XMem++: Production-level Video Segmentation From Few Annotated Frames

Maksym Bekuzarov[1*], Ariana Bermudez[1*] Joon-Young Lee [3] Hao Li [1,2]
{maksym.bekuzarov,ariana.venegas}@mbzuai.ac.ae jolee@adobe.com hao@hao-li.com
[1]MBZUAI [2]Pinscreen [3]Adobe Research

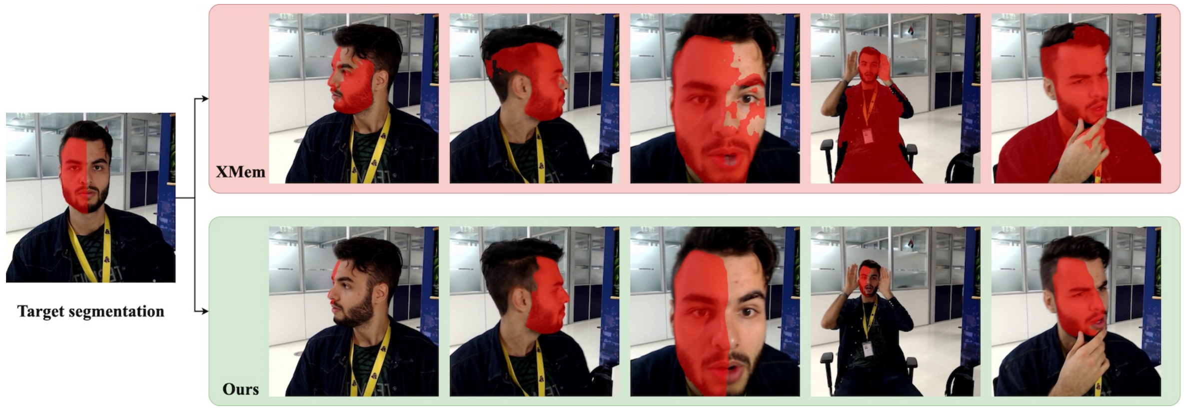

Figure 1. A demonstration of partial region segmentation (left half of the face) for extreme poses. We compare our method with the current SOTA, XMem [4], for a 1 min video (1800 frames), with only 6 frames (0.33%) annotated. No retraining or fine-tuning required.

Abstract

*Despite advancements in user-guided video segmentation, extracting complex objects consistently for highly complex scenes is still a labor-intensive task, especially for production. It is not uncommon that a majority of frames need to be annotated. We introduce a novel semi-supervised video object segmentation (SSVOS) model, **XMem++**, that improves existing memory-based models, with a permanent memory module. Most existing methods focus on single frame annotations, while our approach can effectively handle multiple user-selected frames with varying appearances of the same object or region. Our method can extract highly consistent results while keeping the required number of frame annotations low. We further introduce an iterative and attention-based frame suggestion mechanism, which computes the next best frame for annotation. Our method is real-time and does not require retraining after each user input. We also introduce a new dataset, **PUMaVOS**, which covers new challenging use cases not found in previous benchmarks. We demonstrate SOTA performance on challenging (partial and multi-class) segmentation scenarios as well as long videos, while ensuring significantly fewer frame annotations than any existing method. Project page: https://max810.github.io/xmem2-project-page/*

[*]These authors contributed equally to the work

1. Introduction

Video Object Segmentation (VOS) [48] is a widely performed vision task with applications ranging from object recognition, scene understanding, medical imaging, to filter effects in video chats. While fully automated approaches based on pre-trained models for object segmentation are often desired, interactive user guidance is commonly practiced to annotate new training data or when precise rotoscoping is required for highly complex footages such as those found in visual effects. This is particularly the case when the videos have challenging lighting conditions and dynamic scenes, or when partial region segmentation is required. While automatic VOS methods are designed to segment complete objects with clear semantic outlines, interactive video object segmentation (IVOS) and semi-supervised video object segmentation (SSVOS) techniques [48] are more flexible, and typically use a scribble or contour drawing interface for manual refinement such as those found in commercial software solutions such as Adobe After Effects and Nuke. Despite advancements in IVOS and SSVOS techniques, rotoscoping in film production is still a highly labor-intensive task, and often requires nearly every frame of a shot to be annotated and refined [36].

State-of-the-art IVOS and SSVOS techniques use memory-based models [27] and have shown impressive segmentation results on complex scenes based on user-provided mask annotations, but they are often designed to improve single annotation performances [27, 43, 44, 6, 4] and are still not suitable for production use cases. In particular, they tend to over-segment known semantic outlines (person, hair, faces, entire objects) and fail on partial regions (e.g., half of a person's face, a dog's tail), harsh lighting conditions such as shadows, and extreme object poses. As a result, inconsistent segmentations are obtained when only a single annotated frame is provided, due to the inherent ambiguity of the object's appearance, Especially when it varies too much due to large viewing angles and complex lighting conditions, which limits the scalability of these techniques, and it is often unclear which frame annotations to prioritize, especially for long sequences.

We propose a new SSVOS framework, **XMem++**, which uses a permanent memory module that stores all annotated frames and makes them available as references for all the frames in the video. While most SSVOS methods focus on single-frame mask annotations, our approach is designed to handle multiple frames that can be updated by the user iteratively with optimal frames being recommended by our system. While we adopt the cutting-edge architecture of XMem [4] as backbone, we show that our important modification enables accurate segmentation of challenging video objects (including partial regions) in complex scenes with significantly fewer annotated frames than existing methods. Our modification does not require any re-training or calibration and additionally shows improved temporal coherence in challenging scenarios (Fig. 1, 6, 8). Our attention-based frame suggestion method predicts the next candidate frame for annotation based on previous labels while maximizing the diversity of frames being selected. Our system supports both sparse (scribbles) and dense (full masks) annotations and yields better quality scaling with more annotations provided than existing methods. The video segmentation performs in real-time, and frame annotations are instantly taken into account with a pre-trained network.

We further introduce a new dataset, **PUMaVOS** for benchmarking purposes, which includes new challenging scenes and use cases, including occlusion, partial segmentation, and object parts segmentation, where the annotation mask boundaries may not correspond to visual cues.

We evaluate the performance of our algorithm both qualitatively and quantitatively on a wide range of complex video footages as well as existing datasets, and demonstrate SOTA segmentation results on complex scenes. In particular, we show examples where our method achieves higher accuracy and temporal coherence than existing methods with up to $5\times$ fewer frame annotations and on $2\times$ twice less on existing benchmarks (Section 5). We further demonstrate the effectiveness of our frame annotation candidate selection method by showing that it selects semantically meaningful frames, similar to those chosen by expert users. We make the following contributions:

- We have introduced a new VOS model, **XMem++**, that uses a permanent memory module that effectively utilizes multiple frame annotations and produces temporally-smooth segmentation results without overfitting to common object cues.
- We further propose the use of an attention-based similarity scoring algorithm that can take into account previously predicted frame annotations to suggest the next best frame for annotation.
- We present a new dataset, **PUMaVOS**, which contains long video sequences of complex scenes and non object-level segmentation cues, which cannot be found in existing datasets.
- We achieve SOTA performance on major benchmarks, with significantly fewer annotations, and showcase successful examples of complex multi-class and partial region segmentations that fail for existing techniques.

2. Related Works

Video Object Segmentation. A wide range of video object segmentation (VOS) methods have been introduced in the past decade [48], spanning a broad spectrum of computer vision applications, including visual effects. Many solutions have been deployed in established commercial video editing software such as Adobe After Effects and Nuke. While early VOS techniques were often based on classic optimization methods and graph representations [33, 37], recent ones are typically using deep neural networks.

Semi-supervised video object segmentation (SSVOS) aims at segmenting objects in a video using a frame of reference (usually the first [3] but some models also support multiple annotations). To help users create the annotation, interactive VOS methods (IVOS) were introduced [28, 5], providing users a convenient way to create annotation masks commonly using scribbles and dots selection interface.

To facilitate user annotations, some IVOS methods suggest a fine-tuning approach, which makes any iterative user interaction slow during inference as retraining is required [3, 39]. Although more efficient alternatives like online adaptation were introduced, their output quality is generally poorer [25, 34, 29, 2].

Attention-based methods use different techniques such as similarity or template matching algorithms to dictate which frames need to be focussed on from a set of available frames (referred to as memories) [27, 9, 47, 15, 11]. Multiple authors have focused on facilitating the model to use local/pixel-to-pixel information which improves the quality of the masks using either kernels[35], optical flow[40, 46], transformers [24, 18, 43, 44, 46] or improvements to the spatial-temporal memory [38, 42, 5, 6, 21, 20, 23, 22, 19].

Most recently, XMem [4] has proposed a resource-efficient method that compresses the feature memory and supports the usage of multiple annotated frames references. We base our approach on it, as the architecture is resource-efficient, quick, extendable, and demonstrates SOTA results on modern benchmarks.

Figure 2. Assessed Quality (AQ) and Visual Quality (VQ) using quality assessement modele from IVOS-W [45], demonstrated on one of the videos from **PUMaVOS**. **Left half of the face** is being segmented. *Assessed quality does not correspond with visual quality for partial segmentation.*

Frame Annotation Candidate Selection. The task of annotation candidate prediction is finding a specific frame or set of frames for the user to annotate, in the first or consecutive interaction rounds that maximize the overall video segmentation quality (defined with metrics like *Intersection over Union* (IoU) and *F-score*). The authors of BubbleNets [12] propose a VOS architecture that simultaneously learns to predict the optimal candidate frame to annotate, by using uses a bubble-sort [16] style algorithm to find the close-to-best candidate. The authors of IVOS-W [45] claim that annotating the frame with the lowest quality is suboptimal and introduce a Reinforcement Learning Agent to intelligently select the candidates based on the assessed quality of all frames. GIS-RAmap [13] introduces an end-2-end deep neural network, that operates on sparse user input (scribbles), and uses the R-attention mechanism to segment frames and directly predicts the best annotation candidates.

These works exhibit the following limitations: [12, 45] work under the assumption that it is possible to directly estimate the segmentation quality or frame importance without explicit information of the target object, which makes them highly domain-dependent and does not hold for partial segmentation, as illustrated by Fig. 2c. [12] and [13] do not use annotation information when selecting annotation candidates, which limits their usefulness for partial segmentation, occlusions, or multiple similar objects in the scene (use-case illustrated in rows 1-2 in Fig. 7).

Video Segmentation Datasets. Earlier datasets had annotations for videos without heavy occlusions or appearance changes [26, 17, 10]. DAVIS [30, 31] became the benchmark dataset for the previous decade and is still being used because of the high resolution of the videos and the quality of the annotations. The benchmark released in 2016 had only single-object annotations, and the extended one in 2017 incorporated multiple-object annotations. For both datasets, their clips are 2-4 seconds long. Youtube-VOS [41] presented a large dataset of around 4000 videos where each is 3-6 seconds long. Given the number of videos, they have a variety of categories and every 5-th frame is annotated. OVIS [32] is a dataset of severe occlusions, and is very challenging for the current VOS methods. MOSE [8] uses a subset of OVIS [32] as well as other videos, for a total of 2149 clips, from 5 to 60 seconds long, 12 on average. LVOS [14] dataset consists of 220 long videos, on average 115 seconds each. Most recently, BURST [1] introduced a large dataset with almost 3000 videos, that can be used for tasks like VOS and Multi-Object Tracking and Segmentation. Even though there are resources with long videos and high-quality masks, many use cases from the industry are not reflected, such as partial objects, reflections, and segmentation targets without clear boundaries.

3. XMem++

3.1. Overview

Fig. 3 provides an overview of XMem++. Given a video, the end user first selects a frame f_i they want to annotate, to give the model the template of the object(s) that will be segmented. The user then provides the annotations in the form of scribbles (that are then converted to a dense segmentation mask m_i) or the mask m_i directly. The segmentation model (Section 3.2) then processes m_i, puts it into the permanent memory and segments the given target(s) in all other frames by predicting segmentation masks $\hat{\mathbf{m}}$. The annotation candidate selection algorithm (Section 3.3) takes m_i and $\hat{\mathbf{m}}$ and predicts the k next best frames to annotate, in order of importance. The user then provides the annotations for some or all of them, and the process is repeated until the segmentation quality is satisfactory. The annotation candidate selection module takes into account all of the previously annotated frames, so it avoids selecting the frames that are similar to those already annotated.

The segmentation module is described in Fig. 4. It is based on XMem architecture [4] and consists of a convolutional neural network with multiple parts and three types of separate memory modules. Given a sequence of frames $\mathbf{f}$ and at least one segmentation mask m_i containing the target object(s), the mask is processed together with the corresponding frame f_i by the model and stored in the permanent working memory as a reference for segmenting other frames. For every frame in the memory, two feature maps are extracted and saved - a smaller "key", containing information about the whole frame, used for matching similar frames, and a corresponding larger "value" with target-specific information, used in predicting the segmentation mask. When predicting the segmentation for a frame f_j, the model searches for similar frames in all three memory modules by calculating pixel-wise attention across stored "keys" using a scaled L_2 similarity measure, aggregates the information from them and uses it to predict the segmentation for the frame f_j. The model also stores its own predictions in the temporary working memory modules and uses them together, usually every n-th frame. The long-term memory module was introduced in XMem. It limits memory usage and allows processing longer videos by frequently

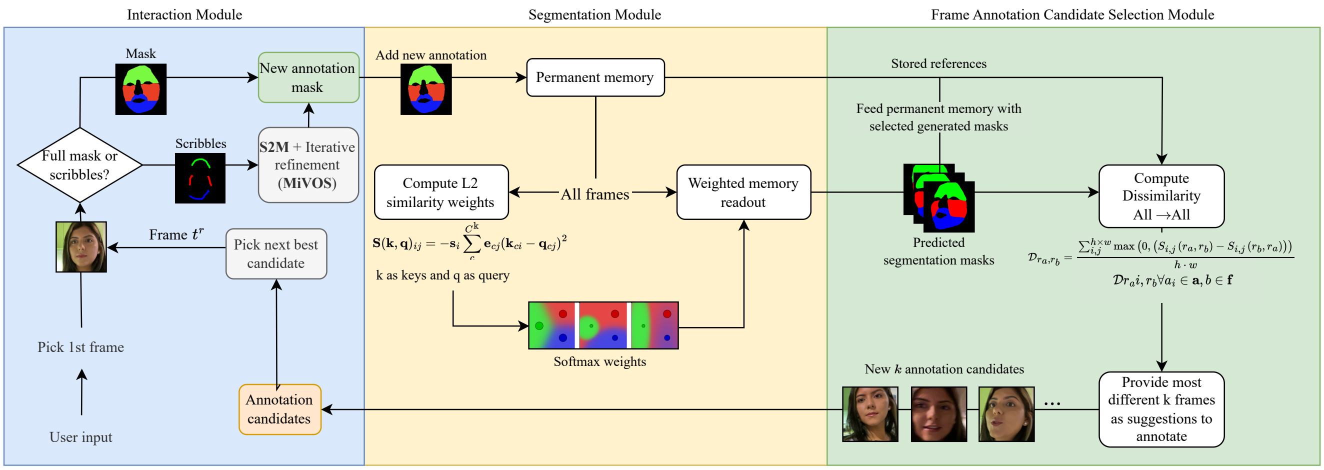

Figure 3. Interaction flow of our system. The user provides initial annotations, segmentation is performed, then using predicted masks new annotation candidates are chosen and given to the user. This loop is repeated until satisfactory segmentation quality is achieved.

compressing and removing outdated features from temporary working memory. Sensory memory is a small module that captures motion information by processing differences between consecutive frames, also unchanged from [4].

3.2. SSVOS with Permanent Memory Module

Our main contribution is the introduction of a "permanent" working memory module, that changes how the annotated frames are treated before and during model inference. In the original work, there was only temporary working memory, so the first frame was always annotated and permanently kept there for predicting segmentation masks for new frames. We define this frame as a "ground-truth reference", meaning that the segmentation mask associated with it is 100% accurate because it is provided by the user. All the other frames, which would be added to the working memory, were only added there temporarily and are likely to be moved to long-term memory eventually, especially for longer videos. We define these frames as "imperfect references", as the associated segmentation mask is predicted by the model, thus it is likely to have errors. This approach works great with only 1 annotation provided, however, two problems arise when using multiple annotations. First, visible "jumps" in quality appear when the model encounters a new annotated frame during the segmentation process, and corrects its predictions for the *following* frames, but not for the *previous* ones. Second, additional annotated frames were treated like "imperfect references" - thus only having an impact on a limited part of the video, and likely to be compressed and moved to long-term memory, reducing their impact even further.

To address these issues, we propose to add another, "permanent" working memory module (labeled dark green in Fig. 4), implemented to store only "ground-truth references" - i.e., annotated frames for the duration of the whole video. During inference, the annotated frames are processed separately and added to the new permanent working memory before the segmentation process starts. The "permanent" working memory module stays unchanged throughout the whole inference, its contents are never compressed or moved to the long-term memory. During memory readout, the keys and values in the permanent memory are simply concatenated to those of the temporary memory.

This allows the model to produce smooth transitions when the target object changes between two scenes, as the model now has access to references from both (refer to Fig. 10 from Appendix for illustration). The module also decouples the frame position from frame content, helping the model to find matches across frames regardless of their positions in the video. This results in an increased scaling of segmentation quality and efficiency, as well as fixes the "jumping" quality issue (Section 5).

3.3. Attention-Based Frame Selection

Choosing the right frames to annotate leads to higher overall segmentation accuracy, which was demonstrated by previous works [45, 13, 12]. We designed an algorithm for this based on an empirical idea - to select the most *diverse subset of frames* that capture the target object in different illumination conditions, pose and camera viewpoint, inspired by [13]. Given the task to select b frames from a video, we assume there are b different "scenes" in it, and sample the most representative frame from each.

Given a previous annotations ($a \geq 1$ since there is at least one mask provided by the user) and the predictions of the model for the rest of the frames in the video, we extract the "key" features k_i of size (h, w) with our segmentation module, weighted by corresponding mask m_i, obtaining a region-weighted mask r_i. This allows the algorithm to focus on target-specific region similarity, while still having

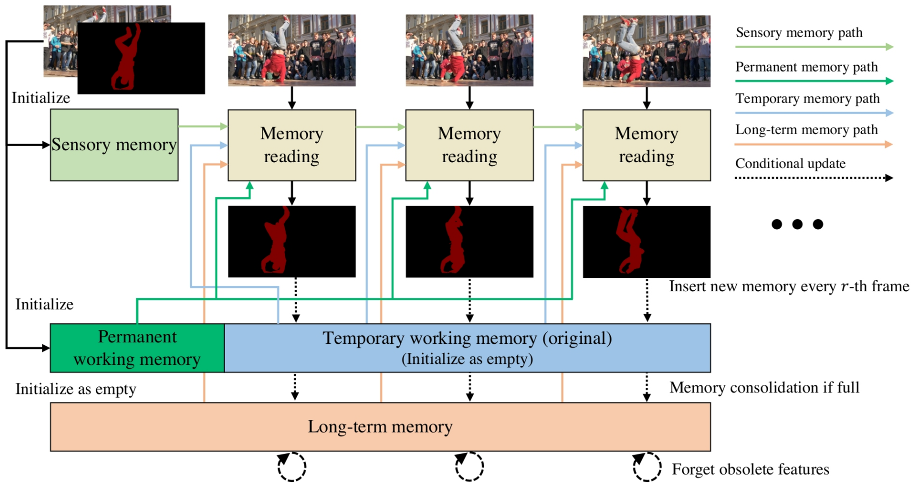

Figure 4. **XMem++** architecture. The new permanent memory module is shown in dark-green.

information about surrounding regions. The influence of the mask over the holistic frame features is controlled by parameter $\alpha, \alpha \in [0..1]$. With $\alpha = 0$ the algorithm ignores the annotation masks, predicting the candidates only based on the overall frame appearance, with $\alpha = 1$ it only looks at the masked regions, ignoring the rest of the pixels.

$$r_a = \alpha k_a \odot m_a + (1 - \alpha) k_a$$

We then iteratively pick $b - a$ candidates with the highest dissimilarity $\mathcal{D}$, using negative pixelwise L-2 similarity S from [4] with added cycle consistency.

$$\mathcal{D}_{r_a, r_b} = \frac{\sum_{i,j}^{h \times w} \max\left(0, \left(S_{i,j}(r_a, r_b) - S_{i,j}(r_b, r_a)\right)\right)}{h \cdot w}$$

Given frames $\mathbf{f}$ and previous annotations $\mathbf{f}$, we compute the dissimilarity across them: $\mathcal{D}_{r_a i, r_b} \forall a_i \in \mathbf{a}, b \in \mathbf{f}$. We then select $\operatorname{argmax}\left(\operatorname{argmin} \mathcal{D}_{r_a i, r_b}\right) \forall a_i \in \mathbf{a}, b \in \mathbf{f}$, the frame with the *largest minimal distance* to all the existing annotations, as the next candidate. This process is repeated $b - a$ times. Due to ambiguity in pixel-to-pixel mapping, often a lot of pixels from f_a map to other pixels from f_a, thus average self-dissimilarity is > 0. Cycle consistency $(S_{i,j}(r_a, r_b) - S_{i,j}(r_b, r_a))$ ensures that frames are only dissimilar if pixels in f_i map to different positions in f_j, then from f_j back to f_i. This guarantees that self-dissimilarity $\mathcal{D}_{r_i, r_i} = 0$. Refer to the Appendix for a step-by-step explanation of the algorithm.

This allows our algorithm to demonstrate the desirable properties: it does not select candidates similar to already chosen ones, is generic and applicable to any memory-based segmentation model, and does not assume that the mask should match the visual object cues, which is violated in the case of partial segmentation, shown in Sec. 2.

4. Dataset and Benchmark

We provide a new benchmark that covers use cases for multipart partial segmentation with visually challenging situations (segments of parts of the scene with little-to-no pixel-level visual cues) called Partial and Unusual MAsk Video Object Segmentation (**PUMaVOS**). To the best of our knowledge, there are no currently available datasets like this. We contemplate scenarios from the video production industry that still conventional VOS methods struggle to tackle. We focus on partial objects such as half faces, neck, tattoos, and pimples, which are frequently retouched in film production as shown in Fig 5. Our dataset consists of 24 clips 29 seconds long on average, 21K densely annotated frames. To generate the annotations, we adopted a similar approach to MOSE [8] that used a framework with XMem [4] to create masks for each frame, but instead we used our method, **XMem++**. In MOSE the videos were annotated every 5th frame (20% of the video), while in our case we noticed that complex scenes require 8% to 10% and simple scenes required 4% to 6% of total frames to be annotated.

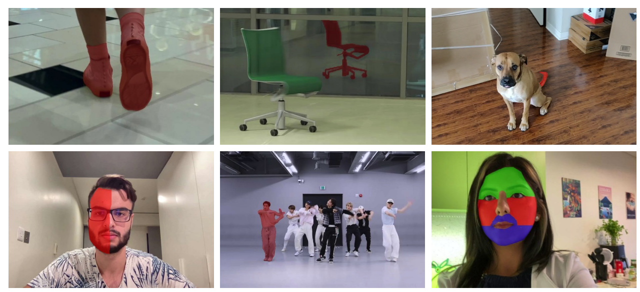

Figure 5. Samples of our dataset (**PUMaVOS**). Best viewed zoomed. See Fig. 9 in Appendix for more examples.

5. Results

We test our segmentation framework on a wide range of complex scenes (varying poses and deformations of human subjects, faces, rigid objects, t-shirts) and annotation tasks (object, partial, multi-class segmentation) in the presence of occlusions, lighting variations, and cropped views. In particular, we showcase rotoscoping examples that occur frequently in labor-intensive real production settings, where over 50% of frames would need to be annotated or refined [36]. Qualitative results are presented in Fig. 1, Fig. 6, and Fig. 8. Our videos are recorded at 30 fps and 1920×1080 resolution, and the input is resized to 854×480 when processed by our model. We also refer to the accompanying video and supplemental material to view the results.

In row a) of Fig. 6, we show a partial segmentation result where only the front part of the guitar body is annotated, not the side or back. Only 6 out of 924 total frames were annotated (0.6% total) in order to successfully produce reliable segmentations through a challenging sequence with frequent occlusions, pose variations, and rapid movement. Rows 6 in Fig. 8 depicts an example where the goal is to composite the reflection of an object (e.g., a chair) into a different scene, which is a common use case in visual effects production. Our chair rotates throughout the video, which changes its projected outline significantly, but a consistent segmentation can be performed by only annotating 6 frames out of 411 total frames (1.5%). This use-case is especially challenging since the scene contains a larger, more prominent object with the exact same appearance and movement. A challenging multi-class segmentation example for a long and diverse sequence is illustrated in row 3 of Fig. 8, where the subject's face is segmented into 3 different regions, one of which consists of two separate parts. All regions are segmented simultaneously, which prevents them from overlapping with each other which can happen when each region is segmented independently. The additional challenge in this scenario is that the regions are often not separated by prominent visual cues such as boundaries, corners, edges, etc. Highly consistent results can be extracted even in the presence of extreme lighting variations, moving background and head poses, where again only 6 out of 951 (0.6%) total frames have been annotated.

Evaluation. We evaluate our framework on a diverse set of complex videos from MOSE [8] dataset, as well as on long videos provided by LVOS [14]. We use the training subset of MOSE (since it does not provide annotation for the test subset), consisting of 1507 videos, and a validation subset of LVOS consisting of 50 videos*. We compare the performance of multiple SSVOS models when given 1, 5, and 10 uniformly-sampled annotated frames. For comparison, we picked existing works in SSVOS and IVOS, that support the usage of multiple annotation frames by design as well as support dense annotations, since MOSE and LVOS do not provide scribbles information.

The performance of **XMem++** with only one annotation given is equivalent to XMem. We see that on both datasets **XMem++** demonstrates noticeably better performance scaling in terms of $\mathcal{J}$ and $\mathcal{F}$ metrics, for 5 and 10 annotated frames provided at input. Moreover, it can be seen that **XMem++** achieves comparable or higher segmentation quality with fewer annotations provided (5) than the competition (10), thus making it on average $2\times$. Furthermore, we evaluate our model and XMem on a subset of **PUMaVOS** dataset and observe that on some videos the efficiency of **XMem++** is up to $5\times$ higher.

	Number of annotations provided						$\lvert\mathcal{D}_{1\to10}\rvert$	
	1 frame		5 frames		10 frames			
Method	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
TBD$_{DAVIS}$	42.72	**53.38**	52.17	63.64	60.04	71.65	+17.3	+18.3
TBD$_{YT}$	41.15	50.79	57.19	67.83	65.07	75.72	+23.9	+24.9
XMem	**44.06**	52.33	56.30	66.05	62.62	73.18	+18.5	+20.9
XMem++			**67.36**	**78.11**	**75.74**	**86.35**	**+31.7**	**+34.0**

Table 1. Quantitative results on LVOS [14] validation dataset. $\mathcal{J}$ and $\mathcal{F}$ mean Jaccard index and boundary F-score correspondingly, as defined in [30]. TBD$_{DAVIS}$ and TBD$_{YT}$ stand for TBD [7] model trained on DAVIS [30] and YouTube-VOS [41] datasets accordingly. *At $k = 5$ annotation frames **XMem++** achieves higher quality ($\mathcal{J}$ and $\mathcal{F}$) then all other models at $k = 10$ frames.* $\lvert\mathcal{D}_{1\to10}\rvert$ denotes the increase in segmentation quality from 1 to 10 annotated frames.

	Number of annotations provided						$\lvert\mathcal{D}_{1\to10}\rvert$	
	1 frame		5 frames		10 frames			
Method	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
TBD$_{DAVIS}$	43.34	49.70	56.57	63.00	63.15	69.15	+19.8	+19.5
TBD$_{YT}$	48.28	54.05	62.71	68.94	68.24	74.23	+20.0	+20.2
STCN	54.51	60.69	58.73	65.86	62.78	70.11	+8.3	+9.4
XMem	**57.21**	**63.98**	67.95	76.41	77.78	85.26	+20.6	+21.3
XMem++			**77.11**	**84.56**	**82.87**	**90.20**	**+27.7**	**+26.5**

Table 2. Quantitative results on MOSE [8] training dataset. $\mathcal{J}$ and $\mathcal{F}$ mean Jaccard index and boundary F-score correspondingly, as defined in [30]. TBD$_{DAVIS}$ and TBD$_{YT}$ stand for TBD [7] model trained on DAVIS [30] and YouTube-VOS [41] datasets accordingly. Since the training subset of MOSE dataset includes some very short videos, we only considered videos with $>= 50$ frames each for comparison. Results from a total of 722 videos are presented. $\lvert\mathcal{D}_{1\to10}\rvert$ denotes the increase in segmentation quality from 1 to 10 annotated frames.

The behavior of our annotation candidate selection module is demonstrated in Fig. 7. Often there is more than one possible target in the video, so selecting frames for the right one is important. A video is presented in rows 1 and 2 of Fig. 7, where two people change their head pose, but one at a time. Our algorithm successfully adapts the recommended frames based on which person is being segmented. Rows 3 and 4 depict a more complicated video sequence, where the target object has extreme lighting vari-

*One of the videos provided does not have any target objects on frame #0, which an unsupported use-case for some of the models used, so only 49 out of 50 videos were included in the evaluation.

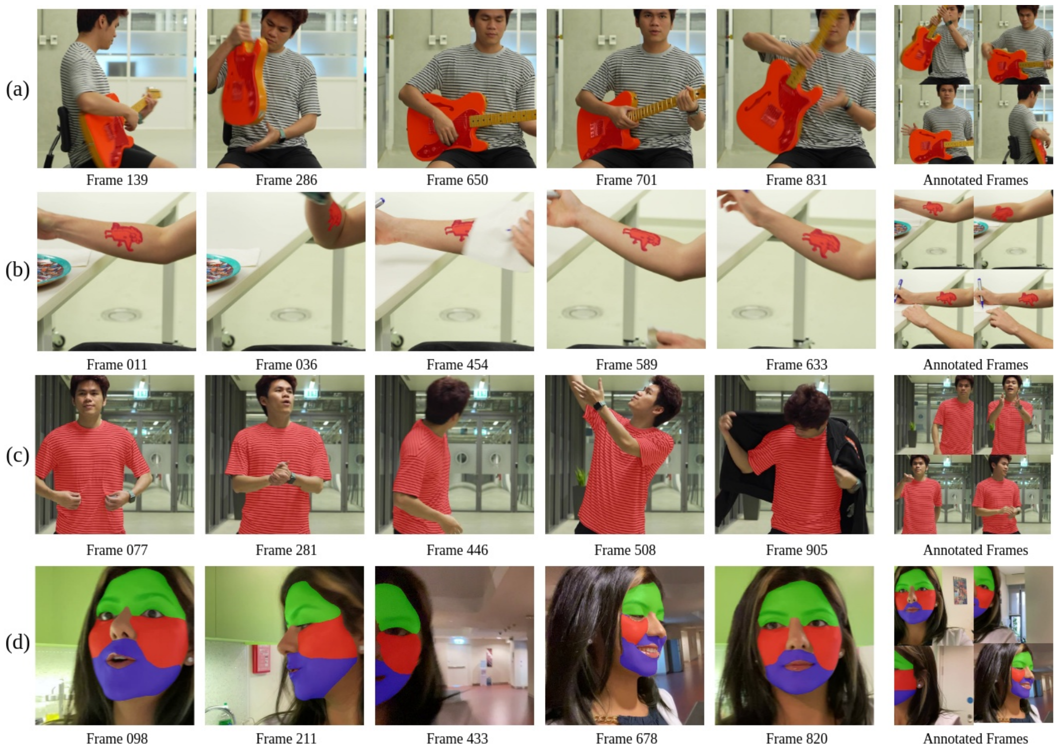

Figure 6. Results of our method **XMem++**. Here we show some use cases that can be used in the industry such as changing the front of a guitar for color purposes (row a), deformable objects such as tattoos (b) or shirts (c), multi-region with no explicit boundary (d).

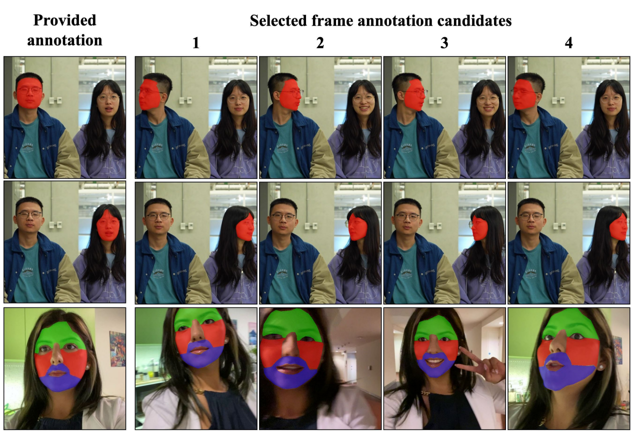

Figure 7. Demonstrated behavior of our target-aware frame selection algorithm. Rows 1 and 2 depict the same video sequence, but with different target annotation. Numbers indicate the importance ranking produced by the algorithm.

	XMem		**XMem++**		# Annotations		$\mathbb{E}$
Sequence	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	XMem	**XMem++**	
Vlog 3-part	85.43	89.70	**85.45**	**89.89**	45	10	4.5×
Lips	**87.24**	**94.79**	86.93	94.43	45	10	4.5×
Half face	93.26	98.13	**93.31**	**98.35**	50	10	5×

Table 3. Quantitative results on a subset of **PUMaVOS** dataset. $\mathcal{J}$ and $\mathcal{F}$ mean Jaccard index and boundary F-score correspondingly, as defined in [30]. $\mathbb{E}$ is the frame usage efficiency of **XMem++** compared to XMem.

ations, rapid movement, disappearances, pose and expression variations (whenever applicable). The algorithm selects frames that capture these varieties without repeating or selecting frames with no target object. Our experiments indicate that the selected annotation candidates are very similar to those chosen by expert users, which makes it practical in real-life scenarios where an end user is likely to work with multiple videos at the same time, thousands of frames long, and rewatching them for multiple rounds to select the frames manually is often infeasible.

Comparison. We compare our results with 3 SOTA methods in Table 2, and with 2 in Table 1, and Fig. 8. We demonstrate that our model produces smooth and temporally continuous segmentation in rows 3-6 of Fig. 8, where both other methods produce incorrect masks missing a part of the target object. In rows 1-3 of Fig. 8 our method successfully segments challenging multi-part regions of the face, that are mostly not visually aligned with the low-level image cues, and the masks produced by TBD and XMem are “bleeding” into the neighbouring regions, as well as have sharp,

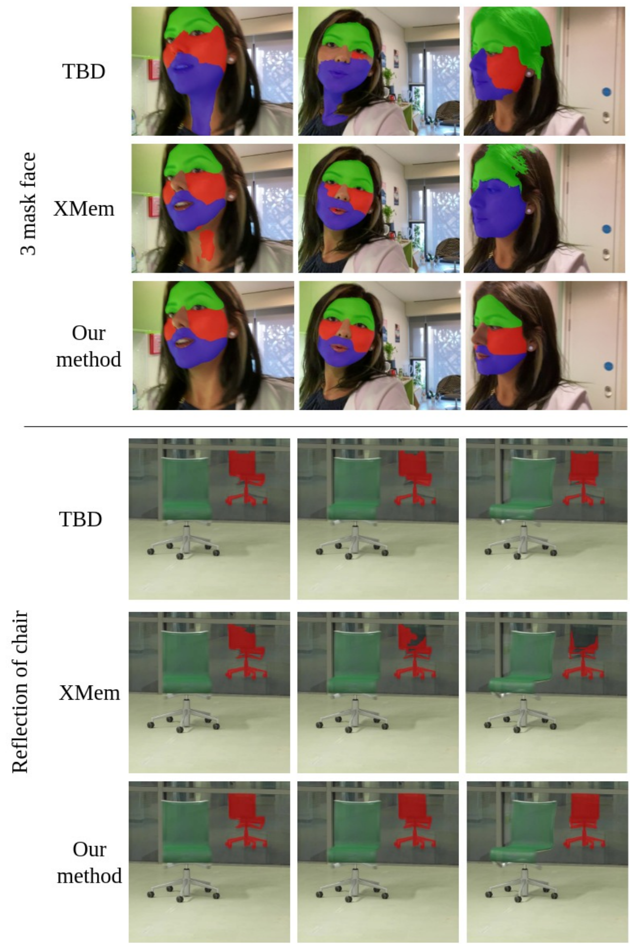

Figure 8. Comparison of our method with TBD[7] and XMem[4] with only 6 frames provided (0.6% of total 952 frames).

"torn"-looking edges. In Tables 2 and 1 we demonstrate that our method results in higher segmentation quality given the same frame annotations, and is at least $2\times$ as efficient in the number of annotations necessary on generic videos. We additionally show in Table 3 that in more challenging practical sequences **XMem++** can be up to $5\times$ more efficient.

Limitations. The segmentation quality of our method sometimes suffers on blurry frames with quickly moving objects, and the similarity measure for the annotation candidate selection is not well-defined on such data either. With multiple similar/identical objects in the frame, the method can sometimes switch to the wrong target if they occlude each other. Use cases of extreme deformation (clothing) and high-detail objects (hair) remain an active challenge. Visual illustrations are provided in the Appendix.

Performance. In the original XMem, given n frames, the processing time is bound by $O\left(\frac{n^2 k}{z} + \frac{n}{q}\right)$, where k is the maximum size the working memory (typically $k = 100$), q is the memory insertion frequency (typically $q = 5$), and z is a compression rate for long-term memory ($z > 600$) [4].

Given m annotated frames, **XMem++** loads them into the permanent memory (static $+m$ factor), with the working memory size $= k + m$, thus processing time is $O\left(\frac{n^2(k+m)}{z} + \frac{n}{q} + m\right)$. In practice m is likely to be small, $m \leq 20$, thus $m \leq 0.2k$, having a slowdown of $< 1.2\times$ on memory readout, and an even smaller effect on the overall segmentation process. On RTX-3090 GPU with a 500-frame video and 5 annotations provided, at 854×480 resolution, **XMem++** yields 32 FPS (35 FPS excluding loading the frames into permanent memory), and XMem yields 39 FPS. Total memory usage only increases by a static factor of $+m$, as we store m additional annotations.

6. Discussion

We introduced a highly robust semi-supervised and interactive video segmentation framework, **XMem++**, with automatic next best frame prediction for user annotation. We have shown that by introducing a permanent memory module to XMem [4], efficient usage of multiple annotated frames is possible, for segmenting a particular object or region, even with drastic changes in the appearance of the object. Our approach achieves better segmentation results than current SOTA VOS methods with significantly fewer frame annotations (in our experiments, up to $5\times$ fewer annotations in highly challenging cases). Our approach further demonstrates the ability to reliably segment partial regions of an object (e.g., the left half of a face) with only a few frame annotations, which is a notoriously difficult task for any existing segmentation methods. As highlighted in our accompanying video, even for highly challenging and long scenes, our masks are temporally smooth without the need for additional post-processing. Hence, our method is suitable for production use cases, such as rotoscoping, where accurate region segmentation and minimal user input is needed.

Our proposed solution is also suitable for non-expert users, as it suggests the next best frame for the user to annotate using an effective yet simple attention-based algorithm. Our experiments indicate that the predicted frames are often very similar to those chosen by expert users, which is always superior to randomly chosen ones. We also show that our framework can be conveniently used to collect and annotate a new dataset, **PUMaVOS**, covering challenging practical segmentation use-cases, such as partial segmentations, multi-object-part segmentation, complex lighting conditions, which cannot be found in existing datasets.

Future Work. While our framework significantly improves the current SOTA within the context of IVOS, we believe that further reduction in frame annotations and complex shape segmentations is possible. In particular, we plan to investigate methods that incorporate dense scene correspondences and on-the-fly generative data augmentation of the segmented regions, which can even be used to improve the robustness of the frame prediction further.

References

[1] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video, 2022. 3

[2] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 777–794. Springer, 2020. 2

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[4] Ho Kei Cheng and Alexander G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model, 2022. 1, 2, 3, 4, 5, 8, 12

[5] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion, 2021. 2

[6] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021. 2

[7] Suhwan Cho, Heansung Lee, Minhyeok Lee, Chaewon Park, Sungjun Jang, Minjung Kim, and Sangyoun Lee. Tackling background distraction in video object segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 446–462. Springer, 2022. 6, 8

[8] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2302.01872*, 2023. 3, 5, 6

[9] Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Capsulevos: Semi-supervised video object segmentation using capsule routing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8480–8489, 2019. 2

[10] Qingnan Fan, Fan Zhong, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM Trans. Graph.*, 34(6):195–1, 2015. 3

[11] Wenbin Ge, Xiankai Lu, and Jianbing Shen. Video object segmentation using global and instance embedding learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16836–16845, 2021. 2

[12] Brent A. Griffin and Jason J. Corso. Bubblenets: Learning to select the guidance frame in video object segmentation by deep sorting frames, 2019. 3, 4

[13] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Guided interactive video object segmentation using reliability-based attention maps, 2021. 3, 4

[14] Lingyi Hong, Wenchao Chen, Zhongying Liu, Wei Zhang, Pinxue Guo, Zhaoyu Chen, and Wenqiang Zhang. Lvos: A benchmark for long-term video object segmentation. *arXiv preprint arXiv:2211.10181*, 2022. 3, 6, 13

[15] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8879–8889, 2020. 2

[16] Kenneth E. Iverson. *A Programming Language*. John Wiley & Sons, New York, 1962. 3

[17] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 656–671. Springer, 2014. 3

[18] Meng Lan, Jing Zhang, Lefei Zhang, and Dacheng Tao. Learning to learn better for video object segmentation. *arXiv preprint arXiv:2212.02112*, 2022. 2

[19] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1332–1341, 2022. 2

[20] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 735–750. Springer, 2020. 2

[21] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3430–3441. Curran Associates, Inc., 2020. 2

[22] Zhihui Lin, Tianyu Yang, Maomao Li, Ziyu Wang, Chun Yuan, Wenhao Jiang, and Wei Liu. Swem: Towards real-time video object segmentation with sequential weighted expectation-maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1362–1372, 2022. 2

[23] Yong Liu, Ran Yu, Fei Yin, Xinyuan Zhao, Wei Zhao, Weihao Xia, and Yujiu Yang. Learning quality-aware dynamic memory for video object segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 468–486. Springer, 2022. 2

[24] Yunyao Mao, Ning Wang, Wengang Zhou, and Houqiang Li. Joint inductive and transductive learning for video object segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9670–9679, 2021. 2

[25] Tim Meinhardt and Laura Leal-Taixé. Make one-shot video object segmentation efficient again. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10607–10619. Curran Associates, Inc., 2020. 2

[26] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2013. 3

[27] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. *CoRR*, abs/1904.00607, 2019. 1, 2

[28] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Fast user-guided video object segmentation by interaction-and-propagation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5247–5256, 2019. 2

[29] Hyojin Park, Jayeon Yoo, Seohyeong Jeong, Ganesh Venkatesh, and Nojun Kwak. Learning dynamic network using a reuse gate function in semi-supervised video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8405–8414, 2021. 2

[30] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 3, 6, 7

[31] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 3

[32] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation: Dataset and challenge. 2021. 3

[33] S. Avinash Ramakanth and R. Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 376–383, 2014. 2

[34] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[35] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 629–645, Cham, 2020. Springer International Publishing. 2

[36] The Creative Cloud Team. Moving art: How to create a rotoscope animation in photoshop cc, Sep 2017. 1, 6

[37] Sudheendra Vijayanarasimhan and Kristen Grauman. Active frame selection for label propagation in videos. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2

[38] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, and Song Bai. Swiftnet: Real-time video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1296–1305, 2021. 2

[39] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1140–1148, 2018. 2

[40] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1286–1295, 2021. 2

[41] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark, 2018. 3, 6

[42] Xiaohao Xu, Jinglu Wang, Xiao Li, and Yan Lu. Reliable propagation-correction modulation for video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2946–2954, 2022. 2

[43] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2

[44] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2

[45] Zhaoyuan Yin, Jia Zheng, Weixin Luo, Shenhan Qian, Hanling Zhang, and Shenghua Gao. Learning to recommend frame for interactive video object segmentation in the wild, 2021. 3, 4

[46] Ye Yu, Jialin Yuan, Gaurav Mittal, Li Fuxin, and Mei Chen. Batman: Bilateral attention transformer in motion-appearance neighboring space for video object segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 612–629. Springer, 2022. 2

[47] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6949–6958, 2020. 2

[48] Tianfei Zhou, Fatih Porikli, David Crandall, Luc Van Gool, and Wenguan Wang. A survey on deep learning technique for video segmentation, 2021. 1, 2