Rethinking RNN-based Video Object Segmentation

Fatemeh Azimi[1,2], Federico Raue[2], Jörn Hees[2], and Andreas Dengel[1,2]

[1] TU Kaiserslautern
[2] German Research Center for Artificial Intelligence (DFKI)
{firstname.lastname}@dfki.de

Abstract Video Object Segmentation is a fundamental task in computer vision that aims at pixel-wise tracking of one or multiple foreground objects within a video sequence. This task is challenging due to real-world requirements such as handling unconstrained object and camera motion, occlusion, fast motion, and motion blur. Recently, methods utilizing RNNs have been successful in accurately and efficiently segmenting the target objects as RNNs can effectively memorize the object of interest and compute the spatiotemporal features which are useful in processing the visual sequential data. However, they have limitations such as lower segmentation accuracy in longer sequences. In this paper, we expand our previous work to develop a hybrid architecture that successfully eliminates some of these challenges by employing additional correspondence matching information, followed by extensively exploring the impact of various architectural designs. Our experiment results on YouTubeVOS dataset confirm the efficacy of our proposed architecture by obtaining an improvement of about 12pp on YoutTubeVOS compared to RNN-based baselines without a considerable increase in the computational costs.

Keywords: Video Object Segmentation · Recurrent Neural Networks · Correspondence Matching.

1 Introduction

One-shot Video Object Segmentation (VOS) is the task of densely tracking the intended foreground objects in a video, given the first mask of the object's appearance. VOS plays an important role in various applications such as video editing, autonomous driving, and robotics.

During the last years, a wide variety of learning-based solutions have been proposed for VOS trying to maximize the segmentation accuracy via addressing different challenging scenarios such as tracking smaller objects, handling occlusion, fast motion, crowded scenes with similar object instances, etc [45,47,1,4,49]. The suggested approaches in the literature can be roughly categorized to three main groups.

The first category naively tries to extend an image segmentation model to video domain [9,29]. During inference, these models try to adapt the trained

network to the specific scene and foreground object. This is usually done via further finetuning the network using the single object mask provided for the first frame (this process is known as online training). Therefore, these models are relatively slow and their performance is sub-optimal as training on a single image can result in overfitting behavior.

The second class deploys a memory component for memorizing the object of interest and processing the motion information [47,1,40]. Although using memory is a natural choice for processing the sequential data and these methods can achieve a good performance without requiring online training, their accuracy is limited by the functioning of the memory module. For example, their performance considerably drops for longer sequences due to the limited memory capacity and error propagation.

The third group is based on template matching [45,41,49]. These methods capture the target in each frame through finding the correspondences between the frame at hand and a reference frame (e.g. the given mask at $t = 0$). These approaches can also obtain a good performance with a fast run-time; however, their performance degrades in scenes with multiple similar object instances or when the object appearance changes drastically with respect to the reference frames. As it is expected, the model struggles to find the similarities in these scenario, resulting in low segmentation accuracy.

In this paper, we extend our previous work [2] that builds on top of a sequence-to-sequence (S2S) [47] baseline for VOS, due to its good performance and straightforward design and training procedure. The S2S architecture is an encoder-decoder network with an RNN module in the bottleneck which is responsible for processing the spatiotemporal features and tracking the target object. To improve the performance of this method in segmenting the longer sequences, we take inspiration from the matching-based algorithms [45]. We hypothesize that the matching-based methods complement S2S model by providing additional training signals that can enhance the segmentation accuracy and reduce the adverse effect of error propagation. Utilizing the reliable information in the reference frame can be especially useful in handling occluded scenes where the RNN may struggle to lose the target object after several time steps. To this end, we employ both RNN and matching branches and develop a fusion block to merge the RNN spatiotemporal features with the template matching and refer to our model as hybrid S2S (HS2S).

Additional to [2], we experiment with two architecture variants of our model. In the first form, we explore the effectiveness of bidirectional design [33] where in addition to utilizing the information from the past time steps, we integrate the future frames via a bidirectional RNN network. In the second variation, we explore a multi-task training setup by joint training the VOS model together with the unsupervised optical flow objective. Our intuition is that since optical flow and VOS are well-aligned tasks (in both cases the model has to learn pixel motion between the consecutive frames), training the model with both objectives might bring additional benefits via utilizing the optical flow-related constraints. We perform extensive experiments and ablations on YouTubeVOS dataset [47]

to study the role of different components in our HS2S model. Our experimental results confirm the effectiveness of our hybrid design by obtaining an increase of about 12pp in the overall segmentation accuracy compared to the RNN-based baseline.

2 Related Work

In this section, we provide a summary of the traditional variational methods based on energy minimization as well as more recent learning-based approaches proposed for solving the VOS task.

In [16,7,39], the authors attempt to solve foreground object estimation using supervoxels to capture similar regions across space and time. These methods cluster similar pixels across spatial dimensions into superpixel nodes and find the edges between these nodes across time and space by employing motion and appearance similarities to form the supervoxels. Accordingly, the object masks are obtained via processing and merging the connected supervoxels. In [8], Brox *et al.* develop a bottom-up approach for segmenting the foreground objects and utilize the optical flow motion information to enforce temporal consistency across a video shot. The main idea here is that pixels with similar motion patterns should belong to the same object. Similarly, Papazoglou *et al.* [25] propose a two-stage segmentation algorithm where in the first stage, the initial segmentation masks are obtained through processing the optical flow and motion boundaries. In the next stage, the masks are refined by applying two smoothness constraints. The first constraint enforces spatio-temporal consistency across video frames while the second implies that the foreground objects should only change smoothly over time. In [12], the authors suggest that optical flow only provides local information across neighboring frames which is not optimal. To address this limitation, they develop a model that integrates non-local information across space and time.

With [11,20,14] and the release of specialized and large-scale VOS datasets [47,30,29], the learning-based solutions for VOS have replaced more traditional models during the last few years. In [9], the authors present a training strategy to extend a network designed for medical image segmentation [22] for VOS. Starting from VGG16 [34] weights pretrained on ImageNet [11], they further train the network with segmentation objective on a VOS-specific dataset. During the inference, they perform online training and additionally fine-tune the network to be specialized for capturing the object of interest within the test scene. Perazzi *et al.* [28] also only rely on static images; to track the target object, they guide the network by inputting the object mask from the previous time step to the network. This method also needs online training for achieving acceptable performance. To address this limitation, [48] employs a modulator network that generates the normalization values of the main segmentation network, specific to each video.

Another line of work suggests utilizing RNNs for computing the spatio-temporal features, integrating the motion information, and memorizing the target object [47,38,1,40]. These methods achieve good performance without online

training, however, their segmentation accuracy worsens for longer sequences due to limited RNN memory and error propagation. This limitation is improved by incorporation of an external memory in [23]; however, this causes additional hardware memory constraints to the system. As a result, in practice, only a fraction of the frames can be stored in the memory which can be suboptimal.

Differently, Wug *et al.* suggest detecting the foreground object via finding the correspondences between each frame and reference frames using a Siamese architecture [45]. In [50], the authors propose a transductive approach that instead of only relying on a limited number of reference frames, additionally integrate the information from the past segmented frames while [17,6], develop a system that learns the appearance model of the object of interest and using this model, it captures the target throughout the rest of the video. These methods are efficient with a good performance, but their accuracy degrades in the presence of multiple similar objects as the model is confused by finding multiple correspondences. To improve this challenge, [49] additionally incorporates background correspondence matching. They demonstrate this design helps the model to better handle ambiguities in the correspondence search.

3 Method

In this part, we describe our proposed architecture based on a hybrid propagation policy, referred to as HS2S. Our model builds on top of S2S [47], a sequence-to-sequence model for video object segmentation. Following a detailed study on the performance of the S2S model, we address multiple shortcomings of the S2S design by inserting additional information obtained from correspondence matching.

The S2S model consists of an encoder-decoder network similar to U-Net [31] that learns the mapping between the RGB color space and the object segmentation mask. S2S uses a ConvLSTM [46] module between the encoder and the decoder, intended for processing the spatiotemporal features and tracking the target object. This memory component is accountable for maintaining the temporal coherency between the predicted segmentation masks across several video frames. To this end, S2S utilizes an *initializer* network that generates the initial ConvLSTM hidden states through processing the first RGB frame and foreground object mask. The S2S model can be summarized as follows [47]:

$$h_0, c_0 = \text{Initializer}(x_0, y_0) \tag{1}$$

$$\tilde{x}_t = \text{Encoder}(x_t) \tag{2}$$

$$h_t, c_t = \text{RNN}(\tilde{x}_t, h_{t-1}, c_{t-1}) \tag{3}$$

$$\tilde{y}_t = \text{Decoder}(h_t) \tag{4}$$

where h and c are the hidden and cell states for the ConvLSTM, t is the time step, x is the RGB input, y is the ground-truth mask and $\tilde{y}$ is the predicted output.

Having a closer look at the failure cases in the S2S model, we observed the model's performance degrades for longer videos. There are multiple factors that can potentially contribute to this limitation. First, the limited memory of RNNs is an inherent challenge for RNN-based architectures. Due to this issue, the model struggles to fully capture the essential information in the scene as well as the evolution of the object's appearance. Moreover, as the video sequences become longer, they tend to lose access to the information from the earlier time steps. This is particularly problematic for the occluded scenes; as, if the model forgets the occluded object, it will not be able to re-capture the object once it re-appears in the scene. Finally, due to the feedback connection in the RNN module, the erroneous predictions will flow to the future time steps. These results in drift and error propagation in the final results, exacerbating the model's accuracy for segmenting the frames further in time.

3.1 Hybrid Sequence-to-Sequence VOS.

We propose to supplement RNN module with *correspondence matching* for tracking the object of interest in a video. The benefits of matching-based solutions [45] for one-shot VOS gives the opportunity to solve some problems inherent in RNNs. For example, Oh *et al.* design a siamese architecture that obtains a good segmentation accuracy through matching with reference frames at $t = 0$ and the guidance information from the previous mask [45]. However, these models struggle in scenes with similar objects or when the object's appearance changes drastically over time as the model cannot detect the object via matching to the reference frames anymore.

We hypothesize that the pros and cons of the RNN-based and matching-based VOS solutions complement each other. The informative signals computed from template matching are crucial for better handling the occluded videos; no matter how long the occlusion duration, the model would still have the chance to re-capture the object through matching it with the reference frames at $t = 0$. Moreover, these additional training signals reduce the adverse effect of error propagation thus improving the overall segmentation quality. On the other hand, integrating the motion features and the spatiotemporal model learned by RNNs can serve as prior for the approximate object location at time step t. This combination helps the model to disambiguate the situations with similar objects by employing the location prior.

Our proposed architecture is shown in Figure 1. We utilize one encoder for the input frames and another encoder for the reference frames and masks. As suggested by previous works [28,45], we utilize the time steps 0 and $t-1$ as our reference frames. Time step 0 is specifically important since the ground-truth segmentation mask for this frame is available at inference; therefore, the information from this step is highly reliable for the model. Moreover, integrating

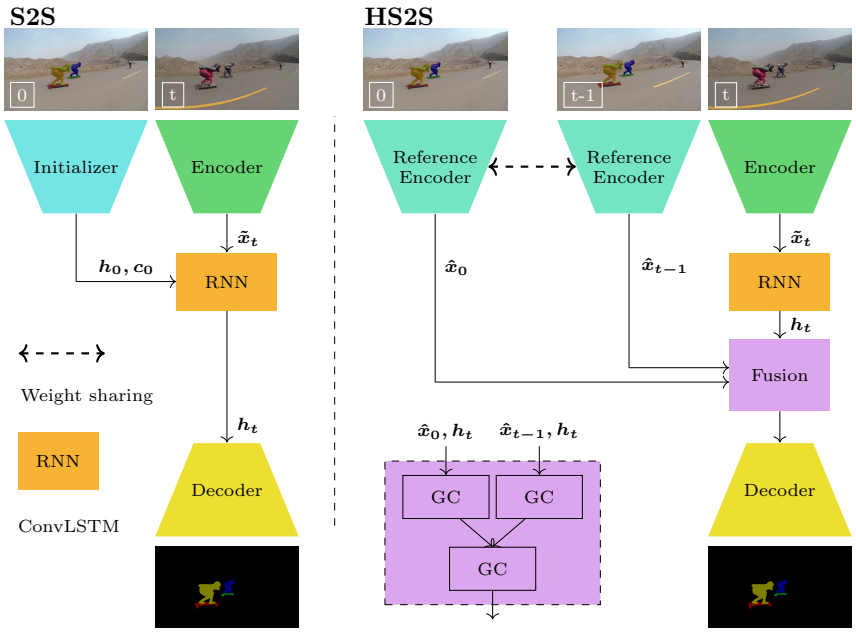

Figure 1: The HS2S architecture combining the RNN-based and matching based features for VOS. We utilize a fusion block consisting of multiple Global Convolution (GC) [26] layers that merges the spatiotemporal features from RNN with the reference frames from the first and the previous time steps.

the information from $t-1$ serves as an additional signal about the approximate object location.

To reduce the model complexity, we replace the initializer network with a teacher-forcing training strategy [28]. Having the initializer network removed, we simply initialize the memory hidden states with zero vectors. Next, we feed the segmentation mask from the previous time step as an additional input to the encoder [28]. By receiving the previous segmentation mask as input, the model is informed about the approximate location of the target object.

In the next step, we merge the reference and the spatiotemporal RNN features through a nonlinear fusion function. To properly combine the information from these to branches, this module requires both local and global connections across the spatial feature dimensions. Accordingly, one could design this module using convolution layers with very big kernel sizes [26], or utilize attention mechanisms to span the height and width dimensions and incorporate all the features [43,44]. Ablation on the impact of various designs for merge block architecture is provided in Section 4.5.

Finally, the output of the merge block is then passed through a decoder network and transformed into the segmentation masks through a stack of upsampling and convolution layers. With the same notation as in eqs. (1) to (2) and (4), the overall steps can be summarized into the formulation below [2]:

$$h_0, c_0 = \mathbf{0} \tag{5}$$

$$\hat{x}_0 = \text{Reference\_Encoder}(x_0, y_0) \tag{6}$$

$$\hat{x}_{t-1} = \text{Reference\_Encoder}(x_{t-1}, y_{t-1}) \tag{7}$$

$$\tilde{x}_t = \text{Encoder}(x_t, y_{t-1}) \tag{8}$$

$$h_t, c_t = \text{RNN}(\tilde{x}_t, h_{t-1}, c_{t-1}) \tag{9}$$

$$\tilde{y} = \text{Decoder}(\tilde{x}_0, \tilde{x}_{t-1}, h) \tag{10}$$

As the architectures for encoder and reference encoder in Figure 1 are identical, we experimented with weight-sharing between the two encoders; however, this architecture resulted in lower segmentation accuracy. This behavior can be due to two reasons: First, reduction in model's expressive power, and second, the misalignment between the reference features and the teacher-forcing training strategy. As we can see in Figure 1, the RGB frame and mask fed to the ref-encoder are from the same time step while the input to the encoder differs by 1 ($t-1$ and t). Therefore, the functions learned by these two encoders are not the same and that weight sharing would not be applicable to this scenario.

Training Loss. We train our model with a combination of Balanced BCE loss and an additional auxiliary term of border classification [2]:

$$L_{\text{total}} = \lambda \; L_{\text{seg}} + (1 - \lambda) \; L_{\text{aux}} \tag{11}$$

The BCE term assigns either foreground or background label to each pixel in the image. As in the image, the portion of background pixels usually outweighs the foreground, the training will be biased towards paying higher attention to the background. This issue is addressed by multiplying the loss terms with a balancing factor [9]:

$$L_{\text{seg}}(\mathbf{W}) = \sum_{t=1}^{T} (-\alpha \sum_{j \in Y_+} \log P(y_j = 1 | X; \mathbf{W}) - (1 - \alpha) \sum_{j \in Y_-} \log P(y_j = 0 | X; \mathbf{W})) \tag{12}$$

with α being the ratio of background to foreground pixels. The Border classification objective additionally classifies each pixel based on their relative distance to the object border. As a result, this term provides finer information about the pixel location and improves the quality of detected object edges. Further explanation and an in-depth analysis of this loss-term's effects can be found in [1].

3.2 Bidirectional Architecture

In HS2S architecture, the video frames are processed sequentially passing the information from the past to the future. Bidirectional sequence-to-sequence architectures enable the model to integrate information from the past as well as the future; they have been effective in improving the performance of sequential processing tasks such as Machine Translation [33,36,38,13]. Therefore, it is natural to conjecture that integrating the information from the future frames might benefit the HS2S model. However, based on the task definition in VOS, we need the object mask in the last frame ($t = T$) to process the video backward in time. Otherwise, the model will not recognize which object to track. To address this challenge, we design the bidirectional HS2S (Bi-HS2S) architecture shown in Figure 2. As explained earlier, there are two different ways to inform the net-

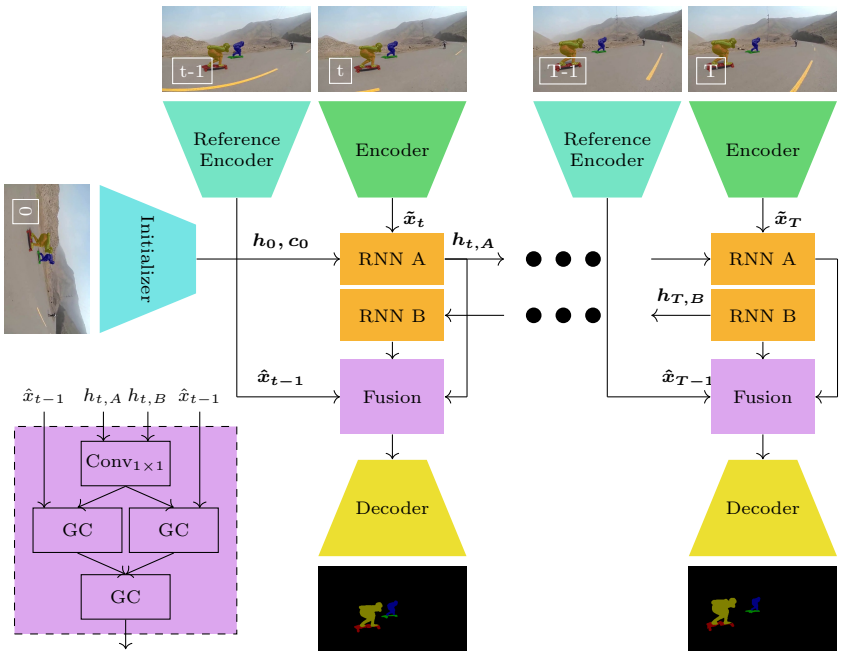

Figure 2: The Bidirectional HS2S architecture. The hidden states from the forward and backward RNNs are combined using a convolution layer and then merged with the reference features and passed to the decoder.

work about the object of interest. One is through using an initializer network that processes the first RGB and the mask frames and initializes the memory hidden states. The second way is by simply feeding the segmentation mask from

the previous time step to the encoder network as a guidance signal. The second option does not fit the bidirectional design as in the backward processing of the video sequence, we do not have access to the initial object mask. As a result, we resort to the first alternative.

As illustrated in Figure 2, we initialize the memory hidden states with the initializer network in the forward path. For the backward processing, we simply initialize the backward memory with the last hidden state h_t obtained from the forward path. Our intuition is that h_t contains information about the target object at $t = T$ and can serve as a reasonable initialization. Finally, we combine the information from the forward and backward paths together with the reference features via the fusion block and pass it to the decoder to predict the segmentation masks.

3.3 Multi-task Training with Optical Flow Prediction

In multi-task learning, several tasks are combined within a single problem formulation and network architecture. This approach has been shown to be a successful training technique when the combined tasks are aligned in the objective and can provide supplemental information to each other [32]. In this section, we take inspiration from RAFT [37], a recent state-of-the-art optical flow architecture and design an architecture that combines video object segmentation with optical flow prediction, referred to as RAFT-HS2S. Our intuition is that VOS and optical flow objectives are similar as they both tend to learn the pixel movement from one frame to the next. Accordingly, we explore whether combining these two learning objectives brings additional information to the model and enhances the segmentation accuracy.

RAFT model [37] receives two consecutive images (I_t and I_{t-1}) as input and generates the flow field capturing the pixel motion between the consecutive frames. It consists of two encoders with the same architecture but separate weights; The first encoder extracts the features f_t and f_{t-1} while the second encoder only processes I_{t-1} to provide additional context to the network. Inspired by traditional optical methods, RAFT iteratively refines the estimated flow utilizing a ConvGRU [35] that produces the flow delta at each time step.

Motivated by the commonality in VOS and optical flow training objectives, we adapt HS2S to accommodate the RAFT components as depicted in Figure 3. As can be seen in this plot, the reference encoder additionally takes the role of context encoding for the RAFT model, and the Encoder is employed for processing I_t and I_{t-1}. For the optical flow loss which is added to the objective in Equation 11, we use an unsupervised objective, namely photometric loss [18]:

$$L_{photometric} = \sum |I^{(1)} - w(I^{(2)})| \tag{13}$$

Here $L_{photometric}$ is the photometric loss over all the pixels and w is the warping operation that warps $I^{(2)}$ to $I^{(1)}$ using the optical flow between these two frames. This term implies that having the precise motion, the pixel colors resulting from warping one image to the other should match.

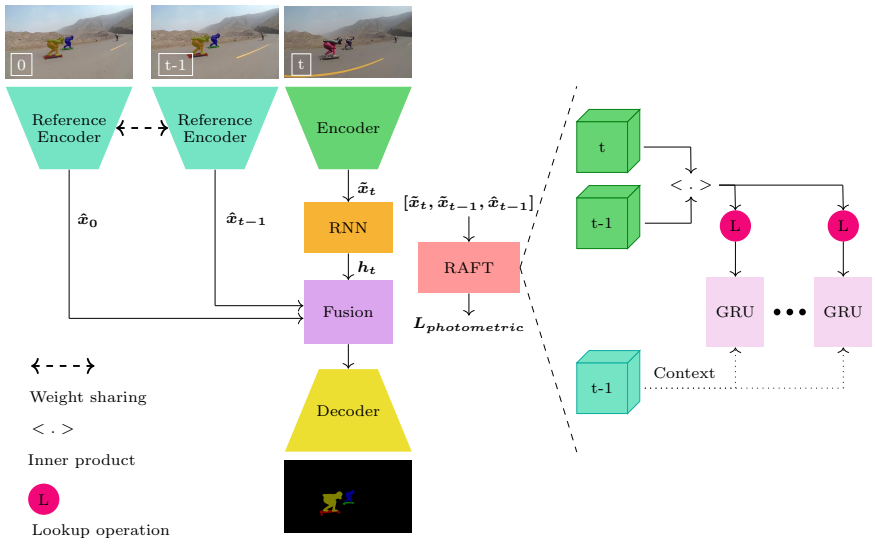

Figure 3: The multi-task training setup in RAFT-HS2S, combining HS2S with an optical flow method named RAFT. RAFT module computes the correlation between frames at t and $t-1$ using the inner product between the respective feature vectors and generates an initial estimate of the optical flow between these consecutive frames. Then, it iteratively refines the approximated flow using a ConvGRU module that performs lookup operations based on the correlation volume and a context feature vector computed from the frame at $t-1$.

4 Experiments

4.1 Implementation Details

In this section, we explain the implementation and the training details of our HS2S model explained in Section 3.1. We use the same experimental setup and hyperparameters for the other architecture variants unless mentioned otherwise. Additional ablations are provided in Sections 4.3 to 4.5.

The encoder backbone in Figure 1 is based on ResNet50 [14] architecture pretrained on ImageNet [11], with the following modifications. We remove the final fully connected layer which generates the image classification output and add $conv_{1\times1}$ to reduce the number of channels in the bottleneck from 2048 to 1024. In the first layer, we add a convolution layer to process the segmentation mask and combine it with the RGB features.

For the memory module, we use a ConvLSTM layer [46] with 3×3 filters as suggested in [47]. The merge module consists of a stack of two Global Convolution layers [26] with a kernel size of 7×7, following the setup in [45].

The decoder consists of 5 upsampling layers followed by convolution layers with 5×5 kernel sizes and $1024, 512, 256, 128, 64$ number of channels respectively. The last layer activation function is a sigmoid nonlinearity that outputs the probability of each pixel belonging to the foreground or background. Moreover, we utilize skip connections [31] and skip-memory connections [1] for obtaining refined segmentation masks and better tracking the smaller objects.

As mentioned in Section 3.1, we use a teacher-forcing training strategy, feeding the ground-truth segmentation masks from time step $t-1$ as input in time t. Since we do not have access to the ground-truth labels during the inference, the masks predicted by the model are used instead. As pointed out by [5,3], this training approach is problematic since the model predictions are often erroneous, and the accumulation of errors result in a gap between the training and testing phases. Following the recipe suggested in [5], we deploy a curriculum learning policy to address this issue. At the beginning of the training when the model does not generate high-quality masks, we use the ground-truth labels; once the training loss is stable, we follow a probabilistic scheme to decide whether to choose from the ground-truth or use the model prediction as input. The probability of selecting the model predictions is gradually increased from 0 to 0.5.

We use a batch size of 16 and Adam [19] optimizer with a starting learning rate of $1e-4$. Once the training loss is stabilized, the learning rate is reduced every 5 epochs by a factor of 0.9.

4.2 Experimental Results

We assess our method on YouTubeVOS [47], the largest dataset for VOS consisting of $3,471$ and 474 videos in training and validation sets respectively. We report F and J scores, the standard metric for evaluating VOS models [27]. YouTubeVOS evaluation additionally reports *seen* and *unseen* scores to separately measure the model's accuracy for the objects that have been present or absent during the training. The *unseen* scores quantify the model generalization to new object types.

In Table 1, we present the results obtained from HS2S model (plus its variants as explained in Sections 3.1 to 3.2) as well as our baseline S2S, and the other state-of-the-art models. The results provided in the upper half of the table are for the methods with additional online training. Using the first object mask, these approaches further train the network at test time; as a result, they often achieve a better accuracy but they are considerably slower. As can be seen from the results in Table 1, our HS2S method reaches a significant improvement in comparison with the S2S baseline and outperforms this approach even when it is fine-tuned by additional online training (S2S(OL)). Moreover, we observe that utilizing the Bi-HS2S leads to further improvement of about 1pp while RAFT-HS2S achieves similar performance as HS2S. This implies that the information provided from the optical flow loss is already included in the VOS objective and combining these additional terms does not bring additional benefits to the model.

Method	OL	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}	Overall
OSVOS [21]	✓	59.8	54.2	60.5	60.7	58.8
MaskTrack [28]	✓	59.9	45.0	59.5	47.9	50.6
OnAVOS [42]	✓	60.1	46.6	62.7	51.4	55.2
S2S(OL) [47]	✓	**71.0**	**55.5**	**70.0**	**61.2**	**64.4**
OSMN [48]	✗	60.0	40.6	60.1	44.0	51.2
RGMP [45]	✗	59.5	45.2	-	-	53.8
RVOS [40]	✗	63.6	45.5	67.2	51.0	56.8
A-GAME [17]	✗	66.9	61.2	-	-	66.1
S2S(no-OL) [47]	✗	66.7	48.2	65.5	50.3	57.7
S2S++ [1]	✗	68.7	48.9	72.0	54.4	61.0
STM- [23]	✗	67.1	63	69.4	71.6	68.2
TVOS [50]	✗	-	-	-	-	67.2
HS2S [2]	✗	73.6	58.5	77.4	66.0	68.9
Bi-HS2S	✗	**74.9**	**59.6**	**78.0**	**66.7**	**69.8**
RAFT-HS2S	✗	73.2	58.7	77.3	66.1	68.8

Table 1: Comparison of the experimental results from the HS2S model [2] with state-of-the-art methods on YouTubeVOS dataset. OL refers to methods with additional Online Training.

In Figure 4, we provide a qualitative comparison between the segmentation results from S2S [47] baseline and our HS2S model. As can be seen, our model can successfully maintain the segmentation accuracy for later time steps. Moreover, our method can handle scenes with multiple similar objects, which is challenging for matching-based algorithms.

To quantify the impact of our hybrid architecture on the accuracy of longer videos, we select a subset of sequences from YouTubeVOS training set that consist of more than 20 frames for evaluation and use the rest for training. We resort to using this subset since the ground-truth masks for the YoutubeVOS validation set are not provided. Then, we calculate the segmentation accuracy for earlier frames ($t < 10$) and later frames ($t > 10$) independently. As can be seen in Table 2a, These two models have similar performance for earlier frames, while HS2S significantly outperforms the S2S model for frames in the further time steps. Additionally, we experiment with our HS2S model when using either 0th or $(t-1)$th frame as a reference in order to assess the role of each one in the final performance. The results for this experiment are provided in Table 2b.

4.3 Ablation on the Impact of Encoder Architecture

The encoder networks in Figure 1 are responsible for extracting descriptive features which will be then processed through the memory and decoded into a segmentation mask via the decoder network. Thus, improving the quality of the encoder network is expected to directly reflect on the segmentation quality. In this section, we study the behavior of HS2S model when employing various

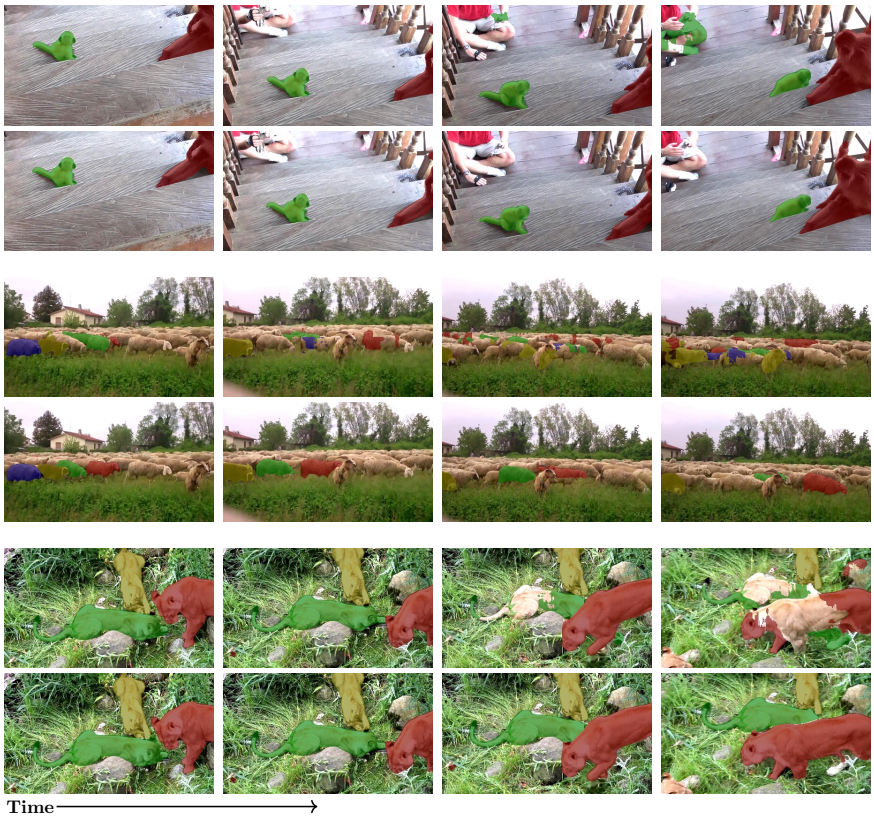

Figure 4: Visual Comparison between the S2S and HS2S results in the upper and lower rows, respectively. We observe that our hybrid method can successfully maintain the segmentation accuracy at the later time steps.

encoder architectures including VGG [34], ResNet50 [14], DeepLab [10], and Axial-DeepLab [43].

The DeepLab backbone is a modified ResNet50 with less pooling operations resulting in increased spatial feature dimension at the output (Higher spatial dimensions are presumably beneficial for dense prediction methods due to preserving fine local information). Furthermore, it consists of an Atrous Spatial Pyramid Pooling (ASPP) module composed of a stack of convolution layers with various dilation rates. We experiment with a DeepLab backbone pretrained on image segmentation, with and without the ASPP module. Different than CNN-based backbones, Axial-DeepLab [43] consists of fully-attentional blocks. In this model, the authors propose to break the attention into horizontal and vertical attentions to reduce the computational cost of the attention-based backbone (from quadratic to linear). However, this model still requires significantly higher

Method	$F_{l<10}$	$J_{l<10}$	$F_{l>20}$	$J_{l>20}$
S2S*	74.4	73.7	54.5	54.6
HS2S (ours)	**77.1**	**76.3**	**65.5**	**64.2**

(a) Performance comparison considering shorter ($t < 10$) and longer ($t > 10$) sequences [2].

Method	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}
HS2S_0	72.6	55.4	76.7	61.2
HS2S_{t-1}	72.2	55.1	76.1	61.3

(b) Performance of HS2S model when either the first or previous frame is used as reference [2].

Table 2: **(a)** Comparison of our model with S2S baseline for shorter and longer sequences (S2S* is our implementation of this model with ResNet50 backbone which achieves about 1pp higher accuracy compared to [47]). We observe that our method significantly improves the performance for frames in later time steps. **(b)** The performance of HS2S model when either the first frame is used as reference ($\text{HS2S}_{0,t-1}$ shows the accuracy of our model when either 0_{th} or the $t-1_{th}$ frames are used as reference). We observe that both frames contribute to boosting the segmentation accuracy and the best results are obtained when using both reference frames.

memory compared to the CNN-based backbones. Due to memory limitation, we experimented with *small* Axial-DeepLab architecture as elaborated in [43]

Backbone	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}	*overall*
VGG16 [34]	71.4	56.0	74.9	64.8	66.8
ResNet-50 [14]	73.6	58.5	77.4	66.0	68.9
ResNet-101 [14]	73.9	58.5	77.3	66.2	69.0
ResNet-50-DeepLab (without ASPP) [10]	73.3	59.1	77.2	66.0	68.9
ResNet-50-DeepLab (w/ ASPP) [10]	72.3	56.9	76.5	64.2	67.4
Axial [43]	70.5	55.0	73.1	61.8	65.1

Table 3: An ablation on the impact of backbone network.

As can be seen in Table 3, we obtained the best results when applying the ResNet-based encoder. Surprisingly, integrating the additional ASPP module from the DeepLab architecture resulted in lower performance. This behavior can be due to the added complexity from combining the spatiotemporal RNN features with multi-scale processing in the ASPP module resulting in a more challenging optimization problem and sub-optimal performance.

4.4 Ablation on the Impact of RNN Architecture

One of the main blocks in the HS2S architecture is the RNN block, accountable for memorizing the target object. In this section, we provide an ablation studying the HS2S performance when deploying three different RNN-based memories.

The first variant is ConvLSTM [46]. This module is developed for processing sequential visual data by replacing the fully connected layers in LSTM with convolution layers, adjusting the LSTM layer for visual pattern recognition.

As the second model, we study DeepRNN [24]. In this model, the authors address the challenges in training deep RNN models. Although deeper networks are expected to learn better representations compared to their shallow counterparts in the case of CNNs, deep RNN architectures designed by simply stacking the RNN layers does not lead to considerable improvement in the model accuracy. In [24], Pang *et al.* suggest this behavior roots in the complex optimization operation when dealing with RNNs. Processing the entangled spatial and temporal information in sequential visual data leads to the optimization process becoming overly complex. This condition could become even more extreme for deeper RNNs, resulting in sub-optimal performance. To this end, they propose to disentangle the information related to the spatial flow from the temporal flow. They design a *Context bridge module (CBM)* which is composed of two computing blocks for processing the representation and the temporal flows. By enforcing these sources of information to flow independently, the optimization process could potentially be simplified. In our experiments, we deployed a stack of 5 RNN layers following the setup proposed in [24].

In the third variant, TensorLSTM [35], the authors attempt to improve the learning of long-term spatiotemporal correspondences for processing longer videos. They design a higher-order convolutional LSTM architecture named TensorLSTM that can better capture extended correlations. TensorLSTM consists of a preprocessing and a convolutional tensor-train module. The preprocessing module computes feature vectors from multiple overlapping sliding windows from the previous hidden states. These embeddings are then further processed through the convolutional tensor-train module and passed to the LSTM. Consequently, they are able to efficiently integrate the information from the previous hidden states and improve the capturing of long-term correlations.

Method	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}	$overall$
ConvLSTM [46]	73.6	58.5	77.4	66.0	68.9
DeepRNN [24]	72.4	57.3	75.8	64.9	67.6
Tensor-TrainLSTM [35]	**74.7**	**60.2**	**78.5**	**66.4**	**70.0**

Table 4: An ablation on the choice of RNN module.

As it can be seen from the results in Table 4, TensorLSTM achieves a better segmentation accuracy compared to the other variants. This implies that in HS2S architecture, we do not require deeper RNNs to carry the information about the object of interest. However, accessing the information from multiple frames over an extended time period is beneficial for the model. In a way, TensorLSTM applies attention to a limited past context via the sliding-window mechanism in the preprocessing module. This observation is in line with employing the dual

propagation strategy in Section 3.1 where simply merging the information from the time-step $t-1$ improves the segmentation results.

4.5 Ablation on Various Designs for the Fusion Block

In this section, we study the model's performance when working with three different fusion block architectures in Table 5. Intuitively, the fusion block needs to provide global connections across the spatial dimensions as the object might be displaced to a further location compared to the reference frames. Additionally, it has to assign higher attention to the locations that belong to the foreground. In HS2S$_{sim}$, the spatiotemporal RNN features are merged with the reference features based on cosine similarity. HS2S$_{GC}$ merges these two branches using global convolution layers as suggested in [26] while HS2S$_{attn}$ replaces this operation with an attention layer [15]. We obtained similar performance for different fusion architecture options, but the design using attention attained the highest accuracy.

Method	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}	*overall*
HS2S$_{GC}$	73.6	58.5	77.4	66	68.9
HS2S$_{sim}$	72.3	56.4	76.2	62.5	66.9
HS2S$_{attn}$	**73.9**	**58.7**	**77.5**	**66.3**	**69.1**

Table 5: An ablation on the impact of backbone network.

5 Conclusion

In this paper, we expanded our previous HS2S approach [2] to generate a hybrid architecture for VOS that combines the merits of RNN-based and matching-based methods. As before, we find that all of our hybrid approaches are especially beneficial for the segmentation of objects that are occluded and reappearing, as well as in cases where many similar objects need to be tracked and segmented. In this paper, we have investigated two derived architectures: a bi-directional and a multi-task (optical flow) extension of our previous approach. Further, we expanded our previous ablation study by investigating further segmentation backbones, RNN, and fusion blocks of the underlying architectures. Resulting from these investigations, we found that our bi-directional extension (Bi-HS2S) improves over our previous architectures by nearly 1 pp and more than 12pp when compared to other state-of-the-art RNN-based baselines (such as S2S(no-OL) [47]). To our surprise, our multi-task extension also taking optical flow into account (RAFT-HS2S) failed to improve over HS2S. In the expanded ablation study, we found that the ResNet-101 based backbone network, TensorTrainLSTM RNN architecture, and attention fusion blocks seemed to be the most beneficial design choices.

For future work, we aim to investigate the potential benefit of utilizing depth information either as input or as an unsupervised learning objective integrated into our HS2S model.

6 Acknowledgement

This work was supported by the TU Kaiserslautern CS PhD scholarship program, the BMBF project ExplAINN (01IS19074), and the NVIDIA AI Lab (NVAIL) program. Further, we thank Christiano Gava, Stanislav Frolov, Tewodros Habtegebrial and Mohammad Reza Yousefi for the many interesting discussions and proofreading of this paper. Finally, we thank all members of the Deep Learning Competence Center at the DFKI for their feedback and support.

References

1. Azimi, F., Bischke, B., Palacio, S., Raue, F., Hees, J., Dengel, A.: Revisiting sequence-to-sequence video object segmentation with multi-task loss and skip-memory. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 5376–5383. IEEE (2021), arXiv:2004.12170
2. Azimi, F., Frolov, S., Raue, F., Hees, J., Dengel, A.: Hybrid-s2s: Video object segmentation with recurrent networks and correspondence matching. In: VISAPP. pp. 182–192 (2021), arXiv:2010.05069
3. Azimi, F., Nies, J.F.J.N., Palacio, S., Raue, F., Hees, J., Dengel, A.: Spatial transformer networks for curriculum learning. arXiv preprint arXiv:2108.09696 (2021)
4. Azimi, F., Palacio, S., Raue, F., Hees, J., Bertinetto, L., Dengel, A.: Self-supervised test-time adaptation on video data. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3439–3448 (2022)
5. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems. pp. 1171–1179 (2015)
6. Bhat, G., Lawin, F.J., Danelljan, M., Robinson, A., Felsberg, M., Van Gool, L., Timofte, R.: Learning what to learn for video object segmentation. In: European Conference on Computer Vision. pp. 777–794. Springer (2020)
7. Brendel, W., Amer, M., Todorovic, S.: Multiobject tracking as maximum weight independent set. In: CVPR 2011. pp. 1273–1280. IEEE (2011)
8. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: European conference on computer vision. pp. 282–295. Springer (2010)
9. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 221–230 (2017)
10. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Faktor, A., Irani, M.: Video segmentation by non-local consensus voting. In: BMVC. p. 8 (2014)

13. Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional lstm networks for improved phoneme classification and recognition. In: International conference on artificial neural networks. pp. 799–804. Springer (2005)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multidimensional transformers. arXiv preprint arXiv:1912.12180 (2019)
16. Jain, S.D., Grauman, K.: Supervoxel-consistent foreground propagation in video. In: European conference on computer vision. pp. 656–671. Springer (2014)
17. Johnander, J., Danelljan, M., Brissman, E., Khan, F.S., Felsberg, M.: A generative appearance model for end-to-end video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8953–8962 (2019)
18. Jonschkowski, R., Stone, A., Barron, J.T., Gordon, A., Konolige, K., Angelova, A.: What matters in unsupervised optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 557–572. Springer (2020)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
21. Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: Video object segmentation without temporal information. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2018)
22. Maninis, K.K., Pont-Tuset, J., Arbeláez, P., Van Gool, L.: Deep retinal image understanding. In: International conference on medical image computing and computer-assisted intervention. pp. 140–148. Springer (2016)
23. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9226–9235 (2019)
24. Pang, B., Zha, K., Cao, H., Shi, C., Lu, C.: Deep rnn framework for visual sequential applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 423–432 (2019)
25. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1777–1784 (2013)
26. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters–improve semantic segmentation by global convolutional network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4353–4361 (2017)
27. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Computer Vision and Pattern Recognition (2016)
28. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2663–2672 (2017)
29. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 724–732 (2016)

30. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)
31. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
32. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
33. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE transactions on Signal Processing **45**(11), 2673–2681 (1997)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
35. Su, J., Byeon, W., Kossaifi, J., Huang, F., Kautz, J., Anandkumar, A.: Convolutional tensor-train lstm for spatio-temporal learning. arXiv preprint arXiv:2002.09131 (2020)
36. Sundermeyer, M., Alkhouli, T., Wuebker, J., Ney, H.: Translation modeling with bidirectional recurrent neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 14–25 (2014)
37. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
38. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4481–4490 (2017)
39. Vazquez-Reina, A., Avidan, S., Pfister, H., Miller, E.: Multiple hypothesis video segmentation from superpixel flows. In: European conference on Computer vision. pp. 268–281. Springer (2010)
40. Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., Giro-i Nieto, X.: Rvos: End-to-end recurrent network for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5277–5286 (2019)
41. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: Feelvos: Fast end-to-end embedding learning for video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9481–9490 (2019)
42. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. arXiv preprint arXiv:1706.09364 (2017)
43. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: European Conference on Computer Vision. pp. 108–126. Springer (2020)
44. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
45. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7376–7385 (2018)
46. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015)
47. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018)

48. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6499–6507 (2018)
49. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by foreground-background integration. In: European Conference on Computer Vision. pp. 332–348. Springer (2020)
50. Zhang, Y., Wu, Z., Peng, H., Lin, S.: A transductive approach for video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6949–6958 (2020)