely challenging task. Most previous best-performing methods adopt matching-based transductive reasoning or online inductive learning. Nevertheless, they are either less discriminative for similar instances or insufficient in the utilization of spatio-temporal information. In this work, we propose to integrate transductive and inductive learning into a unified framework to exploit the complementarity between them for accurate and robust video object segmentation. The proposed approach consists of two functional branches. The transduction branch adopts a lightweight transformer architecture to aggregate rich spatio-temporal cues while the induction branch performs online inductive learning to obtain discriminative target information. To bridge these two diverse branches, a two-head label encoder is introduced to learn the suitable target prior for each of them. The generated mask encodings are further forced to be disentangled to better retain their complementarity. Extensive experiments on several prevalent benchmarks show that, without the need of synthetic training data, the proposed approach sets a series of new state-of-the-art records. Code is available at <https://github.com/maoyunyao/JOINT>.

1. Introduction

Semi-supervised video object segmentation (VOS) aims at segmenting the target object in a video sequence with the supervision given in the first frame by a pixel-wise segmentation mask. It has received tremendous attention in recent years for its wide applications. Since the target-specific information is only given in the first frame, and the target may undergo fast-moving and dramatic deformation, how to make full use of the limited information to perform accu-

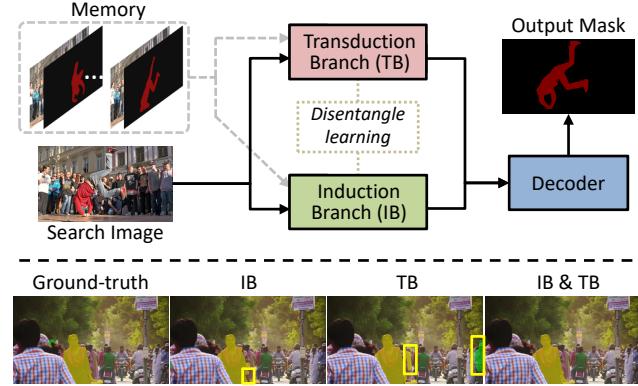


Figure 1. An overview of our approach. The transduction branch (TB) aggregates rich spatio-temporal cues from sampled history frames and propagates them to the current frame, and the induction branch (IB) performs online inductive learning to obtain discriminative target information.

rate segmentation is thus extremely challenging.

Top-performing methods can be roughly categorized as transductive reasoning and online inductive learning. In transductive formulation, direct reasoning from reference frames (labeled samples) to the current frame (unlabeled sample) is performed to facilitate segmentation. In recent transductive solutions [19, 23, 28, 33, 37, 44, 45, 54], feature matching has become the mainstream choice, where pixel-level affinity or distance maps between the current frame and reference frames are obtained to deliver rich historical target information. This specific-to-specific reasoning favorably retains the temporal information with attractive time efficiency. Despite achieving the state-of-the-art performance, it heavily relies on the offline learned feature embeddings for accurate matching, thus suffers limited generalization and discrimination capabilities.

On the other hand, online inductive learning utilizes reference frames to train a target model (general rule), which is then applied to subsequent frames to perform segmentation. Recently, efficient online discriminative learning [4, 12] in visual object tracking has been introduced to the VOS community for its well-acknowledged adaptivity and generaliz-

*Corresponding authors: Wengang Zhou and Houqiang Li

ability. The few-shot learner in [5, 36] provides superior distractor discrimination capability. Nevertheless, this inductive formulation treats reference frames as independent training samples and fails to explore the underlying context [48, 55, 58]. Rich temporal information that resides in the video flow is thus not fully exploited, which has been proven by previous transductive inference works [33, 57] to be crucial for obtaining spatio-temporal consistent results.

The above analysis indicates that transductive reasoning and online inductive learning are naturally complementary. The former performs better in spatio-temporal dependency modeling but struggles to discriminate similar distractors, while the latter is just the other way around. Although it is intuitive to jointly integrate these two models, how to explore their complementary potentials in a unified framework has been rarely involved. Since they deal with the VOS task via different perspectives, there exist two main challenges for this seemingly straightforward integration. First, most transductive approaches rely on intermediate results as features [33, 37, 25, 28] or distance maps [41, 54], while the online inductive learning directly outputs masks [27, 36] or intermediate encodings [5]. How to design an appropriate merging strategy to effectively fuse these diverse representations while retaining their complementarity is an open problem. Second, how to tightly bridge these two different models to avoid redundant computations for efficient online VOS deserves further exploration.

In this work, as shown in Figure 1, we propose a novel two-branch architecture to jointly integrate transductive reasoning and online inductive learning within a unified framework for high-performance VOS. The transduction branch aggregates rich spatial-temporal information while the induction branch provides superior discrimination capability. To solve the aforementioned problems and narrow the gap between the two branches, we make several key designs in the proposed framework: (1) In the transduction branch, we extend the attention mechanism adopted in previous matching-based VOS frameworks [33, 57, 37] into a lightweight transformer [7, 40] architecture, which is carefully designed to facilitate temporal information propagation. To unify the inputs and outputs of the two branches, we further adopt a two-head label encoder for them to generate mask encodings as the target information carrier for VOS. (2) We propose the mask encoding decoupling regularization to reduce their redundancy and make the learned target information more differentiable and complementary. (3) Finally, our two lightweight branches mutually share plentiful blocks such as backbone, partial label generator, and segmentation decoder, making our framework efficient and end-to-end trainable. We perform extensive experiments on DAVIS [35] and YouTube-VOS [52] benchmarks. Our proposed approach outperforms other state-of-the-art methods with comparable running efficiency.

In summary, we make the following three contributions:

- We propose a novel two-branch architecture to tackle the video object segmentation, which absorbs the merits of both offline learned transductive reasoning and online inductive learning.
- For the transduction branch, a lightweight transformer architecture is proposed to conduct spatio-temporal dependency modeling and content propagation. To our best knowledge, this is the first attempt to leverage the transformer architecture in VOS.
- To bridge the gap between two branches and better exploit their complementary characteristics, we propose to learn disentangled mask encodings.

2. Related Work

Matching based methods. Recent methods [25, 28, 33, 37, 38, 41, 49, 54, 57] adopt feature matching for video object segmentation. In these methods, embeddings are firstly obtained through a backbone network for both reference and current frames. Pixel-level comparison is then performed between them to obtain target-specific information for the current frame, which may further be fed into a segmentation decoder to obtain the final result. Among them, STMVOS [33] maintains a memory bank during online inference, and feature matching is realized by applying the non-local cross-attention mechanism, where the memory embeddings are read out to facilitate object segmentation in the current frame. GC [25], KMN [37], and GraphMemVOS [28] improve STMVOS in different aspects. TVOS [57] is the first to pose video object segmentation from a transductive inference perspective. In TVOS, a spatio-temporal dependency graph is constructed by adopting a predefined pixel-wise similarity metric. And the graph is utilized to propagate labels from sampled history frames to the current frame.

Different from previous methods [33, 57] that adopt isolated attention mechanisms for transductive reasoning, in our transduction branch, we introduce a lightweight yet complete transformer architecture to perform spatio-temporal dependency modeling and target information propagation. And we further learn the intermediate representations suitable for the propagation operation.

Online learning based methods. In many early works [1, 6, 21, 24, 29, 31, 42, 50], online fine-tuning is widely used to online introduce the target information. Despite the promising results, this plain inductive learning method is extremely time-consuming, making it unfavorable for many real-time applications. Thus, many efforts [9, 10, 30, 47, 53, 56] have been made to avoid first-frame fine-tuning.

In visual object tracking, online discriminative learning [4, 11, 12, 13, 46] has received significant attention for its superior performance and high efficiency. In these methods, efficient optimization strategies are adopted to online train convolutional filters, which are applied to subsequent

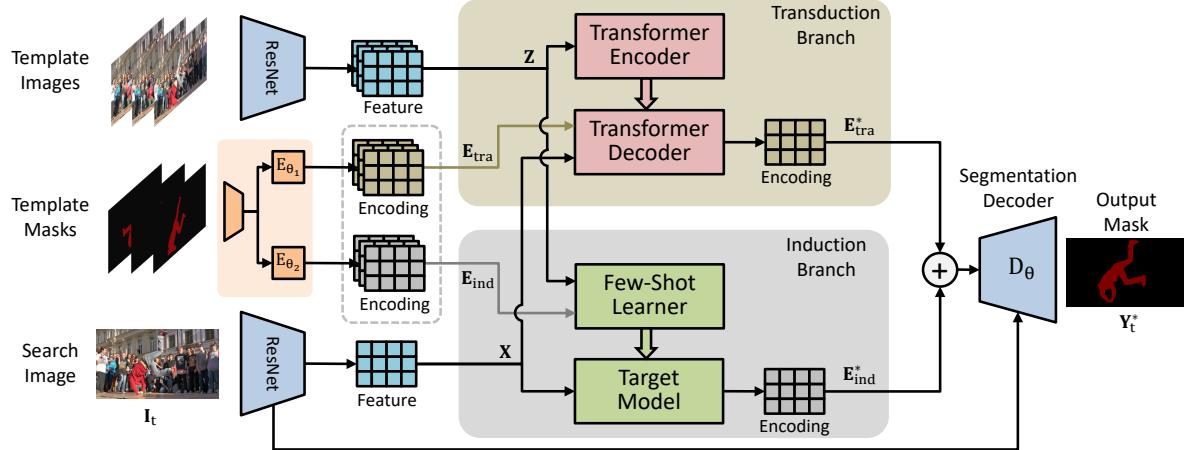


Figure 2. The overall pipeline of our approach. It consists of two complementary branches. The lightweight transformer architecture in transduction branch produces fine-grained and spatio-temporal consistent cue while the few-shot learner in induction branch provides discriminative information through online inductive learning. Two branches are integrated by learning disentangled mask encodings.

frames to carry out foreground-background classification. In [27], the closed-form ridge regression [3] is introduced to solve the VOS problem, it online optimizes a parameter matrix that maps features to segmentation masks. In FRTM [36], the online optimization paradigm in ATOM [12] is revisited and being carefully modified to adapt to the VOS task. By applying Conjugate Gradient and Gauss-Newton algorithms, the few-shot learner in FRTM is capable of learning a powerful target-specific model from the limited number of templates during inference. The learned target model conducts foreground-background classification in a pixel-wise manner, and the obtained low-resolution result is further refined by a segmentation decoder. Later LWL [5] further proposes to learn what the few-shot learner should learn. Different from FRTM, it adopts a label encoder to generate the few-shot label, which contains richer target information than the single-channel segmentation mask. Thanks to the online inductive learning, the capability to discriminate against similar instances is greatly enhanced. But the spatio-temporal consistency of the result may not be well guaranteed. In this work, we aim to integrate the aforementioned matching-based transductive reasoning and online inductive learning into a unified framework to fully exploit the complementary characteristics between them.

3. Method

3.1. Overall Pipeline

We propose a new approach for video object segmentation, which consists of two functional branches. The main idea is that matching-based transductive reasoning and online inductive learning are naturally complementary. The overall pipeline of our approach is illustrated in Figure 2. The first frame and sampled history frames constitute the template images, and the current frame serves

as the search image. Firstly, both template and search images are fed into the ResNet-50 [16] network to obtain the res3 features $\mathbf{Z} \in \mathbb{R}^{N \times H \times W \times C}$ and $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ respectively, where N is the number of template images. A two-head label encoder is employed to encode the template masks into mask encodings $\mathbf{E}_{\text{tra}} \in \mathbb{R}^{N \times H \times W \times D}$ and $\mathbf{E}_{\text{ind}} \in \mathbb{R}^{N \times H \times W \times D}$ for two parallel branches. After that, the transduction branch takes both \mathbf{Z} and \mathbf{X} as input and propagates \mathbf{E}_{tra} to search image according to the pixel-level affinity between features. The propagated result is denoted as $\mathbf{E}_{\text{tra}}^* \in \mathbb{R}^{H \times W \times D}$. Meanwhile, the online few-shot learner in the induction branch learns a target model by solving an optimization problem where \mathbf{Z} and \mathbf{E}_{ind} are treated as training sample pairs. The target model is then applied to \mathbf{X} to obtain $\mathbf{E}_{\text{ind}}^* \in \mathbb{R}^{H \times W \times D}$ of the search image, which contains discriminative target information. Finally, the obtained mask encodings from both branches along with search features from different backbone layers are incorporated and fed into the segmentation decoder to predict the final result. Note that two branches are integrated in a complementary manner by learning disentangled mask encodings, which will be discussed in Section 3.4.

3.2. Transduction Branch

As shown in Figure 3, in the transduction branch, a lightweight transformer [40] architecture is introduced to perform spatio-temporal information transduction. The attention mechanism, which is the most important component in transformer, has a strong capability in non-local dependency modeling. It transforms value $\mathbf{V} \in \mathbb{R}^{n_k \times d_v}$ according to the dot-product similarity between query $\mathbf{Q} \in \mathbb{R}^{n_q \times d_k}$ and key $\mathbf{K} \in \mathbb{R}^{n_k \times d_k}$. Here in our approach, the attention mechanism is slightly modified to better adapt to the VOS task. Firstly, query and key are normalized along the channel dimension before the dot-product operation. Then,

the intermediate result $\bar{\mathbf{Q}}\bar{\mathbf{K}}^\top \in \mathbb{R}^{n_q \times n_k}$ is rescaled to obtain a suitable softmax distribution [8, 17]. The above computing process can be formulated as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\bar{\mathbf{Q}}\bar{\mathbf{K}}^\top}{\tau}\right)\mathbf{V}, \quad (1)$$

where $\bar{\cdot}$ denotes ℓ_2 normalization along channel dimension and the denominator τ is the scaling factor.

Transformer encoder. The transformer encoder takes template feature $\mathbf{Z} \in \mathbb{R}^{N \times H \times W \times C}$ as input, which is further flattened into $\tilde{\mathbf{Z}} \in \mathbb{R}^{NHW \times C}$ for subsequent matrix operations. In self-attention layer, the query and key are obtained by applying a linear transformation $\phi(\cdot)$ to the flattened feature $\tilde{\mathbf{Z}}$ whose channel dimension is reduced from C to $C/4$. And the attention value $\mathbf{A}_{\tilde{\mathbf{Z}}} \in \mathbb{R}^{NHW \times C}$ is computed according to Eq. (1) as follows:

$$\mathbf{A}_{\tilde{\mathbf{Z}}} = \text{Attention}\left(\phi(\tilde{\mathbf{Z}}), \phi(\tilde{\mathbf{Z}}), \tilde{\mathbf{Z}}\right). \quad (2)$$

This attention value, as a residual term, is added to the original template feature $\tilde{\mathbf{Z}}$, and the result is further fed into the Instance Normalization [39] layer to obtain the encoded template feature $\tilde{\mathbf{Z}}_{\text{enc}} \in \mathbb{R}^{NHW \times C}$ as follows:

$$\tilde{\mathbf{Z}}_{\text{enc}} = \text{Instance Norm}\left(\mathbf{A}_{\tilde{\mathbf{Z}}} + \tilde{\mathbf{Z}}\right). \quad (3)$$

The transformer encoder enables template features to mutually reinforce to be more compact and representative, thus suitable for subsequent feature matching procedures performed in the transformer decoder.

Transformer decoder. The transformer decoder consists of a self-attention layer and a cross-attention layer. Firstly, the self-attention layer processes search feature $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ in a similar way to the transformer encoder, *i.e.*, a residual attention term is obtained and being merged to the original search feature as follows:

$$\begin{aligned} \mathbf{A}_{\tilde{\mathbf{X}}} &= \text{Attention}\left(\phi(\tilde{\mathbf{X}}), \phi(\tilde{\mathbf{X}}), \tilde{\mathbf{X}}\right), \\ \tilde{\mathbf{X}}_{\text{attn}} &= \text{Instance Norm}\left(\mathbf{A}_{\tilde{\mathbf{X}}} + \tilde{\mathbf{X}}\right), \end{aligned} \quad (4)$$

where $\tilde{\mathbf{X}} \in \mathbb{R}^{HW \times C}$ is the flattened search feature, $\mathbf{A}_{\tilde{\mathbf{X}}} \in \mathbb{R}^{HW \times C}$ and $\tilde{\mathbf{X}}_{\text{attn}} \in \mathbb{R}^{HW \times C}$ denote the attention value and output of the self-attention layer, respectively. Then, the cross-attention layer, which is the most important component in our lightweight transformer architecture, propagates rich temporal information according to the pixel level correspondence between the search image and template images. It takes $\tilde{\mathbf{X}}_{\text{attn}} \in \mathbb{R}^{HW \times C}$ and the encoded template feature $\tilde{\mathbf{Z}}_{\text{enc}} \in \mathbb{R}^{NHW \times C}$ as inputs to generate query $\psi(\tilde{\mathbf{X}}_{\text{attn}})$ and key $\psi(\tilde{\mathbf{Z}}_{\text{enc}})$ respectively, and transforms template mask encoding $\mathbf{E}_{\text{tra}} \in \mathbb{R}^{N \times H \times W \times D}$ to search image according to the similarity between query and key as follows:

$$\tilde{\mathbf{E}}_{\text{tra}}^* = \text{Attention}\left(\psi(\tilde{\mathbf{X}}_{\text{attn}}), \psi(\tilde{\mathbf{Z}}_{\text{enc}}), \mathbf{E}_{\text{tra}}\right), \quad (5)$$

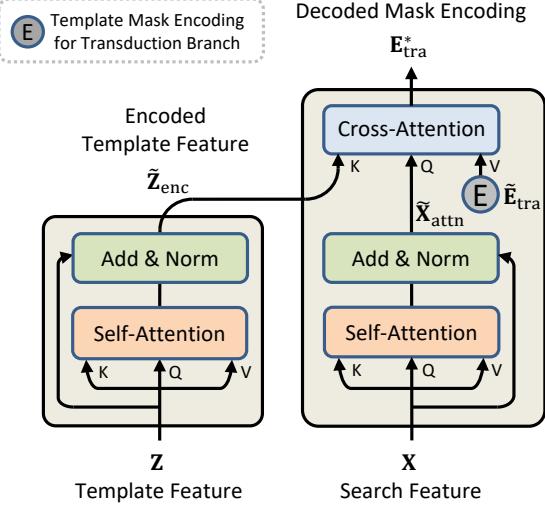


Figure 3. An overview of our lightweight transformer architecture, which is adopted in the transduction branch. It is carefully designed to provide fine-grained and temporally consistent target information propagation.

where \mathbf{E}_{tra} is flattened into $\tilde{\mathbf{E}}_{\text{tra}} \in \mathbb{R}^{NHW \times D}$ before being used as the value. And $\tilde{\mathbf{E}}_{\text{tra}}^* \in \mathbb{R}^{HW \times D}$ is reshaped back to obtain the decoded mask encoding $\mathbf{E}_{\text{tra}}^* \in \mathbb{R}^{H \times W \times D}$.

3.3. Induction Branch

The transduction branch provides fine-grained and temporally consistent mask encoding propagation, but its performance largely relies on the representation capability of the offline trained feature embeddings. Due to the absence of online adaptation, it does not perform well when encountering novel targets, and it is also difficult to discriminate similar instances. To make up for these shortcomings, in the induction branch, we adopt the few-shot learner proposed in LWL [5] to perform online inductive learning. Taking template feature \mathbf{Z} and mask encoding \mathbf{E}_{ind} as training sample pairs, it online optimizes the kernel of a convolutional layer $T_\omega : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H \times W \times D}$ by minimizing the following squared error:

$$L(\omega) = \frac{1}{2} \sum_{i=1}^N \|\mathbf{W}_i \cdot (T_\omega(\mathbf{Z}_i) - \mathbf{E}_{\text{ind},i})\|^2 + \frac{\lambda}{2} \|\omega\|^2, \quad (6)$$

where ω is the kernel to be optimized, \mathbf{W} represents the element-wise importance weights generated from labels (as with the mask encodings), i is the index of training samples, and λ is a learned regularization term. The steepest descent method is applied to iteratively minimize the squared error instead of the direct closed-form solution [3], as the latter requires time-consuming matrix inversion operations which are harmful to the running speed. For the detailed derivation of the steepest descent method please refer to [5]. Note that the entire optimization process is fully differentiable,

so it can be offline trained together with the rest part of the network in an end-to-end manner.

This online optimized inductive target model, *i.e.*, the convolutional kernel $\omega \in \mathbb{R}^{K \times K \times C \times D}$, has excellent discrimination capability. It maps the search feature \mathbf{X} into a D -dimensional target-aware mask encoding $\mathbf{E}_{\text{ind}}^* = T_\omega(\mathbf{X})$, which greatly compensates for the output of the transduction branch.

3.4. Disentangled Mask Encodings

As stated above, both branches produce mask encodings with rich target specific information. These mask encodings are element-wisely added together in our approach. And the result, along with the search features from different backbone layers, are processed by the segmentation decoder D_θ to generate the mask prediction \mathbf{Y}_t^* as follows:

$$\mathbf{Y}_t^* = D_\theta \left(\mathbf{E}_{\text{tra}}^* + \mathbf{E}_{\text{ind}}^*, \mathbf{X}_t^{[1,2,3,4]} \right), \quad (7)$$

where $\mathbf{X}_t^{[1,2,3,4]}$ is a simplified denotation of search features from different backbone layers (layer 1 to 4).

Since the two branches show diverse characteristics in dealing with the VOS task, the mask encoding suitable for each of them should be different. To this end, we propose a two-head label encoder to learn the intermediate expressions suitable for each branch. Furthermore, to reduce their redundancy and make the learned target information more differentiable and complementary, we decouple the generated mask encodings by minimizing their similarity. Specifically, we adopt the widely used cosine similarity in our approach. Given two vectors a and b , the cosine similarity can be calculated by $\cos(a, b) = \frac{a \cdot b}{\|a\| \cdot \|b\|}$. The adopted regularization loss is defined as follows:

$$\mathcal{L}_{\text{cos}}^t = \begin{cases} \cos(\text{vec}(E_{\theta_1}(\mathbf{Y}_0)), \text{vec}(E_{\theta_2}(\mathbf{Y}_0))) & t = 0 \\ \cos(\text{vec}(E_{\theta_1}(\mathbf{Y}_t^*)), \text{vec}(E_{\theta_2}(\mathbf{Y}_t^*))) & t > 0 \end{cases}, \quad (8)$$

where $\text{vec}(\cdot)$ is the vectorization operator that flattens the mask encodings into one-dimensional vectors, \mathbf{Y}_0 is the ground-truth of \mathbf{I}_0 and \mathbf{Y}_t^* is the predicted mask of \mathbf{I}_t , θ_1 and θ_2 denote different parameters of the two-head label encoder. Note that since each head ends with a ReLU layer, the cosine similarity will not be negative. As shown in Figure 4, the two-head label encoder generates diverse mask encodings for each branch. But there still exists plenty of information redundancy in between. And if we further adopt the regularization loss defined in Eq. (8), the generated mask encodings are well decoupled along the channel dimension. This ensures that after element-wise addition, the two mask encodings do not mutually disturb. So that the segmentation decoder can make use of the complementary information provided by the two branches to generate the final segmentation result.

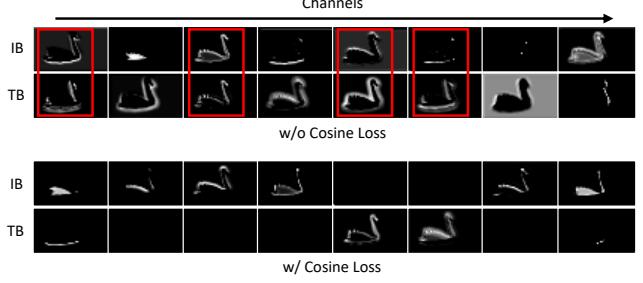


Figure 4. Visualization of the generated mask encodings. IB and TB denote induction branch and transduction branch, respectively. We can find that different branches learn diverse mask encodings. However, without any constraints, the generated encodings have plenty of information redundancy (marked in red boxes), which is alleviated by adopting the cosine loss in our approach.

3.5. Online Inference

During online inference, a memory bank \mathbf{M} is maintained to memorize the sampled history images and corresponding segmentation masks. Given a test video sequence $\mathcal{S} = \{\mathbf{I}_t\}_{t=0}^{N_{\text{test}}}$ with the initial segmentation mask \mathbf{Y}_0 of the first frame \mathbf{I}_0 , we first initialize the memory bank with $\mathbf{M}_0 = \{(\mathbf{X}_0, \mathbf{Y}_0)\}$, where $\mathbf{X}_0 = F_\theta(\mathbf{I}_0)$ is the res3 feature extracted from \mathbf{I}_0 by the backbone network F_θ . Then the initialized memory bank is used to perform object segmentation on subsequent frames as stated in Section 3.1. To better exploit the temporal information and adapt to the appearance changes in the scene, we update the memory bank with the most recently processed frames. Specifically, the new template is sampled every $T = 5$ frame and added to the memory bank along with the predicted segmentation mask. And except for the first frame, we remove the oldest sample to ensure that the size of the memory bank does not exceed $N_{\text{max}} = 20$. For multiple objects, our approach processes each of them independently and merges the predicted masks using the soft-aggregation operation [32]. Note that the merging step is performed only during online inference.

3.6. Offline Training

The whole network in our approach is end-to-end trained using generated mini-sequences $\mathcal{S}_{\text{train}} = \{(\mathbf{I}_t, \mathbf{Y}_t)\}_{t=0}^{N_{\text{train}}-1}$, which are randomly sampled from densely annotated video segments. During offline training, our network processes mini-sequences in a similar way to the online inference stage. To fully exploit the generated mini-sequences, the memory bank is updated every frame during offline training. We adopt two loss functions to supervise the learning of our network, namely segmentation loss and cosine similarity loss. The segmentation loss is used to supervise the generated mask predictions, which is computed as follows:

$$\mathcal{L}_{\text{seg}}^t = \mathcal{L}_{\text{lov}}(\mathbf{Y}_t^*, \mathbf{Y}_t), \quad (9)$$

where \mathbf{Y}_t^* is the predicted segmentation mask of \mathbf{I}_t , and \mathcal{L}_{lov} is the Lovasz segmentation loss [2]. The final loss $\mathcal{L}_{\text{final}}$ is the weighted sum of the segmentation loss $\mathcal{L}_{\text{seg}}^t$ and the aforementioned cosine similarity loss $\mathcal{L}_{\text{cos}}^t$, as follows:

$$\mathcal{L}_{\text{final}} = \frac{1}{N_{\text{train}} - 1} \sum_{t=1}^{N_{\text{train}}-1} \mathcal{L}_{\text{seg}}^t + \frac{\lambda}{N_{\text{train}}} \sum_{t=0}^{N_{\text{train}}-1} \mathcal{L}_{\text{cos}}^t, \quad (10)$$

where the hyperparameter λ is set to 0.01.

4. Experiments

We evaluate the proposed approach on DAVIS 2017 [35] and YouTube-VOS [52] datasets. For the DAVIS benchmark, we follow its standard protocol, where the \mathcal{J} score measures the region similarity, the \mathcal{F} score indicates boundary accuracy, and $\mathcal{J}\&\mathcal{F}$ is the mean of them. For comparison on YouTube-VOS datasets, \mathcal{J} and \mathcal{F} scores are reported on both the training (seen) categories and the unseen categories, and the overall score is their average. All results are obtained through the official evaluation toolkit (for DAVIS) or evaluation server (for YouTube-VOS).

4.1. Implementation Details

The backbone feature extractor used in our approach is ResNet-50 [16], which is initialized with the Mask R-CNN [15] weights. In both branches, an additional convolutional block is adopted to reduce the channel dimension of backbone `res3` feature from 1024 to 512. The scaling factor τ in Eq. (1) is set to 1/30. For the few-shot learner in the induction branch, we follow the settings used in LWL [5]. The adopted two-head label encoder generates mask encodings with channel dimension $D = 16$.

Template and search images are cropped from original frames, which are 5 times the previously estimated target size (no larger than the original frames). And the cropped patches are further resized to 832×480 .

Our network is trained on the train split of YouTube-VOS [52] and DAVIS [34] datasets. We sample $N_{\text{train}} = 4$ frames from a video segment of length $N'_{\text{train}} = 100$ to generate the mini-sequences, where the random flipping, rotation, and scaling are adopted for data augmentation. The whole training process contains 180k iterations with a batch size of 20. The ADAM [22] optimizer is adopted and the initial learning rate is set to 0.01, which is further reduced by a factor of 5 after 40k, 80k, 115k, and 165k iterations. The backbone weights are fixed in the first 90k iterations and then being optimized together in the rest 90k ones. It takes about 96 hours on 8 Nvidia GTX 1080Ti GPUs to finish the offline training process. During online inference, our approach operates at about 8 FPS on single object sequences. Code and pre-trained models will be made publicly available.

Table 1. Ablation study for branch complementarity. TB and IB denote the transduction branch and induction branch, respectively. t/s denotes second per frame. The performance is evaluated on the YouTube-VOS 2018 [52] validation set in terms of mean Jaccard (\mathcal{J}) and boundary (\mathcal{F}) scores on both seen and unseen categories.

TB	IB	$\mathcal{J}_{\text{seen}}$	$\mathcal{F}_{\text{seen}}$	$\mathcal{J}_{\text{unseen}}$	$\mathcal{F}_{\text{unseen}}$	Overall	t/s
✓		81.2	85.4	75.1	83.2	81.2	0.22
	✓	80.4	84.9	76.4	84.4	81.5	0.15
✓	✓	81.5	85.9	78.7	86.5	83.1	0.25

Table 2. Ablation study for disentangled mask encodings. For the single-head label encoder, two branches share the same mask encoding. The performance is evaluated on the YouTube-VOS 2018 [52] validation set in terms of mean Jaccard (\mathcal{J}) and boundary (\mathcal{F}) scores on both seen and unseen categories.

Version	Label Encoder	Cosine Loss	$\mathcal{J}_{\text{seen}}$	$\mathcal{F}_{\text{seen}}$	$\mathcal{J}_{\text{unseen}}$	$\mathcal{F}_{\text{unseen}}$	Overall
(1)	single-head		80.7	85.3	76.8	84.5	81.8
(2)	two-head		81.1	85.6	77.6	85.3	82.4
(3)	two-head	✓	81.5	85.9	78.7	86.5	83.1

4.2. Ablation Study

To validate the effectiveness of the key components in our proposed method, we perform two comparative experiments on the YouTube-VOS 2018 [52] validation set.

Branch complementarity. We first perform an ablation study to demonstrate the complementarity of the two branches in our approach. The experimental results are presented in Table 1. The overall performance degrades from 83.1% to 81.2% when the transduction branch is applied alone. And we can further discover that the performance degradation is mainly reflected on the unseen categories, where $\mathcal{J}_{\text{unseen}}$ and $\mathcal{F}_{\text{unseen}}$ are dropped from 78.7% and 86.5% to 75.1% and 83.2%, respectively. This proves the insufficient generalization capability of matching-based approaches to a certain extent. And we further report the result of LWL [5] as the performance of applying induction branch alone. Compared with the full version, LWL [5] has decreased results in all performance metrics, with an overall score of 81.5%. The above results show that there does exist a strong complementary relationship between the two branches of our approach. Qualitative comparisons on DAVIS 2017 validation set are shown in Figure 5.

Disentangled mask encodings. We then conduct several ablation experiments to verify the effectiveness of our proposed disentangled mask encodings for exploiting the complementarity of the two branches. Specifically, we set up three mask encoding generation strategies as follows:

- (1) We first adopt a single-head label encoder for the mask encoding generation, *i.e.*, both branches adopt the same mask encoding.
- (2) We next replace the above-mentioned single-head label encoder with a two-head one. In this setting, mask

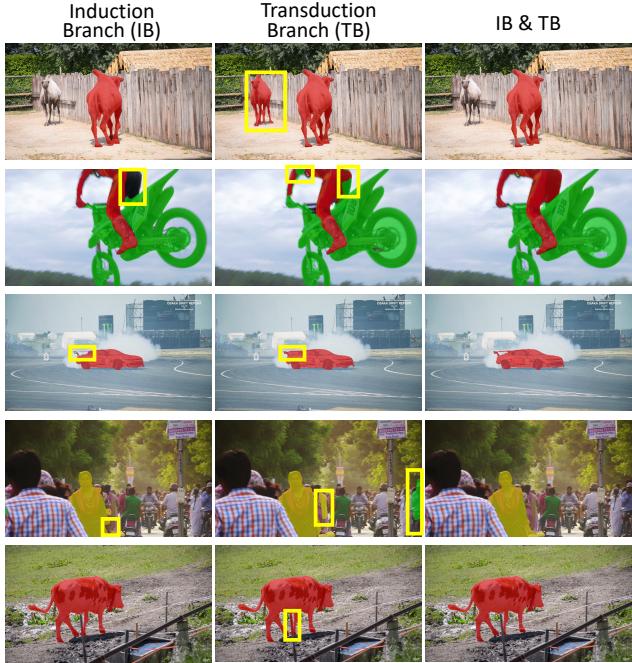


Figure 5. Qualitative comparisons on DAVIS 2017 validation set. Noticeable failures are marked with yellow boxes. By jointly exploring induction branch (IB) and transduction branch (TB), our method shows superior VOS accuracy. Best view in zoom in.

encoding for each branch is independently generated.

- (3) Based on (b), in this version, we further introduce the cosine similarity loss proposed in Section 3.4 to force the generated mask encodings to be disentangled.

As shown in Table 2, directly applying a single-head label encoder achieves an overall score of 81.8%. If we replace it with a two-head one, the overall performance is improved from 81.8% to 82.4%. This indicates that the intermediate representations of masks suitable for the two branches are different. And if we further apply the cosine similarity loss to force the generated mask encodings to be disentangled, the performance can be further improved to 83.1%.

4.3. Comparison with State-of-the-art Methods

We compare our approach with previous state-of-the-art methods on several benchmarks including DAVIS 2017 [35], YouTube-VOS 2018, and YouTube-VOS 2019 [52]. In Figure 6, we show some qualitative results in many challenging scenarios, such as occlusion, similar distractors, and appearance changes. Our JOint INductive and Transductive learning-based method is denoted as **JOINT**. **YouTube-VOS** [52]. YouTube-VOS is a large-scale benchmark for multi-object video segmentation which provides a much larger scale of training and test data than DAVIS. For the 2018 version, its validation set contains 474 videos, including 65 training (seen) categories and 26 unseen categories. And the 2019 version further augments the dataset with more video sequences, the number of videos in the val-

Table 3. State-of-the-art comparison on the YouTube-VOS [52] validation datasets. S denotes using synthetic data for offline training, FT denotes online fine-tuning. Our approach has superior generalization capability for unseen categories and outperforms all previous methods by considerable margins on both versions.

Validation 2018 Split							
Methods	S	FT	$\mathcal{J}_{\text{seen}}$	$\mathcal{F}_{\text{seen}}$	$\mathcal{J}_{\text{unseen}}$	$\mathcal{F}_{\text{unseen}}$	Overall
OnAVOS [42]	-	✓	60.1	62.7	46.1	51.7	55.2
OSVOS [6]	-	✓	59.8	60.5	54.2	60.7	58.8
PReMVOS [29]	✓	✓	71.4	75.9	56.5	63.7	66.9
SiamRCNN [43]	-	✓	73.5	-	66.2	-	73.2
STMVOS [33]	✓	-	79.7	84.2	72.8	80.9	79.4
EGMN [28]	✓	-	80.7	85.1	74.0	80.9	80.2
KMNVOS [37]	✓	-	81.4	85.6	75.3	83.3	81.4
S2S [51]	-	-	71.0	70.0	55.5	61.2	64.4
AGAME [20]	-	-	67.8	69.5	60.8	66.2	66.1
CFBI [54]	-	-	81.1	85.8	75.3	83.4	81.4
LWL [5]	-	-	80.4	84.9	76.4	84.4	81.5
CFBI ^{MS} [54]	-	-	82.2	86.8	76.9	85.0	82.7
JOINT (Ours)	-	-	81.5	85.9	78.7	86.5	83.1

Validation 2019 Split							
Methods	S	FT	$\mathcal{J}_{\text{seen}}$	$\mathcal{F}_{\text{seen}}$	$\mathcal{J}_{\text{unseen}}$	$\mathcal{F}_{\text{unseen}}$	Overall
STMVOS [33]	✓	-	79.6	83.6	73.0	80.6	79.2
LWL [5]	-	-	79.6	83.8	76.4	84.2	81.0
CFBI [54]	-	-	80.6	85.1	75.2	83.0	81.0
CFBI ^{MS} [54]	-	-	81.8	86.1	76.9	84.8	82.4
JOINT (Ours)	-	-	80.8	84.8	79.0	86.6	82.8

idation set is increased to 507. The unseen object categories make the YouTube-VOS much suitable for evaluating the generalization capability of algorithms.

We evaluate the proposed approach on both versions of the YouTube-VOS benchmark. As shown in Table 3, we compare our method with previously best-performing algorithms such as PReMVOS [29], STMVOS [33], EGMN [28], KMNVOS [37], CFBI [54], and LWL [5]. We can observe that our approach achieves average scores of 83.1% and 82.8% on the two versions of the validation set respectively, which outperform other state-of-the-art methods by a considerable margin. Moreover, we discover that the generalization capability of our approach is significantly better than previous algorithms. For the training (seen) categories, the matching-based methods like KMNVOS [37] and CFBI [54] perform well. However, when it comes to the performance on unseen categories, the previously mentioned methods plummet sharply, while our approach still maintains a relatively high level. Besides, our approach is superior to inductive learning-based methods such as LWL [5] thanks to the spatio-temporal consistency exploration of our proposed lightweight transformer.

DAVIS 2017 [35]. DAVIS is a popular video object segmentation benchmark. The validation set of DAVIS 2017 contains 30 densely annotated videos, and it is more challenging compared with DAVIS 2016 [34] since the multi-object setting is introduced. Follow [5], we split our ap-

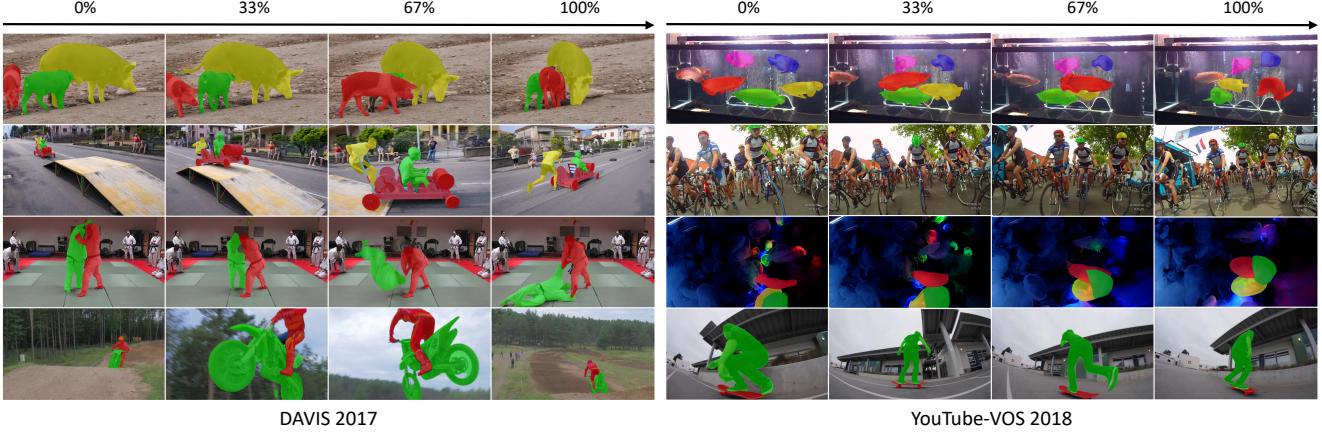


Figure 6. Qualitative result of our approach on DAVIS [35] and YouTube-VOS 2018 [52] validation sets. Our approach shows superior segmentation accuracy on both of them. In the first row, the pig in green and fish in yellow undergo severe occlusion. In the second row, similar distractors and cluttered background exist in the scene. The third and fourth rows are scenes with changes in appearance and perspective, respectively. Our approach successfully handles all these challenging scenarios.

Table 4. State-of-the-art comparation on the DAVIS 2017 [35] validation dataset. S denotes using synthetic data for offline training, FT denotes online fine-tuning, and t/s denotes second per frame. For fair comparison, we initialize the backbone ResNet-50 pre-trained on ImageNet [14] instead of the MaskRCNN [15] weights when only adopting DAVIS 2017 dataset for training.

Add YouTube-VOS for Training						
Methods	S	FT	\mathcal{J}	\mathcal{F}	$\mathcal{J} \& \mathcal{F}$	t/s
AGSSVOS [26]	✓	-	64.9	69.9	67.4	0.10
STMVOS [33]	✓	-	79.2	84.3	81.8	0.32
EGMN [28]	✓	-	80.2	85.2	82.8	0.40
KMNVOS [37]	✓	-	80.0	85.6	82.8	0.24
AGAME [20]	-	-	67.2	72.7	70.0	0.14
FEELVOS [41]	-	-	69.1	74.0	71.5	0.51
FRTM [36]	-	-	73.8	79.8	76.7	0.09
LWL [5]	-	-	79.1	84.1	81.6	0.15
CFBI [54]	-	-	79.1	84.6	81.9	0.17
CFBI ^{MS} [54]	-	-	80.5	86.0	83.3	9
JOINT (Ours)	-	-	80.8	86.2	83.5	0.25

Only DAVIS 2017 for training

Methods	S	FT	\mathcal{J}	\mathcal{F}	$\mathcal{J} \& \mathcal{F}$	t/s
OnAVOS [42]	-	✓	61.0	66.1	63.6	26
AGSSVOS [26]	✓	-	63.4	69.8	66.6	0.10
RGMP [32]	✓	-	64.8	68.6	66.7	0.28
STMVOS [33]	✓	-	69.2	74.0	71.6	0.32
KMNVOS [37]	✓	-	74.2	77.8	76.0	0.24
PReMVOS [29]	✓	✓	73.9	81.7	77.8	37.6
VideoMatch [18]	-	-	56.5	68.2	62.4	0.35
FRTM [36]	-	-	66.4	71.2	68.8	0.09
LWL [5]	-	-	72.2	76.3	74.3	0.15
CFBI [54]	-	-	72.1	77.7	74.9	0.17
JOINT (Ours)	-	-	76.0	81.2	78.6	0.25

proach into two versions depending on whether additional training data is employed or not and report their performance separately. We include the recently proposed LWL [5], CFBI [54], KMNVOS [37], and EGMN [28] for com-

parison. As shown in Table 4, when additionally adopting YouTube-VOS for training, our method exhibits the best performance with an average ($\mathcal{J} \& \mathcal{F}$) score of 83.5%, outperforming all previous approaches in the literature. Compared with CFBI^{MS} [54] (enhanced version of CFBI), our approach is free of multi-scale and flip strategy during evaluation, thus runs 30+ times faster.

When only adopting DAVIS 2017 for training, we initialize the backbone ResNet-50 with ImageNet [14] pretraining weights instead of the MaskRCNN [15] weights for fair comparison. As we can see, in this setup, our approach still outperforms all previous methods with a $\mathcal{J} \& \mathcal{F}$ score of 78.6%. Though the performance of PReMVOS [29] is close to ours, it relies on extensive online fine-tuning, so our approach runs about two orders of magnitude faster than it. Note that methods like STMVOS [33], KMNVOS [37], and PReMVOS [29] rely on additional synthetic data for pre-training. In contrast, our method is free of such necessity.

5. Conclusion

In this work, we design a novel architecture for semi-supervised video object segmentation, which takes advantage of both transductive reasoning and online inductive learning. To bridge the gaps between the two diverse models and better exploit their complementarity, we adopt a two-head label encoder to generate disentangled mask encodings as the carrier of target information. Extensive experiments show that the proposed approach sets several state-of-the-art records on prevalent VOS benchmarks without the need of simulated training data.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Contract 61822208, 61836011, and 62021001, and in part by the Youth Innovation Promotion Association CAS under Grant 2018497. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

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. 2
- [2] M. Berman, A. Triki, and Matthew B. Blaschko. The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018. 6
- [3] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019. 3, 4
- [4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 1, 2
- [5] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *ECCV*, 2020. 2, 3, 4, 5, 6, 7, 8
- [6] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 2, 7
- [7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2
- [8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 4
- [9] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 2
- [10] Jingchun Cheng, Yi Hsuan Tsai, Wei Chih Hung, Shengjin Wang, and Ming Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 2
- [11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, 2017. 2
- [12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: accurate tracking by overlap maximization. In *CVPR*, 2019. 1, 2, 3
- [13] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020. 2
- [14] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 8
- [15] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 6, 8
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6
- [17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4
- [18] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 8
- [19] Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 1
- [20] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, 2019. 7, 8
- [21] Anna Khoreva, Federico Perazzi, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2
- [22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [23] Zihang Lai, Erika Lu, and Weidi Xie. MAST: A memory-augmented self-supervised tracker. In *CVPR*, 2020. 1
- [24] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, 2018. 2
- [25] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *ECCV*, 2020. 2
- [26] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Agss-vos: Attention guided single-shot video object segmentation. In *ICCV*, 2019. 8
- [27] Yu Liu, Lingqiao Liu, Haokui Zhang, Hamid Rezatofighi, and Ian Reid. Meta learning with differentiable closed-form solver for fast video object segmentation. *arXiv preprint arXiv:1909.13046*, 2019. 2, 3
- [28] Xiankai Lu, Wenguan Wang, Danelljan Martin, Tianfei Zhou, Jianbing Shen, and Van Gool Luc. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020. 1, 2, 7, 8
- [29] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Pre-mvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018. 2, 7, 8
- [30] Alan Lukezic, Jiri Matas, and Matej Kristan. D3s - a discriminative single shot segmentation tracker. In *CVPR*, 2020. 2
- [31] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *TPAMI*, 41(6):1515–1530, 2019. 2
- [32] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 5, 8
- [33] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 1, 2, 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*, 2016. 6, 8
- [35] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2, 6, 7, 8
- [36] Andreas Robinson, Felix Järemo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast

- and robust target models for video object segmentation. In *CVPR*, 2020. 2, 3, 8
- [37] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020. 1, 2, 7, 8
- [38] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017. 2
- [39] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4
- [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3
- [41] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L. Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 2, 8
- [42] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 2, 7, 8
- [43] Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: visual tracking by re-detection. *arXiv preprint arXiv:1911.12836*, 2019. 7
- [44] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 1
- [45] Ning Wang, Wengang Zhou, and Houqiang Li. Contrastive transformation for self-supervised correspondence learning. In *AAAI*, 2021. 1
- [46] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, 2021. 2
- [47] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 2
- [48] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2
- [49] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, 2019. 2
- [50] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, 2018. 2
- [51] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 7
- [52] N. Xu, L. Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 2, 6, 7, 8
- [53] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 2
- [54] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020. 1, 2, 7, 8
- [55] Zhengjun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for fine-grained image captioning. *TPAMI*, 2019. 2
- [56] Lu Zhang, Zhe Lin, Jianming Zhang, Huchuan Lu, and You He. Fast video object segmentation via dynamic targeting network. In *ICCV*, 2019. 2
- [57] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *CVPR*, 2020. 2
- [58] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020. 2