Context Adaptation for Video Object Segmentation*

Isidore Dubuisson[1], Damien Muselet[1], Christophe Ducottet[1], and Jochen Lang[2]

[1] Université Jean Monnet Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
isidore.dubuisson@univ-st-etienne.fr
[2] School of Electrical Engineering and Computer Science University of Ottawa, OTTAWA, Canada

Abstract. In this paper we present an adaptation module for feature matching based Semi-automatic Video Object Segmentation methods (SVOS). Most current solutions to adapt SVOS methods during inference are slow or inefficient. Feature matching based methods use affinity between a set of reference and query features to segment a target in the current frame based on a reference. We propose an adaptation module working solely with the user supplied mask in the first frame of a video. Our adaptation of the matching module provides more reliable information to the model for segmentation in all the video frames and does not significantly increase inference time. We demonstrate in our results on OVIS and DAVIS that our solution adapts the feature space to ensure that query features match with the appropriate features from the reference.

Keywords: Video Segmentation · Feature matching · First frame adaptation · Context-Aware

1 Introduction

Semi-automatic Video object Segmentation (SVOS) is a specific task in computer vision where in the first frame the user selects the object to track in the video. This selection is provided to the system as a binary mask that is automatically propagated to the next frames in order to segment the selected object. Since the selection is provided by the user, it does not necessarily belong to a trained class and it can cover only part of an object. This kind of selection makes it a very challenging task since the trained model has to provide and exploit generic semantic features to tackle the diversity of the videos and, at the same time, it has to leverage specific features that can discriminate the selected object from the background. However, the current solutions [18, 12] provide semantic features that are representing the training data and are not optimized for the specific context of a new video.

* Supported by FRANCE CANADA RESEARCH FUND.

This problem is illustrated in Fig. 1 where the first frame of a video is shown with the object selected by the user (mask in green). In the scene, we can see that several puppies look very similar while a single dark puppy has been selected by the user. With the classical solution, the generic features learned on a large dataset cannot differentiate between the puppies even if one is in the selected mask and the others are outside this mask. Consequently, the mask is propagated over all the dark puppies in the given example.

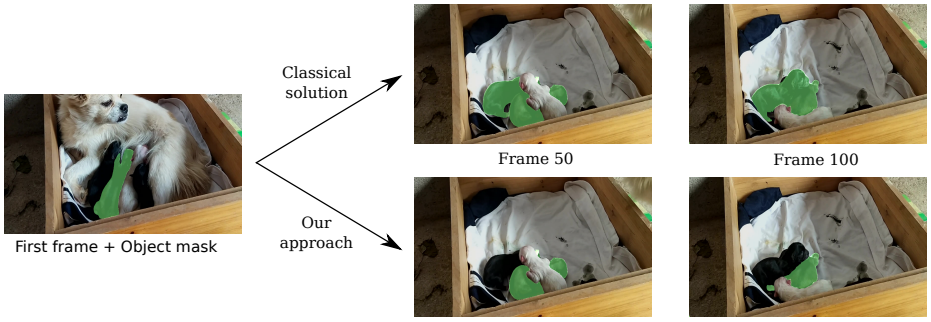


Fig. 1. Segmentation results with and without adaptation to the specific context. The provided mask and its propagation are displayed in green.

The default solution in this case consists of finetuning the deep network on the new data before testing it. But, this requires annotated data for the new context as well as extra time for finetuning the pre-trained network. In this paper, we propose a light solution to adapt a SVOS network to a specific context by leveraging the only annotation available in this context, i.e., the object selected by the user in the first frame. The second row of Fig. 1 shows that the proposed adaptation step provides features that can differentiate between the various dark puppies and hence avoids the over-propagation of the mask.

Obviously, exploiting a single frame to fine-tune a network can easily lead to overfitting because of the small amount of annotated data. Hence, the adaptation step has to be carefully conducted. Recent SVOS solutions [18, 12] store features of the first frames into a memory and leverage these initial features to enrich features of the current frame. Each feature vector is represented by a key and a value. The keys are used for comparison between features while the values are the new features which will be sent to the decoder for reconstructing the segmentation mask. We argue that the keys have two properties that make them good candidates for adaptation. First, they play a crucial role in the selection of the best values. Second, changing the keys does not require to adapt the decoder since the keys do not feed directly into it. Furthermore, the crucial role of the key encoder has been emphasized very recently in [21]. Note that fine-tuning the decoder is not feasible with the small amount of available annotated data in the first frame.

Consequently, in this paper, we contribute a solution to adapt the key vectors of a SVOS network by only using the first frame and the selected object. This adaptation is very simple, fast and we show that it helps to extract features that are specific to each video context and that provide better segmentation results than the generic features provided by the classical solutions. Next, we will briefly review recent semi-automatic VOS methods including applicable finetuning approaches.

2 Related Work

Video object segmentation (VOS) can be fully unsupervised [22] if the objects to be segmented are automatically detected. In our case, we concentrate on the semi-supervised case, where in the first frame the user selects the object to segment over the video [5]. We focus our discussion of related work on SVOS and more specifically, online training.

2.1 Semi-Automatic Video Object Segmentation

Work on SVOS follows two main solution approaches: mask propagation and feature matching. When using mask propagation, the approaches exploit the mask of the previous frame (starting from the ground-truth mask, in the first frame) and propagate it to the current frame [11, 7, 3]. These approaches take advantage of the fact that the motion is smooth between successive frames. They can suffer from propagation drift along the video. In the case of feature matching, the solutions consist of matching features from the current frame with features from previous frames where masks are available [2, 6]. The matched features are then used to predict the mask in the current frame. When the reference features are only extracted from the first frame, the results are not robust to variations of object appearance across frames. This problem can be tackled by methods that store the features of successive frames in a memory [12, 18, 4, 3]. In our solution, we make use of the efficient SWIFTNET [18] that updates the memory only in case of large inter-frame variations and avoids feature redundancies in the memory by storing only the areas that exhibit the most severe feature variations.

2.2 Online training

Various methods [7, 2, 10] leverage the ground-truth of the mask provided in the first frame of a video. In these methods, heavy data augmentation on the first frame is used to fine-tune the whole network. However, this finetuning step is inefficient and causes long delays before the network can predict the segmentation in the subsequent frames of the video. Consequently, Robinson et al. [16] proposed a deep architecture with two complementary networks: one light-weight network that can be trained fast on a single first frame and that only provides a coarse segmentation mask, and a second heavy network that is trained off-line and not fine-tuned. Li et al. resort to a cyclic mechanism that mitigates the

error propagation [20]. Their idea is to check online that the segmentation provided at the current frame agrees with the segmentation in the previous frames, and especially in the first frame because of the available ground truth. Hu et al. [6] exploit the provided ground truth mask of the first frame to obtain a set of foreground and background features and use them as references to classify the features extracted from the current frame as foreground or background. The authors claim that this specific process is general enough such that the fine-tuning step can be omitted. Since fine-tuning on only the first frame can lead to over-fitting, Voigtlaender and Leibe [13] propose to update the network online by selecting training examples from the test frames. Training the network online is time-consuming and not suitable for real-time segmentation. Finally, when it comes to optimizing initialization weights and hyper-parameters for a subsequent fine-tuning, meta-learning is applicable [17, 1].

We take inspiration from Li et al. [8] who select a subset of features among the ones provided by the backbone in order to boost the efficiency of their tracker. This solution exploits only the first frame of a video and employs novel losses in order to select important features. But unlike all prior works, we design a novel loss to efficiently adapt only a small but crucial part of a pre-trained model. Our approach allows to use any available pre-trained feature matching based network and to efficiently fine-tune it. We do not require a new complex architecture or learning process.

3 Our Approach

3.1 Method Overview

Our approach relies on memory based feature matching SVOS architectures such as STM [12] or SWIFTNET [18]. The principle of these architectures is presented in Fig. 2. The main branch (query branch) is a standard segmentation oriented encoder-decoder built upon a feature extraction backbone. Given a query frame, the query encoder Enc_Q is computing a $H \times W \times C$ feature map where H and W are reduced dimensions of the query frame and C is the feature dimension. The second branch (reference branch) is composed of a sibling reference encoder Enc_R associated with a memory module. The purpose of this branch is to encode and store information extracted from past frames and their corresponding segmentation masks. At least the first frame with its reference mask is encoded and stored into the memory. During the processing of the video, additional frames or individual feature vectors may be added to the memory depending on the update memory strategy. For clarity, they are not depicted in Fig. 2..

The key component of the architecture is an affinity module operating at the bottleneck of the query branch. The idea is to enrich information of the query frame with relevant information from the memory to drive the segmentation. The affinity module is a non-local module using a key-value encoding principle. Specific query and reference key-value encoders are added at the end of the initial query and reference encoders to compute keys K_Q and values V_Q, and keys K_R and values V_R, respectively. Keys are used to match features between query

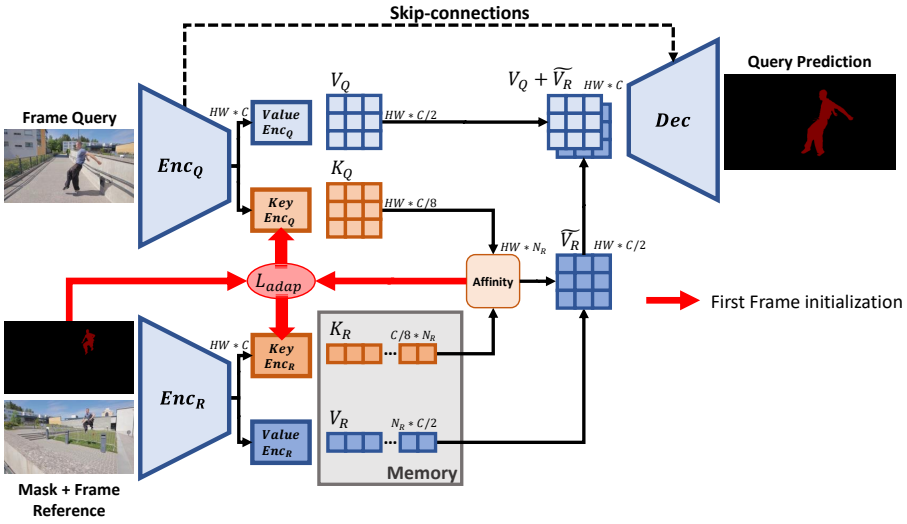


Fig. 2. Proposed SVOS architecture with our key adaptation module

and reference and values are used to encode relevant information for segmenting objects. The dimension of key and value spaces are reduced versions of the initial feature space. Typically, the dimension of the key space is $C/8$ and those of the value space is $C/2$. For each query frame, the affinity module computes the affinity of the query keys K_Q with all the reference keys K_R stored in the memory (N_R feature vectors) and generates a composite value $\widetilde{V}_R$ computed as a linear combination of reference values V_R weighted by softmax affinity scores.

The principle of our method is to add a light adaptation module to both the query and the reference key encoders. Our intuition is that key encoders play a crucial role in the selection of the best values and if they are not adapted to the current context, incorrect selection can lead to errors in the final output. Moreover, adapting key encoders is a light adaptation which does not require to further adapt other parts of the architecture. Practically, the adaptation step is made independently for each video using only the first frame and the associated reference mask. During this step, the first frame is encoded with both, the query and the reference branch and the two key encoding modules are fine-tuned through the minimization of a specific adaptation loss L_{adapt} built upon the result of the affinity calculation. Once the two key encoders are adapted on the first frame, they are applied to subsequently process the complete video. Before presenting the adaptation loss, we focus on the affinity module next.

3.2 In-depth Analysis of the Affinity Module

The affinity module takes as input both query and reference keys flattened in their spatial dimension. In the first frame, the set of key query feature vectors

is $K_Q = (K_Q^i)_{i\in\{1,\dots,HW\}}$ with $K_Q^i \in \mathbb{R}^{C/8}$ and the set of key reference feature vectors is $K_R = (K_R^j)_{j\in\{1,\dots,N_R\}}$ with $K_Q^j \in \mathbb{R}^{C/8}$. An affinity matrix of term A_{ij} is computed as the softmax over j of the dot product between K_Q^i and K_R^j [12]. Thus we have:

$$A_{ij} = \frac{1}{Z_i} \exp(K_Q^i \circ K_R^j) \tag{1}$$

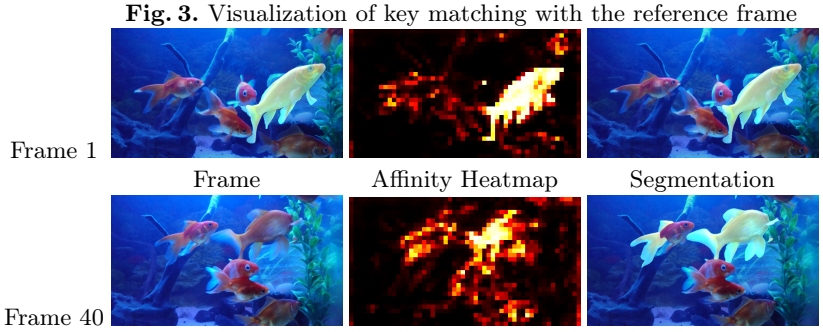
where $\circ$ is the dot product and $Z_i = \sum_j \exp(K_Q^i \circ K_R^j)$ is the softmax normalization. Then the enriched value $\widetilde{V}_Q^i$ associated to feature i of the query vector is obtained by:

$$\widetilde{V}_Q^i = [V_Q^i, \sum_j A_{ij} V_R^j] \tag{2}$$

where $[.,.]$ denotes the concatenation.

Figure 3 presents a visualization of the affinity matrix obtained while processing the *gold-fish* video from DAVIS (cf. Section 4.3). The goal is to segment the largest fish highlighted in green in frame 1 with a fixed memory initialized with frame 1. Affinity heatmaps are obtained by computing for each query patch i its total affinity with all memory patches belonging to the object in the memory. The corresponding heatmap value is thus $h_i = \sum_{j\in\text{object}} A_{ij}$. In frame 1, high values in the heatmap are mostly concentrated at the actual fish location. However, some background pixels belonging to other fish also contain large values but do not result in segmentation errors. In frame 40, more background pixels on other fish have even large values in the heatmap causing an incorrect segmentation result.

Fig. 3. Visualization of key matching with the reference frame



3.3 Adaptation Loss

Figure 3 illustrates the imperfect key encoding already visible in the processing of frame 1: Some background pixels in the query are similar to the object in memory. A reversed analysis could show that some object pixels in the query

may be similar to some background pixels in the memory. The purpose of our adaptation loss is to minimize the sum of affinity values over regions where this affinity should be null. If m_i (resp. m_j) denotes a flattened version of the query object mask (resp. reference object mask), we define object and backgound losses as:

$$\begin{aligned} L_{obj} &= \textstyle\sum_i (1 - m_i) \sum_j m_j A_{ij} \\ L_{bg} &= \textstyle\sum_i m_i \sum_j (1 - m_j) A_{ij} \end{aligned} \tag{3}$$

The total loss is defined as $L_{\text{adapt}} = \alpha L_{obj} + L_{bg}$ where α is adjusted to ensure that the two losses start from the same value at the beginning of the training. It avoids that one of the loss dominates the other which can cause the dominated loss to increase, even though the overall loss decreases. Reducing this total loss helps avoid confusing object features with background features.

Note that because of the reduced spatial image resolution in the encoder, the mask is downscaled and mask values are floats.

4 Experiments and Results

Our main goal is to evaluate the performance when a single object is selected by the user, hence our evaluation is for segmenting a single object in the OVIS dataset [15]. We regard the OVIS dataset appropriate for evaluating the adaptation given the high number of similar objects in each video and the way the objects strongly interact amongst themselves. As the OVIS dataset was not initially designed for SVOS, the first frame annotation is not provided in the test and evaluation set. Hence, we generate the annotation and evaluate on 25 videos (149 objects) randomly selected from the OVIS training set. Results for every annotated object that appears in the first frame are reported in Section 4.2. Annotated objects that appear only in later frames are ignored. Single object evaluation will allow us to study the duality between object and background. For completeness, we will also report multi-object results on DAVIS in Section 4.3.

4.1 Offline Training and Adaptation during Inference

We follow the original authors in training SWIFTNET [18] by pretraining on COCO [9] and finetuning on YOUTUBE-VOS [19] and DAVIS17 [14].

During inference every layer is frozen except the single convolutional layer of the two key encoders. During inference, an adaptation is performed at every first frame initialization to fine-tune the key encoders. The finetuning is performed for 50 epochs with a learning rate of 1. At the end of the finetuning on the first frame, the weights of the key encoders are frozen during the segmentation of the remaining frames of the video. Because no layers are added to the original model, except during initialization in the first frame, there is no impact on run time of our adaptation during the video. Hence, we retain the fast inference speed of SWIFTNET. The average adaptation time on OVIS is only 0.47s for a complete video and hence it has a negligible impact on real-time applications.

Table 1. Jaccard index for single object segmentation in OVIS (training)

Adaptation	Static Memory	Memory Update
-	39.7	40.81
✓	43.85	44.62

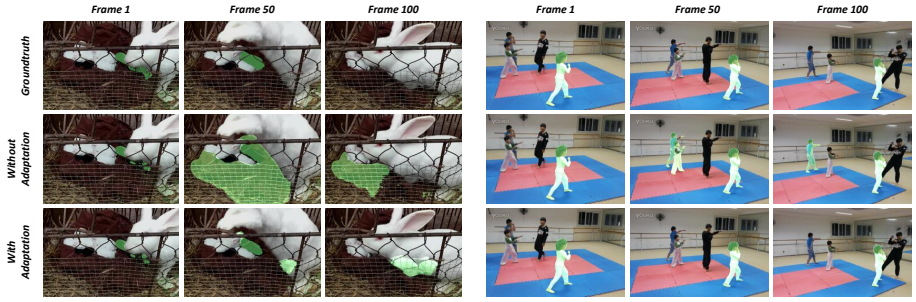


Fig. 4. Visual comparisons of the results with and with-out the adaptation of the keys at different frames. Both videos are from OVIS

4.2 Results for Single Object Segmentation

We report results for the evaluation of our adaptation using the Jaccard index. As shown in Table 1, the adaptation increases the Jaccard index independent of the use of memory updates. The improvement shows that our adaptation can help to bridge the gap between training and testing data as OVIS data was not used in offline training. Our adaptation is also able to handle distracting objects with similar appearance as they are common in OVIS. Fig. 4 shows a comparison of qualitative results. The object of interest is still segmented in frame 50 and 100 in all cases but the adaptation reduces the incorrect propagation of the mask to the background. Note that the memory update is still improving the result with and without adaptation. The memory update is compatible with the adaptation even if the adaptation is processed only on the first frame. This also shows that the feature space learned in the first frame is still suitable throughout the video.

4.3 Multi Object Extension

For completeness, we also report results on multi-object segmentation on DAVIS 2017 in Table 2. To extend our adaptation to multiple objects, we compute the loss for each object in turn, considering the corresponding single foreground mask separately. In the end, we sum all losses together and calculate a single adaptation for the shared encoder. As expected, our first frame adaptation also improves the results for multi-object segmentation on DAVIS. The improvement is smaller than for single object segmentation on OVIS (see Section 4.2) because the model was trained partially on DAVIS and hence the domain gap is smaller to start with. Also, DAVIS does not contain video samples of many very similar objects where accurate keys are essential. Table 2 contains results for

SWIFTNET reported by the original authors. As the weights and training code are not publicly available, we provide results of our own training following the description by Wang et al. [18]. Our own training weights are the baseline for our adaptation.

Table 2. Results on Multi-Object Segmentation in DAVIS 2017. F is the boundary accuracy. Online Learning (OL) Mask-Propagation (MP) and Feature Matching (FM). Results come from original articles.

name	OL		$J\&F$	J	F	FPS
OnAVOS[13]	✓	–	67.9	64.5	70.5	0.08
OSVOS[10]	✓	–	68.0	64.7	71.3	0.22
RGMP[11]	×	MP	66.7	64.8	68.9	7.7
SAT[3]	×	MP	72.3	68.6	76.0	39
STM[12]	×	FM	81.8	79.2	84.3	6.3
SWIFTNET[18]	×	FM	81.1	78.3	83.9	25
XMem[18]	×	FM	87.7	84.0	91.4.9	22.6
SWIFTNET (our training)	×	FM	78.01	75.39	80.62	25
with adaptation	✓	FM	79.02	76.4	81.64	25

5 Conclusions and Future Work

We have proposed a fast and light adaptation method that can be used in any matching-based SVOS method. We have obtained encouraging results on the challenging OVIS dataset for the task of segmenting a single object selected by the user. We have also shown on DAVIS2017 that even on videos closely related to the training, our adaptation improves segmentation results. In the future, we would like to extend our efficient adaptation method to additional frames in the video without requiring additional groundtruth masks.

References

1. 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: Proceedings of the CVF European Conference on Computer Vision (2020)
2. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., , Gool, L.V.: One-shot video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017)
3. Chen, X., Li, Z., Yuan, Y., Yu, G., Shen, J., Qi, D.: State-aware tracker for real-time video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
4. Cheng, H.K., Schwing, A.G.: Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In: Proceedings of the CVF European Conference on Computer Vision (2022)
5. Gao, M., Zheng, F., Yu, J.J., Shan, C., Ding, G., Han, J.: Deep learning for video object segmentation: A review. Artificial Intelligence Review (2022)

6. Hu, Y.T., Huang, J.B., Schwing, A.: Videomatch: Matching based video object segmentation. In: Proceedings of the CVF European Conference on Computer Vision (2018)
7. Khoreva, A., Benenson, R., Ilg, E., Brox, T., Schiele, B.: Lucid data dreaming for video object segmentation. International Journal of Computer Vision **127**(9), 1175–1197 (2019)
8. Li, X., Ma, C., Wu, B., He, Z., Yang, M.H.: Target-aware deep tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
9. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context. In: Proceedings of the CVF European Conference on Computer Vision (2014)
10. Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Gool, L.V.: Video object segmentation without temporal information. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
11. Oh, S.W., Lee, J.Y., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2018)
12. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
13. Paul, V., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: Proceedings of the British Machine Vision Conference (2017)
14. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Gool, L.V.: The 2017 davis challenge on video object segmentation. In: arXiv preprint arXiv:1704.00675 (2017)
15. Qi, J., Gao, Y., Hu, Y., Wang, X., Liu, X., Bai, X., Belongie, S., Yuille, A., Torr, P., Bai, S.: Occluded video instance segmentation: A benchmark. International Journal of Computer Vision (2022)
16. Robinson, A., Lawin, F.J., Danelljan, M., Khan, F.S., Felsberg, M.: Learning fast and robust target models for video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
17. Tim, M., Leal-Taixé, L.: Make one-shot video object segmentation efficient again. In: Proceedings of Advances in Neural Information Processing Systems (2020)
18. Wang, H., Jiang, X., Ren, H., Hu, Y., Bai, S.: Swiftnet: Real-time video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
19. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: Proceedings of the CVF European Conference on Computer Vision (2018)
20. Yuxi, L., Ning, X., Jinlong, P., John, S., Weiyao, L.: Delving into the cyclic mechanism in semi-supervised video object segmentation. In: Proceedings of Advances in Neural Information Processing Systems (2020)
21. Zhang, Y., Li, L., Wang, W., Xie, R., Song, L., Zhang, W.: Boosting video object segmentation via space-time correspondence learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
22. Zhou, T., Li, J., Li, X., Shao, L.: Target-aware object discovery and association for unsupervised video multi-object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)