Modular Interactive Video Object Segmentation: Interaction-to-Mask, Propagation and Difference-Aware Fusion

Ho Kei Cheng
UIUC/HKUST
hokeikc2@illinois.edu

Yu-Wing Tai
Kuaishou Technology
yuwing@gmail.com

Chi-Keung Tang
HKUST
cktang@cs.ust.hk

Abstract

We present Modular interactive VOS (MiVOS) framework which decouples interaction-to-mask and mask propagation, allowing for higher generalizability and better performance. Trained separately, the interaction module converts user interactions to an object mask, which is then temporally propagated by our propagation module using a novel top-k filtering strategy in reading the space-time memory. To effectively take the user's intent into account, a novel difference-aware module is proposed to learn how to properly fuse the masks before and after each interaction, which are aligned with the target frames by employing the space-time memory. We evaluate our method both qualitatively and quantitatively with different forms of user interactions (e.g., scribbles, clicks) on DAVIS to show that our method outperforms current state-of-the-art algorithms while requiring fewer frame interactions, with the additional advantage in generalizing to different types of user interactions. We contribute a large-scale synthetic VOS dataset with pixel-accurate segmentation of 4.8M frames to accompany our source codes to facilitate future research.

1. Introduction

Video object segmentation (VOS) aims to produce high-quality segmentation of a target object instance across an input video sequence, which has wide applications in video understanding and editing. Existing VOS methods can be categorized by the types of user input: semi-supervised methods require pixel-wise annotation of the first frame, while interactive VOS approaches take user interactions (e.g., scribbles or clicks) as input where users can iteratively refine the results until satisfaction.

This paper focuses on interactive VOS (iVOS) which finds more applications in video editing, because typical user interactions such as scribbles or clicks (a few seconds per frame) are much easier than specifying full annotation (∼79 seconds per instance), with the iterative or successive refinement scheme allowing the user more control over result accuracy versus interaction budget trade-off [1].

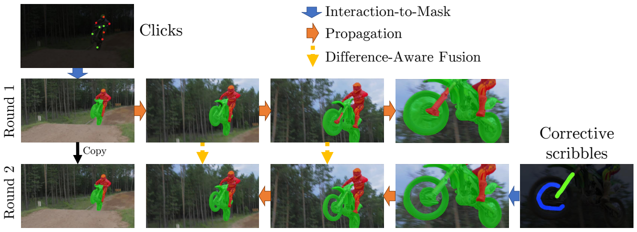

Figure 1. User annotates one of the frames (e.g., with clicks at the top-left frame) and MiVOS bidirectionally propagates the masks to the entire video sequence. Our difference-aware fusion module guides the segmentation network to correct the masks across frames based on user's intended correction on another frame (e.g., with scribbles on the bottom-right frame).

Conceptually, iVOS can be considered as the combination of two tasks: interaction understanding (e.g., mask generation from interactions [2, 3, 4, 5]) and temporal propagation (e.g., semi-supervised VOS methods [6, 7, 8]). Current methods usually perform the two tasks jointly, using interconnected encoders [9, 10, 11] or memory-augmented interaction features [12, 13, 14]. The strong coupling limits the form of user interaction (e.g., scribbles only) and makes training difficult. Attempts to decouple the two tasks fail to reach state-of-the-art accuracy [15, 16] as user's intent cannot be adequately taken into account in the propagation process.

One advantage of unified methods over decoupled methods is that the former can efficiently pick up small corrective interactions across many frames, which is suited to the DAVIS evaluation robot [1]. However, we believe that human users tend to interactively correct a single frame to high accuracy before checking other frames, as the visual examination itself takes time and human labor while free for an evaluation robot. Our method requires less interacted frames by letting the user focus on a single frame multiple times while attaining the same or even better accuracy. Our method is efficient as single-frame interaction can be done almost instantly [4], with the more time-consuming propagation performed only sparsely.

Source code, pretrained models and dataset are available at: https://hkchengrex.github.io/MiVOS. This research is supported in part by Kuaishou Technology and the Research Grant Council of the Hong Kong SAR under grant no. 16201420.

In this paper we present a decoupled modular framework to address the iVOS problem. Note that naïve decoupling may lead to loss of user's intent as the original interaction is no longer available in the propagation stage. This problem is circumvented by our new difference-aware fusion module which models the difference in the mask before and after each interaction to inject the user's intent in propagation. Thus the user's intent is preserved and propagated to the rest of the video sequence. We argue that *mask difference* is a better representation than raw interactions which is unambiguous and does not depend on interaction types. With our decoupling approach, our method can accept different types of user interactions and achieve better performance on various qualitative and quantitative evaluations. Our main contributions can be summarized as follows:

- We innovate on the decoupled interaction-propagation framework and show that this approach is simple, effective, and generalizable.
- We propose a novel lightweight top-k filtering scheme for the attention-based memory read operation in mask generation during propagation.
- We propose a novel difference-aware fusion module to faithfully capture the user's intent which improves iVOS accuracy and reduces the amount of user interaction. We will show how to efficiently align the masks before and after an interaction at the target frames by using the space-time memory in propagation.
- We contribute a large-scale synthetic VOS dataset with 4.8M frames to accompany our source codes to facilitate future research.

2. Related Works

Figure 2 positions our MiVOS with other related works in interactive image/video object segmentation.

Semi-Supervised Video Object Segmentation. This task aims to segment a specific object throughout a video given only a fully-annotated mask in the first frame. Early methods often employ test-time finetuning on the given frame [8, 17, 18, 19, 6, 20] to improve the model's discriminatory power, but such finetuning is often too slow. Recently, diverse approaches have been explored including pixel-wise embedding [21, 22, 23], mask propagation and tracking [6, 24, 25, 26, 27, 28, 29, 30, 31], building a target model [32], and memory features matching [33, 7, 34, 12, 35, 36, 37]. In particular, STM [7] constructs a memory bank from past frames and predicts the mask using a query-key-value attention mechanism. While simple and effective, this method can achieve state-of-the-art results. In this work, we propose to transfer the technical progress of semi-supervised VOS methods to the interactive domain. Our space-time memory network, which is inspired by STM [7], is used in our propagation backbone.

Interactive Video Object Segmentation (iVOS). User-supplied hints are provided in iVOS. The interactions can be used to either segment an object or a correct previously misclassified region [38, 39, 40, 1]. Most recent works [11, 9, 12] have focused on scribble interaction which is used and provided by the DAVIS challenge [41]. A recent method [22] has extended their embedding network in the interactive setting with clicks as user input. Our method can generalize to a wide range of user interactions due to the modular design by simply replacing the interaction-to-mask component.

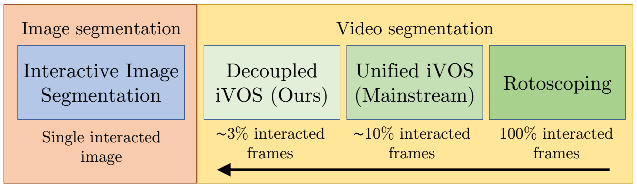

Figure 2. Progress in iVOS [41] has significantly reduced the amount of human labor required to segment objects in videos compared with traditional rotoscoping methods. By leveraging more spatially dense yet temporally sparse interactions, our method further reduces the human effort required to examine the output video in a more tedious, back-and-forth manner (see Section 6.3 for user study) while reaching the same or even better accuracy. Our method can be regarded as lifting 2D image segmentation to 3D.

The majority of current deep learning based iVOS methods is based on deep feature fusion to incorporate user interactions into the segmentation task, where two interconnected encoder networks are designed [9, 10, 11], or scribble features are stored as memory which are referenced later in the segmentation process [12, 13, 14]. These approaches inevitably tie the particular form of user inputs with the mask propagation process. This property makes training difficult as the model needs to adapt to both understanding the interactions and accurately propagating masks at the same time. Alternatively, some methods have attempted to decouple the interaction and propagation network [15, 16] by first generating a mask given an interaction in any types, followed by propagating this mask bidirectionally. But these methods fail to achieve state-of-the-art performance. We believe that this is due to the dismissal of user intent as the propagation network no longer has access to the original user interaction.

This paper proposes to overcome the above problem by considering the difference in the mask domain before and after an interaction round in order to directly and faithfully represent the user intent in the propagation process.

Interactive Image Segmentation. The problem of interactive image segmentation or cutout has a long history with a wide range of applications [42, 43, 44, 2]. The recent adoption of deep convolutional neural network has greatly improved state-of-the-art performance with different types of user interactions such as bounding boxes [3], clicks [45, 4, 4], or extreme points [5, 46]. Our modular approach can adapt to any of these types of interactions by adopting the corresponding interaction-to-mask algorithm in our framework.

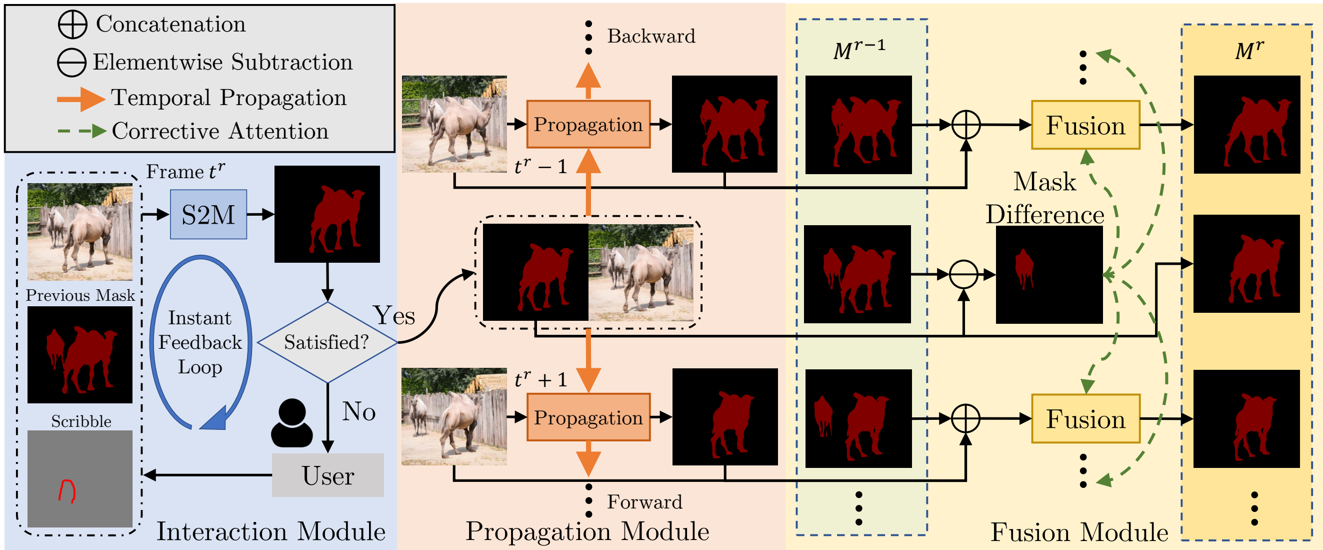

Figure 3. **MiNet** overview. In interaction round r, the user picks a frame t' and interactively correct the object mask until satisfaction using the Scribble-to-Mask (S2M) module (Section 3.2) running in real time. The corrected mask will then be bidirectionally propagated through the video sequence with the propagation module (Section 3.3). To incorporate information from previous rounds, a difference-aware fusion module is used to fuse previous and current masks. The difference in the interacted mask before and after the interaction (which conveys user's intention) is used in the fusion module via an attention mechanism (Section 3.4). In the first round, all masks are initialized to zeros.

3. Method

Initially, the user selects and interactively annotates one frame (e.g., using scribbles or clicks) to produce a mask. Our method then generates segmentation for every frame in the video sequence. After that, the user examines the output quality, and if needed, starts a new "round" by correcting an erroneous frame with further interactions. We denote r as the current interaction round. Using superscript, the user-interacted frame index in the r-th round is t^r, and the mask results of the r-th round is M^r; using subscript, the mask of individual j-th frame is denoted as M^r_j. Refer to supplementary material for a quick index of the paper's notations.

3.1. MiNet Overview

As illustrated in Figure 3, our method consists of three core components: interaction-to-mask, mask propagation, and difference-aware fusion. The interaction module operates in an instant feedback loop, allowing the user to obtain real-time feedback and achieve a satisfactory result on a single frame before the more time-consuming propagation process[1]. In the propagation module, the corrected mask is bidirectionally propagated independently of M^{r-1}. Finally, the propagated masks are fused with M^{r-1} with the fusion module which aims to fuse the two sequences while avoiding possible decay or loss of user's intent. The user intent is captured using the difference in the selected mask before and after user interaction. This difference is fed into the fusion module as guidance.

3.2. Interaction-to-Mask

Various interactive image segmentation methods can be used here as long as they can compute an object mask from user interactions. Users are free to use their favorite segmentation tool or even tailored pipeline for specific tasks (e.g., human segmentation for movie editing). Methods that use information from an existing mask ($M^{r-1}_{t^r}$) might be more labor-efficient but such property is optional.

We design a Scribble-to-Mask (S2M) network to evaluate our method on the DAVIS [41] benchmark. Our pipeline has high versatility not restricted by any one type of such interaction network – we additionally employ click-based interaction [4], freehand drawing, and a local control module that allows fine adjustment which are experimented in the user study Section 6.3.

S2M The goal of the S2M network is to produce a single-image segmentation in real time given input scribbles. Our design is intentionally straightforward with a standard DeepLabV3+ [47] semantic segmentation network as the backbone. The network takes a six-channel input: RGB image, existing mask, and positive/negative scribble maps, and deals with two cases: initial interaction (where the existing mask is empty) and corrective interaction (where the existing mask contains error). Unlike previous methods [14, 9, 11], we train with a simpler single-round approach on a large collection of static images [48, 49, 50, 51]. We are able to leverage these non-video large datasets by the virtue of our decoupled paradigm.

For each input image, we randomly pick one of the two cases (with an empirically set probability of 0.5) and syn-

[1] To the best of our knowledge, most related state-of-the-art works take $>$ 100ms per frame, with current "fast" methods taking $>$ 15ms per frame for propagation. This justifies our single-frame interaction and propagation where the latter runs at $\sim$ 100ms per frame

thesize the corresponding input mask which is either set to zeros or perturbed from the ground-truth with random dilation/erosion [52]. We do not reuse the output mask to form a second training stage [14, 9, 11] to reduce training cost and complications. Input scribbles are then generated correspondingly in the error regions using strategies [41] such as thinning or random Bézier curves.

Local Control While state-of-the-art interactive segmentation methods such as f-BRS [4] often use a large receptive field to enable fast segmentation with few clicks, it may harm the global result when only local fine adjustment is needed toward the end of the segmentation process. Figure 4 illustrates one such case where the global shape is correct except for the ears. With our decoupled approach, it is straightforward to assert local control by limiting the interactive algorithm to apply in a user-specified region as shown in the figure. The region's result can be effortlessly stitched back to the main segmentation.

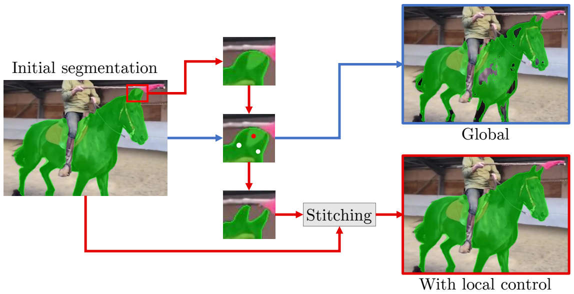

Figure 4. The local control pathway (red) uses an ROI to prevent deterioration spread by the global interaction path (blue) when only a small local refinement (around ears) is needed.

3.3. Temporal Propagation

Given an object mask, the propagation module tracks the object and produces corresponding masks in subsequent frames. Following STM [7], we consider the past frames with object masks as *memory* frames which are used to predict the object mask for the current (*query*) frame using an attention-based memory read operation. Notably, we propose a novel and lightweight top-k operation that integrates with STM and show that it improves both performance and speed without complicated training tricks.

Memory Read with Top-k Filtering We build two encoder networks: the memory encoder and the query encoder. Their network backbones are extracted from ResNet50 [53] up to stage-4 (`res4`) with a stride of 16. Extra input channels are appended to the first convolution of the memory encoder which accepts object masks as input. At the end of each encoder, two separate convolutions are used to produce two features maps: key $\mathbf{k} \in \mathbb{R}^{C^k \times HW}$ and value $\mathbf{v} \in \mathbb{R}^{C^v \times HW}$ where H and W are the image dimensions after stride, and C^k and C^v are set to 128 and 512 respectively.

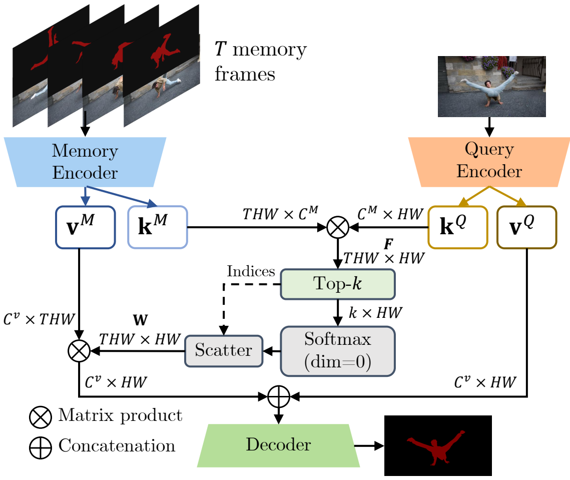

Figure 5. Implementation of our space-time memory reader as described in Section 3.3. Tensor reshaping is performed when needed. Skip-connections from the query encoder to the decoder are omitted for clarity.

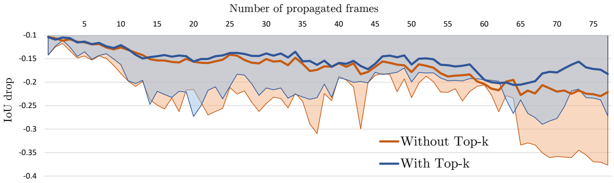

Figure 6. Mean IoU drop along propagation with or without top-k filtering with the color bands showing the interquartile range (the higher the better). With top-k filtering, the propagation is more stable and performs better especially for temporally far away frames when the noise-to-k ratio is large.

Figure 5 illustrates our space-time memory read operation. For each of the T memory frames, we compute key-value features and concatenate the output as memory key $\mathbf{k}^M \in \mathbb{R}^{C^k \times THW}$ and memory value $\mathbf{v}^M \in \mathbb{R}^{C^v \times THW}$. The key $\mathbf{k}^Q$ computed from the query is matched with $\mathbf{k}^M$ via a dot product:

$$\mathbf{F} = \left(\mathbf{k}^M\right)^T \mathbf{k}^Q, \tag{1}$$

where each entry in $\mathbf{F} \in \mathbb{R}^{THW \times HW}$ represents the affinity between a query position and a memory position. Previous methods [7, 54] would then apply softmax along the memory dimension and use the resultant probability distribution as a weighted-sum for $\mathbf{v^M}$. We have two observations on this softmax strategy: 1) For each query position, most of the weights will fall into a small set of memory positions and the rest are noises, and 2) these noises grow with the size of the memory and are performance-degrading when the sequence is long.

Based on these observations, we propose to filter the affinities such that only the top-k entries are kept. This effectively removes noises regardless of the sequence length. Since softmax preserves order, we can apply top-k filtering beforehand to reduce the number of expensive exp calls. In practice, our new top-k strategy not only increases robustness but also overcomes the overhead of top-k (see Table 3). Figure 6 reports the performance increase and robustness brought by top-k filtering. Note that KMN [54] (a recent modification of STM) imposes a Gaussian locality prior on the *query* using the *memory*, while our top-k operation filters the *memory* using the *query*. Refer to the supplementary material for a detailed comparison.

In summary, the affinity of memory position i with query position j can be computed by:

$$\mathbf{W}_{ij} = \frac{\exp\left(\mathbf{F}_{ij}\right)}{\sum_{p \in \text{Top}_j^k(\mathbf{F})}\left(\exp\left(\mathbf{F}_{pj}\right)\right)}, \text{if } i \in \text{Top}_j^k(\mathbf{F}) \tag{2}$$

and 0 otherwise. $\text{Top}_j^k(\mathbf{F})$ denotes the set of indices that are top-k in the j-th column of $\mathbf{F}$. These attentional weights are used to compute a weighted-sum of $\mathbf{v}^M$. For query position j, the feature $\mathbf{m}_j$ is read from memory by:

$$\mathbf{m}_j = \sum_{p}^{THW} \mathbf{v}_p^M \mathbf{W}_{pj} \tag{3}$$

The read features will be concatenated with $\mathbf{v}^Q$ and passed to the decoder to generate the object mask. Skip-connections (not shown for clarity) from the query encoder to the decoder help to create a more accurate mask. The output of the decoder is a stride 4 mask which is bilinearly upsampled to the original resolution. When there are multiple objects, we process each object one by one and combine the masks using soft aggregation [7].

Propagation strategy Figure 7 illustrates our bidirectional propagation strategy, similar to [9]. Given a user-interacted reference frame $M_{t^r}^r$, we bidirectionally propagate the segmentation to other frames with two (forward and backward) independent passes. Given that each interacted frame is sufficiently well-annotated (which is more easily satisfied under our decoupled framework), the propagation stops once hitting a previously interacted frame or the end of the sequence. Following STM [7], every 5th frame will be included and cached in the memory bank. The frame immediately before the query frame will also be included as temporary memory. In interactive settings, all user-interacted frames are trusted and added to the memory bank.

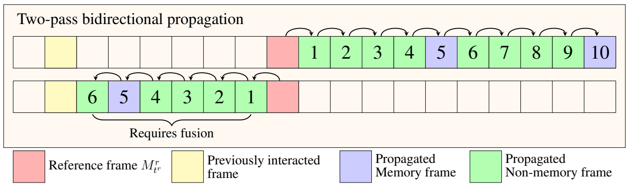
Two-pass bidirectional propagation
1
2
3
4
5
6
7
8
9
10
6
5
4
3
2
1
Requires fusion
Reference frame $M_{t^r}^r$
Previously interacted frame
Propagated Memory frame
Propagated Non-memory frame

Figure 7. Illustration of our propagation scheme. The frames between the current reference frame and previously interacted frame require fusion which is described in Section 3.4.

Evaluation The propagation module can be isolated for evaluation in a semi-supervised VOS setting (where the first-frame ground-truth segmentation is propagated to the entire video). Table 1 tabulates our validation of the effectiveness of top-k filtering (our new dataset **BL30K** to be detailed in Section 4). The algorithm is not particularly sensitive to the choice of k with similar performance for $k = 20$ through 100. $k = 50$ in all our experiments. In principle, the value of k should be linear to the image resolution such that the effective area after filtering is approximately the same. With top-k filtering, our multi-object propagation runs at 11.2 FPS on a 2080Ti.

Methods	Top-k?	BL30K?	$\mathcal{J}\&\mathcal{F}$
RGMP [28]	-	-	66.7
FEELVOS [21]	-	-	71.5
PReMVOS [8]	-	-	77.8
STM [7]	-	-	81.8
CFBI [23]	-	-	81.9
KMN [54]	-	-	82.8
GraphMem [55]	-	-	82.8
Ours	✗	✗	81.5_{-}
Ours	✗	✓	$83.8_{\uparrow 2.3}$
Ours	✓	✗	$83.1_{\uparrow 1.6}$
Ours	✓	✓	$\mathbf{84.5}_{\uparrow \mathbf{3.0}}$

Table 1. Evaluation of our propagation module in the DAVIS 2017 multi-object semi-supervised validation set. Both top-k filtering and BL30K are effective in increasing the performance. ↑ indicates improvement over our baseline. In addition, we obtain 76.5 $\mathcal{J}\&\mathcal{F}$ on the DAVIS `test-dev` set which is more difficult with harsh lighting conditions. Refer to the project website for more results.

3.4. Difference-Aware Fusion

If the propagation ends with hitting a previously interacted frame t^c, there may exist conflicts in frames within t^c and t^r. Fusion is thus required between the current propagated mask $M^{r'}$ and the previous mask results M^{r-1}. Previous approaches [9, 11] often employ a linear weighting scheme which is agnostic to the correction made and thus fails to capture the user's intent. Oftentimes, the user correction will disappear mid-way between t^r and t^c.

As illustrated in Figure 8, we propose a novel learnable fusion module that can keep the user correction in mind during fusion. Specifically, the user correction is captured as the differences in the mask before and after the user interaction at frame t^r:

$$\mathcal{D}^+ = \left(M_{t^r}^r - M_{t^r}^{r-1}\right)_+ \quad \mathcal{D}^- = \left(M_{t^r}^{r-1} - M_{t^r}^r\right)_+ \tag{4}$$

where $(\cdot)_+$ is the $\max(\cdot, 0)$ operator. We compute the positive and negative changes separately as two masks $\mathcal{D}^+$ and $\mathcal{D}^-$. To fuse t_i, which is between t^r and t^c, these masks cannot be used directly as they are not aligned with the target frame t_i. The key insight is that we can leverage the affinity matrix $\mathbf{W}$ in Eq. (2) computed by our space-time memory reader (Figure 5) for correspondence matching. The interacted frame t^r and target frame t_i are used as memory and query respectively. The aligned masks are

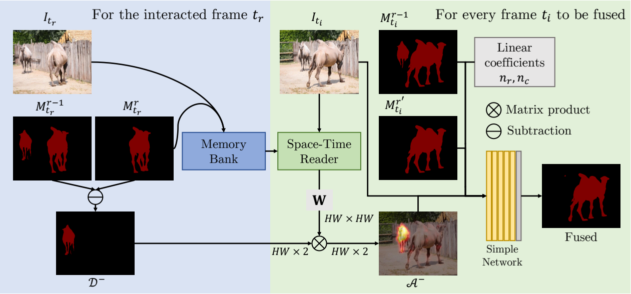

Figure 8. Mechanism of the difference-aware fusion module. The current propagated mask $M_{t_i}^{r'}$ at frame I_{t_i} is fused with the previous mask $M_{t_i}^{r-1}$, guided by the mask difference from interaction at frame t_r. Only the negative terms $\mathcal{D}^-, \mathcal{A}^-$ are shown here for clarity. Note that although a correct mask is captured in $M_{t_i}^{r'}$, it is non-trivial to pick it up in the fusion step as shown in Figure 9.

computed by two matrix products:

$$\mathcal{A}^+ = \mathbf{W}\mathcal{D}^+ \quad \mathcal{A}^- = \mathbf{W}\mathcal{D}^- \tag{5}$$

Where $\mathcal{D}^+$ and $\mathcal{D}^-$ are downsampled using area averaging to match the image stride of $\mathbf{W}$, and the results are upsampled bilinearly to the original resolution. Additionally, traditional linear coefficients are also used to model possible decay during propagation:

$$n_r = \frac{|t_i - t_r|}{|t_c - t_r|} \quad n_c = \frac{|t_i - t_c|}{|t_c - t_r|} \tag{6}$$

Note that $n_r + n_c = 1$. Finally, the set of features $(I_{t_i}, M_{t_i}^{r'}, M_{t_i}^{r-1}, \mathcal{A}^+, \mathcal{A}^-, n_r, n_c)$ are fed into a simple five-layer residual network which is terminated by a sigmoid to output a final fused mask.

As illustrated in Figure 9, our fusion method can capture the user's intention as an aligned attention map, which allows our algorithm to propagate corrections beyond the mid-point. Such fusion cannot be achieved in previous linear-blending methods [9, 11] (non-symmetric blending [11] will fail if we swap the order of interaction). Evaluation of the fusion module is presented in Section 6.2.

4. Dataset: BL30K

High-quality VOS datasets are expensive to collect at a large scale – DAVIS [41] is high-quality yet lacks quantity; YouTubeVOS [56] is large but has moderate quality annotations. In this paper we contribute a new synthetic VOS dataset **BL30K** that not only is large-scale but also provides pixel-accurate segmentations. Table 2 compares the three datasets.

Dataset	# Videos	# Frames	Label Quality
DAVIS [41]	90	6,208	High
YV [56]	3,471	94,588	Moderate
BL30K	29,989	4,783,680	High

Table 2. Comparison between different VOS datasets. Only frames with publicly available ground truths are counted.

Using an open-source rendering engine *Blender* [57, 58], we animate 51,300 three-dimensional models from ShapeNet [59] and produce the corresponding RGB images and segmentations with a two-pass rendering scheme. Background images and object textures are collected using Google image search to enrich the dataset. Each video consists of 160 frames with a resolution of 768 × 512. Compared with FlythingThings3D [60], our videos have a higher frame rate and a much longer sequence length, making ours suitable for the VOS task while [60] is not applicable. Figure 10 shows one sample in our dataset. To the best of our knowledge, BL30K is the largest publicly available VOS dataset to date. Despite that the dataset being synthetic, it does significantly help in improving real-world performance as shown in our ablation study (Section 6.2). Note that this gain is *not* simply caused by more training iterations as extended training on YouTubeVOS [56] and DAVIS [1] leads to severe overfitting in our experiments.

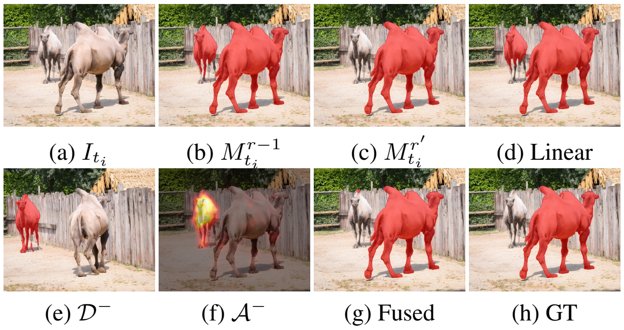
(a) I_{t_i} (b) $M_{t_i}^{r-1}$ (c) $M_{t_i}^{r'}$ (d) Linear
(e) $\mathcal{D}^-$ (f) $\mathcal{A}^-$ (g) Fused (h) GT

Figure 9. Continuing Figure 8, showing popularly used linear blending is insufficient. Suppose the user first annotates $t_c = 25$, then corrects the mask at $t_r = 89$. For the query frame with $t_i = 51$ which is closer to 25 than to 89, linear blending (or any symmetric function that only uses the temporal distance) fails in (d). With our difference aware fusion, we use the mask difference (e) to form an aligned attention (f) that captures the correction. Our result is shown in (g).

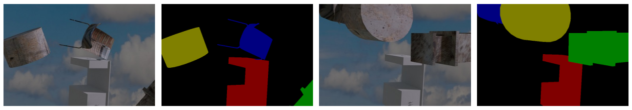
Figure 10. Sample data from the BL30K dataset.

5. Implementation Details

All three modules can be efficiently trained using just two 11GB GPU with the Adam optimizer [61]. The propagation module is first trained on synthetic video sequences from static images following [7], which is then transferred to BL30K, YouTubeVOS [56] and DAVIS [1]. In each training iteration, we pick three random frames in a video sequence, with the maximum distance between frames increased from 5 to 25 gradually (curriculum learning) and annealed back to 5 toward the end of training [62]. The S2M module is independently trained on static images only. The fusion module is trained with the output of a pretrained propagation module, first on BL30K, and then transferred to DAVIS [1]. YouTubeVOS [56] is not used here due to its less accurate annotation. Table 3 tabulates the running time of different components in our model. Refer to our open-sourced code for detailed hyperparameters settings. It takes about two weeks to train all the modules with two GPUs.

	Time (ms) / frame / instance
Scribble-to-Mask (S2M)	29
f-BRS [4]	∼60
Propagation w/o top-k	51
Propagation w/ top-k	44
Fusion	9

Table 3. Running time analysis of each component in our model. Time is measured on the 480p DAVIS 2017 validation set; time for propagation is amortized. For an average of two objects in DAVIS 2017, our baseline performance matches the one reported in STM [14]. Run time of f-BRS depends on the input as adaptive optimization is involved. Note that propagation is performed sparsely which keep our algorithm the fastest among competitors.

6. Experiments

6.1. DAVIS Interactive Track

In the DAVIS 2020 Challenge [41] interactive track, the robot first provides scribbles for a selected frame, waits for the algorithm's output, and then provides corrective scribbles for the worst frame of all the candidate frames listed by the algorithm. The above is repeated up to 8 rounds. To demonstrate the effectiveness of our proposed decoupled method which requires less temporally dense interactions, we limit ourselves to interact with only three frames. Specifically, we force the robot to only pick a new frame in the 1st, 4th, and 7th interactions. Our algorithm stays in an instant feedback loop for the same frame and performs propagation only when the robot has finished annotating one frame. Note that this behavior can be implemented without altering the official API.

Table 4 tabulates the comparison results. Figure 11 plots the performance measured on $\mathcal{J}\&\mathcal{F}$ versus time. Note that, even with the above additional constraint, our method outperforms current state-of-the-art methods. We use the same GPU (RTX 2080Ti) as our closest competitor [14]. Figure 12 provides qualitative comparisons and visual results.

Methods	AUC-$\mathcal{J}$	$\mathcal{J}^{\dagger}$	AUC-$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}^{\dagger}$
Oh *et el.* [9]	69.1	73.4	-	-
MANet [12]	74.9	76.1	-	-
ATNet [11]	77.1	79.0	80.9	82.7
STM [14]	-	-	80.3	84.8
Ours	**84.9**	**85.4**	**87.9**	**88.5**

Table 4. Performance on the DAVIS interactive validation set. Our method outperforms all competitors while receiving only interactions in 3 frames instead of 8. †Interpolated value @60s.

6.2. Ablation Study

Table 5 tabulates the quantitative evaluation on the effectiveness of BL30K and the fusion module. We show that 1) the proposed top-k memory read transfers well to the interactive setting, 2) BL30K helps in real-world tasks despite being synthetic, and 3) Difference-aware fusion module outperforms naïve linear blending and difference-agnostic (learnable) fusion with the same network architecture. Additionally, we show the upper bound performance of our method given perfect interaction masks.

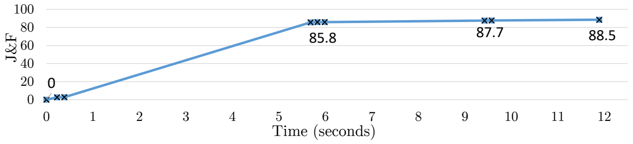

Figure 11. $\mathcal{J}\&\mathcal{F}$ performance on the DAVIS validation set. Clustered points represent real-time corrections in the instant feedback loop; each cluster represents a frame switch and propagation. Our method is highly efficient, achieving better performance in ∼12 seconds on average compared with 55+ seconds in [11] or 37 seconds in [14].

Model	AUC-$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$@60s
STM	80.3	84.8
Baseline	86.0_{-}	86.6_{-}
(+) Top-k	$87.2_{\uparrow 1.2}$	$87.8_{\uparrow 1.2}$
(+) BL30K pretraining	$87.4_{\uparrow 1.4}$	$88.0_{\uparrow 1.4}$
(+) Learnable fusion	$87.6_{\uparrow 1.6}$	$88.2_{\uparrow 1.6}$
(+) Difference-aware (Full model)	$\mathbf{87.9_{\uparrow 1.9}}$	$\mathbf{88.5_{\uparrow 1.9}}$
Perfect interaction	90.2	90.7

Table 5. Ablation study on the DAVIS interactive validation set. Our decoupled baseline already outperforms SOTA by a large margin. Despite the high baseline, we show that top-k memory filtering, pretraining in the BL30K dataset, and the difference-aware fusion module can further improve its performance. In the last row, we replace the interaction module with an oracle that provides ground-truth masks to evaluate the upper-bound of our method given perfect interactions in 3 frames.

6.3. User Study

We conduct a user study to quantitatively evaluate user's preferences and human effort required to label a video using iVOS algorithms. Specifically, we quantify the required human effort by the total *user time* which includes the time for interaction, searching, or pausing to think while excluding all computational time. We linearly interpolate the IoU versus user-time graph and compute the area under curve (AUC) for evaluation. We compare with ATNet [11] which is the best performing method with available source code to the best of our knowledge. We use two variants of our method – one with S2M as the only interaction option (Ours-S2M), and the other allows users to use a combination of S2M, f-BRS [4] and free-hand drawing, with the local control option (Ours-Free).

We recruited 10 volunteers who were given sufficient time to familiarize themselves with different algorithms and the GUI. They were asked to label 5 videos in the DAVIS 2017 multi-object validation set with satisfactory

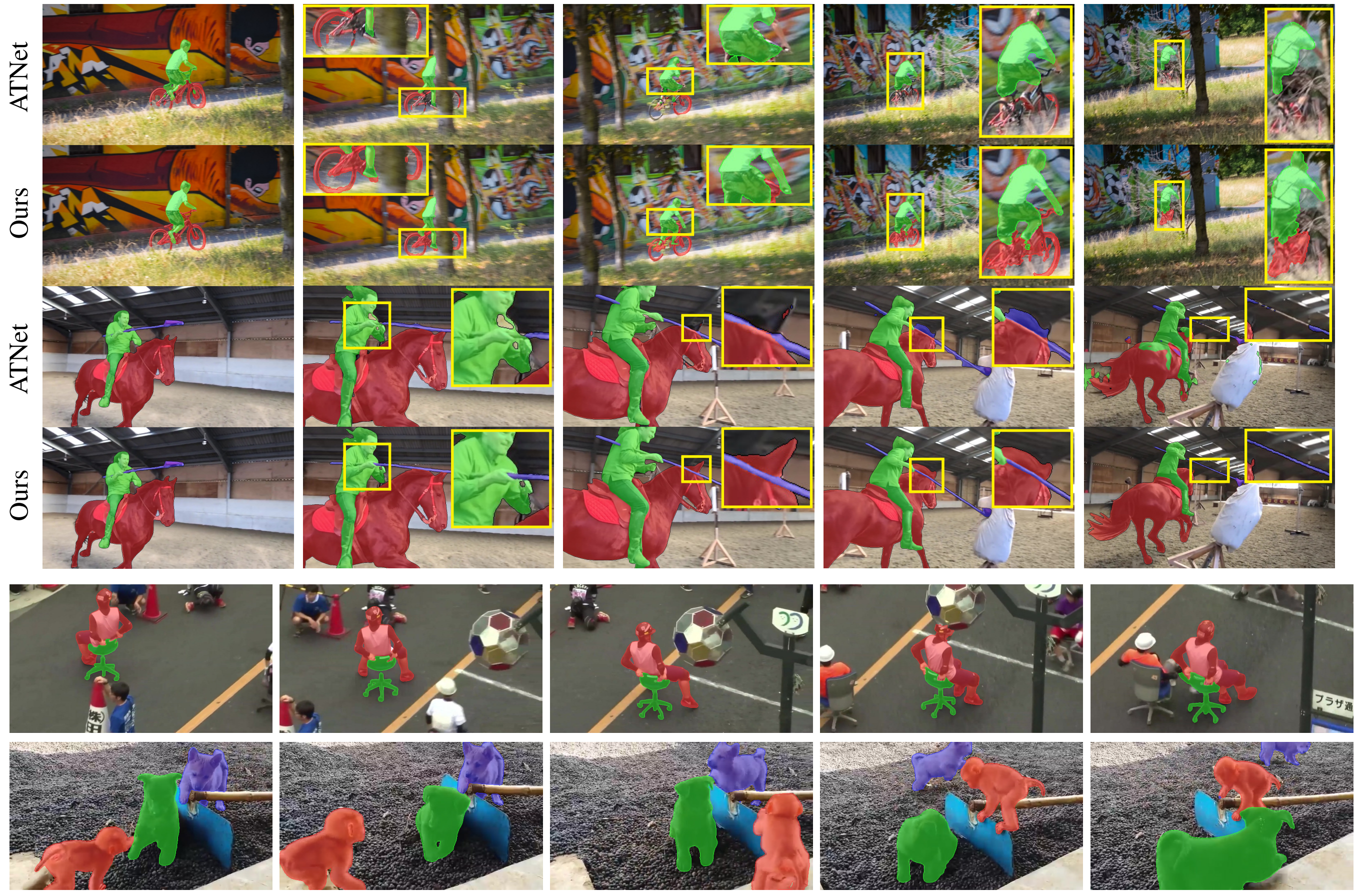

Figure 12. Top four rows: Qualitative comparison of our method with ATNet [11] on the DAVIS interactive track (top two) and on previously unseen Internet video (middle two) with real user interactions (as detailed as possible on two frames). Bottom two rows: More results from our method on real-world videos from the Internet. Additional video results can be found on the project website.

accuracy as fast as possible, within a 2-minute wall clock time limit. To avoid familiarity bias, they studied the images and ground truths of each video before each session. Figure 13 shows the IoU versus user-time plot and Table 6 tabulates the average performance gain after each interaction. Our method achieves better results with less interaction time, while including more interaction options (f-BRS, free-hand drawing, and local control) which allows our method to converge faster and to a higher final accuracy for experienced users.

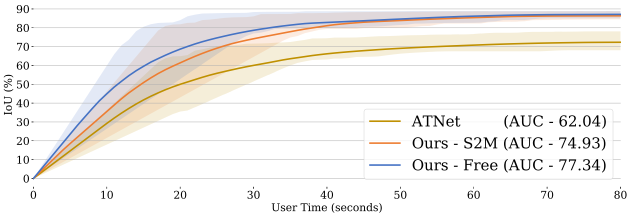

Figure 13. Mean IoU versus user time plot with shaded regions showing the interquartile range. Our methods achieve higher final accuracy and AUC than ATNet [11]. In Ours-Free, users make use of f-BRS [4] to obtain a faster initial segmentation. Experienced users can use free hand drawing and local control to achieve higher final accuracy given more time.

Methods	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Sum
ATNet [11]	62.2	6.82	1.93	2.57	1.61	75.1
Ours-S2M	83.8	1.56	0.64	0.37	0.53	86.9
Ours-Free	84.3	1.69	0.66	0.66	0.62	87.9

Table 6. Mean incremental IoU improvement after each interaction round. Δ_i denotes the IoU gain after the ith frame interaction and propagation. ATNet [11] requires more interactions to achieve stable performance while ours achieves higher accuracy with less interactions. Enabling other interaction modes such as f-BRS or local control (Ours-Free) is beneficial to both the speed and the final accuracy. Note that sum does not equal to the final mean IoU in the left plot because not all users interacted for five rounds.

7. Conclusion

We propose MiVOS, a novel decoupled approach consisting of three modules: Interaction-to-Mask, Propagation and Difference-Aware Fusion. By decoupling interaction from propagation, MiVOS is versatile and not limited by the type of interactions. On the other hand, the proposed fusion module reconciles interaction and propagation by faithfully capturing the user's intent and mitigates the information lost in the decoupling process, thus enabling MiVOS to be both accurate and efficient. We hope our MiVOS can inspire and spark future research in iVOS.

References

[1] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. In *arXiv:1803.00557*, 2018.

[2] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. In *ToG*, 2004.

[3] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. In *BMVC*, 2017.

[4] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *CVPR*, 2020.

[5] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018.

[6] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017.

[7] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[8] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018.

[9] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Fast user-guided video object segmentation by interaction-and-propagation networks. In *CVPR*, 2019.

[10] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Interactive video object segmentation using sparse-to-dense networks. In *CVPR Workshops*, 2019.

[11] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Interactive video object segmentation using global and local transfer modules. In *ECCV*, 2020.

[12] Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *CVPR*, 2020.

[13] Chen Liang, Zongxin Yang, Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregated cfbi+ for interactive video object segmentation. In *CVPR Workshops*, 2020.

[14] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Space-time memory networks for video object segmentation with user guidance. *TPAMI*, 2020.

[15] Arnaud Benard and Michael Gygli. Interactive video object segmentation in the wild. In *arXiv:1801.00269*, 2017.

[16] Quoc-Cuong Tran, The-Anh Vu-Le, and Ming-Triet Tran. Interactive video object segmentation with multiple reference views, self refinement, and guided mask propagation. In *CVPR Workshops*, 2020.

[17] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017.

[18] 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.

[19] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation. In *CVPR Workshops*, 2017.

[20] Kai Xu, Longyin Wen, Guorong Li, Liefeng Bo, and Qingming Huang. Spatiotemporal cnn for video object segmentation. In *CVPR*, 2019.

[21] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019.

[22] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018.

[23] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020.

[24] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *NIPS*, 2017.

[25] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, 2019.

[26] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, 2019.

[27] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018.

[28] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018.

[29] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.

[30] Xi Chen, Zuoxin Li, Ye Yuan, Gang Yu, Jianxin Shen, and Donglian Qi. State-aware tracker for real-time video object segmentation. In *CVPR*, 2020.

[31] 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.

[32] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, 2020.

[33] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018.

[34] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *ECCV*, 2020.

[35] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *CVPR*, 2020.

[36]

[37] Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. Capsulevos: Semi-supervised video object segmentation using capsule routing. In *ICCV*, 2019.

[38] Jue Wang, Pravin Bhat, R Alex Colburn, Maneesh Agrawala, and Michael F Cohen. Interactive video cutout. In *ToG*, 2005.

[39] Brian L Price, Bryan S Morse, and Scott Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009.

[40] Naveen Shankar Nagaraja, Frank R Schmidt, and Thomas Brox. Video segmentation with just a few strokes. In *ICCV*, 2015.

[41] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. In *arXiv:1905.00737*, 2019.

[42] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ToG*, 2004.

[43] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *PACMCGIT*, 1995.

[44] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *IJCV*, 1988.

[45] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, 2016.

[46] Eirikur Agustsson, Jasper RR Uijlings, and Vittorio Ferrari. Interactive full image segmentation by considering all regions jointly. In *CVPR*, 2019.

[47] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[48] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[49] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *ICCV*, 2019.

[50] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. In *TPAMI*, 2015.

[51] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *CVPR*, 2020.

[52] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *CVPR*, 2020.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[54] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020.

[55] 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.

[56] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *ECCV*, 2018.

[57] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[58] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv:1911.01911*, 2019.

[59] Angel Xuan Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. In *arXiv:1512.03012*, 2015.

[60] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[61] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

[62] Peng Zhang, Li Hu, Bang Zhang, and Pan Pan. Spatial consistent memory network for semi-supervised video object segmentation. In *CVPR Workshops*, 2020.