Learning the What and How of Annotation in Video Object Segmentation

Thanos Delatolas, Vicky Kalogeiton, Dim P Papadopoulos

Learning the What and How of Annotation in Video Object Segmentation

Thanos Delatolas[1,2] Vicky Kalogeiton[3] Dim P. Papadopoulos[1,2]
[1] Technical University of Denmark [2] Pioneer Center for AI
[3] LIX, Ecole Polytechnique, CNRS, Institut Polytechnique de Paris
atde@dtu.dk, vicky.kalogeiton@lix.polytechnique.fr, dimp@dtu.dk
https://eva-vos.compute.dtu.dk/

Abstract

Video Object Segmentation (VOS) is crucial for several applications, from video editing to video data generation. Training a VOS model requires an abundance of manually labeled training videos. The de-facto traditional way of annotating objects requires humans to draw detailed segmentation masks on the target objects at each video frame. This annotation process, however, is tedious and time-consuming. To reduce this annotation cost, in this paper, we propose EVA-VOS, a human-in-the-loop annotation framework for video object segmentation. Unlike the traditional approach, we introduce an agent that predicts iteratively both which frame ("What") to annotate and which annotation type ("How") to use. Then, the annotator annotates only the selected frame that is used to update a VOS module, leading to significant gains in annotation time. We conduct experiments on the MOSE and the DAVIS datasets and we show that: (a) EVA-VOS leads to masks with accuracy close to the human agreement 3.5× faster than the standard way of annotating videos; (b) our frame selection achieves state-of-the-art performance; (c) EVA-VOS yields significant performance gains in terms of annotation time compared to all other methods and baselines.

1. Introduction

Video object segmentation (VOS) is the task of segmenting and tracking objects of interest in videos [5, 7, 14–16, 29, 30, 43, 49, 69, 70, 82–84]. VOS is a central task for video understanding and enables various applications including video editing [3, 32], video synthesis [75, 76], and video decomposition [85]. Training a VOS model requires videos in which the target objects have been manually annotated with object segmentation masks [5, 7, 14–16, 43, 49, 83, 84]. This process is expensive and labor-intensive as it requires humans to manually draw a mask at each video frame, which requires 80 seconds per object per frame [42]. For instance, annotating only one object in a short 10-second video would require more than 5 hours. The resource-intensive manual annotation bottlenecks the feasibility of building large-scale VOS datasets, indispensable for training effective models. In turn, this restricts the democratization of annotated video data, thus limiting advances in video understanding.

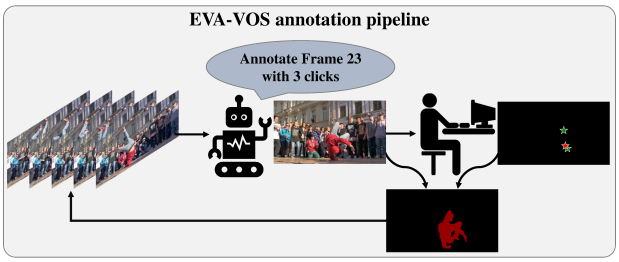

Figure 1. **EVA-VOS.** In contrast to the traditional way of annotating objects in videos, we propose to use a human-in-the-loop approach. We introduce an agent that selects the frame that should be annotated ("What to annotate?") and the annotation type (e.g. clicks, object-mask) ("How to annotate?"). Then, we use the weak annotation to predict a mask for the frame and we propagate it to predict masks for the whole video.

To address these limitations, the research community has turned to two solutions. First, sparsely annotating large VOS datasets [20, 22, 59, 77, 80], and secondly, accelerating the annotation process. Regarding the former, the standard way of annotating a VOS dataset [20, 22, 57–59, 66, 77, 80] starts by selecting a subset of frames via a uniform sampling (usually 1-5 fps). Then, these frames are manually annotated by humans who draw a mask at each selected frame. In some datasets, such as VISOR and UVO [20, 77], the sparsely annotated masks are interpolated to predict dense annotation for all frames. However, relying solely on such sparsely annotated large-scale datasets may introduce limitations, especially for applications that demand fine-grained and accurate segmentations throughout the entire video.

For the latter, i.e., to minimize the annotation cost, the common strategy is interactive segmentation using faster annotation types, such as clicks, scribbles, or bounding boxes. Even though many approaches were proposed for

still images [1,6,10,40,44,53,55] that led to the creation of a larger image segmentation dataset [6], there is limited work in the video domain [8, 11, 13, 19]. The most relevant to our work is Caelles et al. [8] who propose a human-in-the-loop interactive VOS. The annotator provides a scribble at a frame and a VOS method predicts a mask for each frame. Then, the annotator iteratively selects the frame with the worst segmentation quality and provides scribbles. However, despite its innovation, this approach has two main limitations. First, it is unrealistic as the annotator can not identify the worst frame, and even if they could, it would require significant time [28], which defeats the purpose of minimizing the cost. Second, in challenging frames, low-cost annotation types (e.g. scribbles or clicks) are insufficient to create a good mask, and drawing the full mask is required.

To overcome these limitations, we propose **EVA-VOS**, a human-in-the-loop pipeline (Fig. 1). Our contribution is the introduction of an agent that predicts iteratively which *frame* should be annotated (frame selection:"What to annotate?") and which *annotation type* should be used (annotation selection:"How to annotate?"). Our agent is trained to maximize the annotation impact on the segmentation quality while minimizing the annotation cost.

For the frame selection, we train a model to regress the quality of a segmentation mask. Then, we select the frame that has the maximum distance from its closest pre-annotated frame. For the annotation selection, we train a deep RL policy that selects an annotation type (action) by maximizing the fraction of the segmentation quality improvement over the annotation time of the annotation type (reward). Our pipeline iterates between (a) selecting the next frame for annotation and the optimal annotation type, (b) asking annotators to improve a segmentation mask, and (c) predicting new object masks for all frames (Fig. 2).

To evaluate our method, we conduct experiments on the MOSE [22] and DAVIS [58] datasets. We first evaluate each stage of the agent (frame and annotation selection) independently and then we show the final results of our full pipeline. Our results show that (a) EVA-VOS leads to masks with accuracy close to the human agreement 3.5× faster than the standard way of annotating a VOS dataset; (b) Our frame selection method achieves state-of-the-art performance; (c) EVA-VOS yields significant performance gains in terms of annotation time compared to other strong baselines.

2. Related Work

Semi-Supervised Video Object Segmentation (VOS). VOS methods aim to segment a specific object throughout a video given the object mask in the first frame. Existing VOS methods can be divided into three categories: online fine-tuning [7, 46, 47, 71, 78, 82], propagation-based [2, 12, 17, 30, 31, 56, 69, 83, 84] and matching based [14, 16, 29, 43, 49, 65, 70, 79] methods. Online fine-tuning methods overfit on the object mask of the first video frame. Propagation-based methods use the mask of the previous frame to generate the mask of the current frame. This involves progressively passing on features of the target object from one frame to the next. Matching-based algorithms store features of the target object and classify each pixel of the current frame using similarities in the feature space. Current state-of-the art methods: STCN [16], XMem [14], AOT [83], DeAOT [84] achieve remarkable results in the traditional VOS benchmarks (e.g. DAVIS [57,58] and YouTube-VOS [80]). However, their performance drops notably on new challenging benchmarks: VISOR [20], MOSE [22] and VOST [66]. This is because these datasets contain severe occlusions, disappearance/reappearance of objects, and object transformations. Instead, our method identifies where these methods fail and provides extra annotations to refine the results.

Interactive VOS (iVOS) tackles the human-in-the-loop setting, where the annotator provides quick input (e.g. scribbles [8, 11], points [13, 72], text [25, 34, 64]) to a VOS method instead of detailed object masks as in semi-supervised VOS. At each annotation round, the annotator selects the frame with the worst segmentation quality and provides a scribble to refine the output mask. The scribble serves as the input to the VOS method, and the process continues iteratively. ScribbleBox [11] builds on top of iVOS by including a prior step where a tracker [73] predicts a bounding box for each frame and then the annotator inspects and refines them. We argue that iVOS is unrealistic as the annotator is unable to find the worst frame and even if they did, they would need a significant amount of time, which defeats the purpose of iVOS.

VOS dataset annotation. Manually annotating object masks is time-consuming. Annotating a VOS dataset requires per frame object masks, linking object masks over time, and quality assurance [20, 22, 57–59, 66, 77, 80]. As a result, VOS datasets have fewer annotations than image datasets. For instance, the densely annotated DAVIS 17 [58] VOS dataset contains only 13K annotations, vs the 500K annotations in COCO [42]. Youtube-VOS [80] scaled up the number of annotations to 197K by annotating every 5 frames. UVO [77] consists of 200K annotations at 1 fps for training and 30 fps for validation. The sparsely annotated masks are interpolated using STM [49] to cover all frames, and annotators correct the interpolated masks. Similarly, VISOR [20] has 271k annotations. Finally, OVIS [59] and MOSE [22] are annotated every 5 frames without any interpolation resulting in 296K and 431K masks, respectively. These datasets have a similar time-consuming annotation pipeline, constituting a bottleneck. Instead, our method improves this by selecting both the frame and the annotation type (box, clicks, mask) and significantly reduces the cost.

Segment Anything (SAM) was recently introduced [37] and it immediately inspired many new methods [18, 39, 45,

48,61,74,81]. SAM was trained on SA-1B [37] which is the largest dataset for image segmentation as it consists of over 1 billion masks. Track Anything (TAM) [81] uses SAM and XMem [14] to do interactive VOS with clicks. In particular, the initial mask at the target object is generated from SAM using click prompts and XMem tracks the target object throughout the video. If the quality of the output masks from XMem is low, either SAM is prompted with clicks that are extracted from the affinities of XMem or the annotator prompts SAM to refine the segmentation mask that will be propagated by XMem. Our method differs from TAM because it finds both the frame and the annotation type that SAM will be prompted with to maximize the performance of the propagation while minimizing the annotation cost.

Frame Selection in VOS. The goal of this task is to find a set of frames for the annotator to annotate that maximizes the overall video segmentation quality. BubbleNets [24] predict only the initial frame instead of always using the first frame, which is the standard in the field. GIS-RAmap [28], XMem++ [5] and IVOS-W [86] iteratively predict a frame at each annotation round of the iVOS setting. Both GIS-RAmap [28] and XMem++ [5] are VOS models that first segment frames and then predict the next frame for annotation. GIS-RAmap [28] uses the pixel-wise scores of each frame, while XMem++ [5] uses the key features of each frame and all previously annotated frames to predict the next frame. IVOS-W [86] uses reinforcement learning (RL) and is also the most closely related work to ours, as it is not based on any VOS model. Both IVOS-W and our method regress the quality of each frame to predict the next one. However, IVOS-W assumes explicit information about the target object because it extracts the region of the image around it [5]. Instead, our method uses the entire frame and regresses its quality. In Sec. 5.1, we modify IVOS-W [86] to work for different annotation types other than scribbles, and compare it to our method. To the best of our knowledge, our method (EVA-VOS) is the first in the VOS field that predicts the annotation type for each frame.

3. Method

In this work, we propose EVA-VOS, a human-in-the-loop pipeline to annotate videos with segmentation masks using as little annotation as possible (Fig. 2). EVA-VOS consists of four stages: (a) Mask Propagation (Sec. 3.1), (b) Frame Selection (Sec. 3.2), (c) Annotation Selection (Sec. 3.3), (d) Annotation and Mask Prediction (Sec. 3.4).

More formally, at each iteration t, the mask propagation receives the input video $V = \{f_1, f_2, \dots, f_N\}$ of N frames and a set K containing all previously annotated frames to predict a new set of masks $\mathbf{M}^t = \{M_1^t, M_2^t, \dots, M_N^t\}$ for all frames. Then, the frame selection determines the frame f_* that should be annotated given V and $\mathbf{M}^t$. The annotation selection determines the most suitable annotation type a_{f_*} from a pool of candidate annotation types $A = \{a_1, a_2, \dots, a_L\}$. For annotation types, we consider both the case where the annotator manually draws a complete mask (*'mask drawing'*), and the case of weak annotations, where the human intervention is much faster, e.g., *'corrective clicks'*, *'bounding boxes'*, *'scribbles'*, etc. Finally, the annotator annotates f_* with a_{f_*}, and the annotation is passed on to the mask prediction, where a new mask $M_{f_*}^{t+1}$ is predicted and added to K. Note that at $t = 0$, the annotator selects the target object and draws a mask on f_1.

3.1. Mask propagation

At this step, we predict a set of masks $\mathbf{M}^t$ for all frames using all annotated masks from the set K. For this, we use a pre-trained VOS [15] module which takes as input the video V and the masks K and predicts a mask M_i for each frame.

3.2. Frame selection

Given a video V, the predicted masks of each frame $\mathbf{M}^t$, and the set K containing all previously annotated frames, our aim is to find the frame to be annotated f_* at iteration t in order to have the highest improvement of the video segmentation quality at iteration $t + 1$. Annotating f_* will enhance the segmentation quality of the video V via mask propagation, and as a result, we want to select the frame that will have the most impact on the mask propagation stage (Sec. 3.1). Intuitively, we want to select frames that maximize the diversity among the selected ones and at the same time have low segmentation quality so that we maximize the impact on the final performance. We first train a model to assess the segmentation quality of each frame, and then we use the learned frame representations to select f_*.

Architecture. To assess the mask quality of each frame, we introduce the Quality Network (QNet) which takes in a frame f_i and its corresponding mask M_i^t and performs mask quality classification into B classes, where 0 represents the worst quality and $B-1$ the best. The value of B determines the number of bins in which the segmentation quality is divided. QNet consists of two image encoders [27] in parallel branches, one for the frame f_i and one for the mask M_i^t. The embeddings from each encoder are then concatenated and fed into a linear classifier with B outputs.

Training. We train QNet in a supervised way with a cross-entropy loss on a simulated training set. To generate a realistic training set, we simulate a number of iterations with EVA-VOS (Fig. 2); at each iteration, we compute the segmentation quality of each frame and assign a quality label to each mask M_i^t. We simulate our training set with random and oracle selections at each iteration as follows: A random selection chooses f_* randomly, excluding frames in K, while an oracle selection chooses the frame f_* with the worst segmentation quality.

Selected frame f_*. We select f_* as the one with the max-

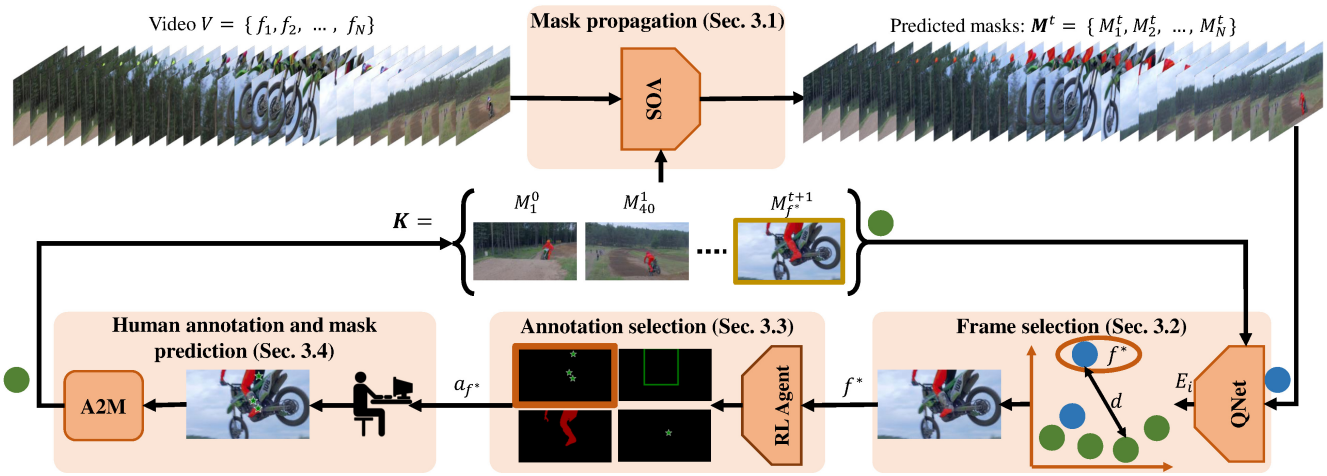

Figure 2. **EVA-VOS**. At each iteration t, Mask propagation (Sec. 3.1) receives a video V of N frames and a set K containing all previously annotated frames to predict a new set of masks $\mathbf{M}^t = \{M_1^t, M_2^t, \ldots, M_N^t\}$ for all frames. Subsequently, the Frame selection (Sec. 3.2) stage selects the frame f_* that should be annotated given the video V, the predicted masks $\mathbf{M}^t$ and all previously annotated frames K. The Annotation selection (Sec. 3.3) takes as input the selected frame f_* and its corresponding mask to predict the most suitable annotation type a_{f_*}. Finally, in the Human annotation and mask prediction (Sec. 3.4) stage, the annotator interacts with f_* using the annotation type a_{f_*} and A2M (in this work, we use SAM [37]) predicts the new mask $M_{f_*}^{t+1}$ of the frame f_*, which is then added to the set K.

imum distance in the feature space from its closest previously annotated frame. To this end, we first extract embeddings E_i from QNet. Next, we compute the L2 distance between each embedding of a frame j in K and all frames of V. Finally, we assign the minimum distance to each embedding i, and we select the frame with the maximum distance. This process can be mathematically described as:

$$f_* = \underset{i \in \{1,2 \ldots N\}}{\arg\max} \left\{ \min_{j \in \{1,2 \ldots t\}} \{d(E_i, E_j)\} \right\} \tag{1}$$

3.3. Annotation selection

Given a pool of annotation types $A = \{a_1, a_2, \ldots, a_L\}$, the goal of this step is to choose the most suitable type a_{f_*} for f_*. Following [9, 38], we formulate this problem as a Markov Decision Process and train a model using reinforcement learning (RL). The model observes the image of f_* and its predicted mask $M_{f_*}^t$ and predicts the most suitable annotation type a_{f_*}. This annotation type is then utilized by the annotator to generate a new mask $M_{f_*}^{t+1}$ for f_*. The annotation is performed iteratively (e.g. 3 clicks are performed one by one). Therefore, we denote the annotation iteration as g. The input $M_{f_*}^t$ has an initial segmentation quality SQ_1 ($g = 1$).

Environment. To give our model the ability to play the annotation selection game, the environment consists of the Human annotation and mask prediction stage (Sec. 3.4). The state of the environment consists of f_* and its mask. Each step g yields SQ_g using the input action, which represents an annotation type from A.

Reward. The reward function reflects the trade-off between the quality of $M_{f_*}^t$ and the cost of the annotation type. Each $a \in A$ requires a different annotation cost denoted by θ_a. The reward at g is formulated by comparing SQ before and after annotation, divided by the total cost tc at g which is the sum of the costs θ_a of all annotation types until g:

$$r = \frac{\text{SQ}_{g+1} - \text{SQ}_g}{tc} \quad . \tag{2}$$

This equation captures the improvement of SQ, normalized by the total cost, and our model is trained to maximize this improvement while minimizing the annotation cost.

Architecture. The model has two image encoders [27, 37] in parallel branches, one for the frame f_i and one for the mask $M_{f_*}^t$. The extracted embeddings from each encoder are then concatenated and fed into two linear layers. The first layer has L outputs (possible annotation types), while the second layer has one output for the RL value.

Training. Following its success in several other tasks [50, 52], we use Proximal Policy Optimization [63] (PPO) to train our model. At training, we use the simulated masks described in Sec. 3.2. At each iteration t, we use f_* and its corresponding mask $M_{f_*}^t$ to play the annotation selection game and train our agent. We perform multiple environment steps and the process terminates when we reach the maximum steps or the type of drawing a mask is selected.

Video Ranking. When EVA-VOS is used to annotate a collection of videos, similar to active learning, we use the predicted value pv of our RL agent to estimate the improvement of each annotation at each video. This allows us to

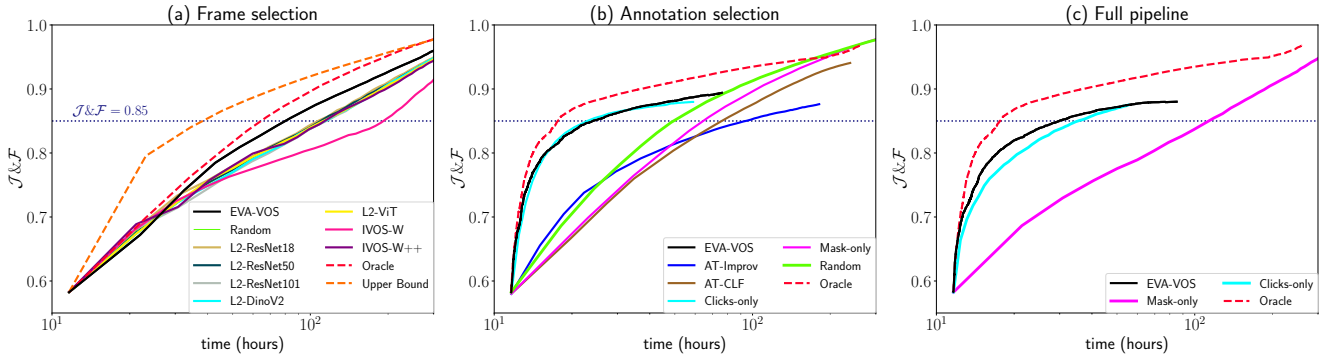

Figure 3. **Experimental results on MOSE.** We report the $\mathcal{J}\&\mathcal{F}$ accuracy as a function of annotation time in hours. (a) The effect of the frame selection stage (for fair comparison we use the same annotation type for all approaches). (b) The effect of the annotation selection stage using the same frame selection (oracle) for all approaches. (c) The results of our full pipeline.

rank the videos and perform more annotation iterations in videos where the RL value is higher. To this end, for each iteration t, we first calculate the score s of each video and perform an annotation iteration to the video with the maximum s. The score for each video is defined as a function of pv, the annotation iteration t, and annotation cost θ_a:

$$s = \frac{pv \cdot \gamma^t}{\theta_a} + c \quad , \tag{3}$$

where $\gamma < 1$ and it allows us to prioritize videos with less annotations (smaller t). γ is necessary since our agent has no information about the number of annotations of each video. Finally, c scales the score s into a positive range.

3.4. Human annotation and mask prediction

Here, the annotator interacts with the selected f_* to create the input annotation type a_{f_*}. When a_{f_*} is *'mask drawing'*, the annotator draws a detailed $M_{f_*}^{t+1}$. Otherwise, i.e. clicks, this step predicts a new mask $M_{f_*}^{t+1}$ using a pre-trained annotation-to-mask (A2M) model [37]. A2M predicts $M_{f_*}^{t+1}$ based on the input weak annotation.

A2M predicts the new mask $M_{f_*}^{t+1}$ of the selected frame f_* with the input annotation type a_{f_*}. There are various models that predict a segmentation mask given a weak annotation type [4, 6, 33, 37, 41]. Since our method is independent of this model, we opt for the recently introduced Segment Anything Model (SAM) [37]. SAM can take as input the following annotation types: $A = \{$clicks (positive and negative), bounding boxes, masks, and text$\}$ to predict $M_{f_*}^{t+1}$. When a_{f_*} is anything but a number of clicks, SAM takes as input the annotation type a_{f_*} and the current mask $M_{f_*}^{t}$ of f_* to predict $M_{f_*}^{t+1}$. Otherwise, when a_{f_*} is a number of clicks, SAM predicts $M_{f_*}^{t+1}$ recursively. In particular, at each iteration, SAM takes as input one click and its own previous mask prediction to output a new segmentation mask. This process is repeated as many times as the number of clicks and the final prediction is the new mask $M_{f_*}^{t+1}$ of the selected frame f_*. It should be noted that this process is initialized with $M_{f_*}^{t}$ which is predicted by the VOS [15] module and SAM can only take as input a mask from its own previous prediction. Therefore, we simulate clicks extracted by $M_{f_*}^{t}$ and input them into SAM to generate a similar mask to the predicted by the VOS module.

4. Experimental setting

Datasets. **DAVIS 17** contains 60 train and 30 validation videos. It provides high-quality annotated masks for each frame. **MOSE** inherits videos from OVIS [59] and it is one of the largest available VOS dataset with 2149 videos, out of which only 1507 come with available ground-truth masks. In our work, to model long-range interactions we only consider videos with 15 to 104 frames leading to MOSE-long dataset with 1166 videos. We split it into 800 training, 150 validation, and 216 test videos.

In our experiments, EVA-VOS is pre-trained on ImageNet [21] and trained on MOSE-long, unless stated otherwise. Following the trend of zero-shot testing [60], to examine cross-dataset generalization, we evaluate EVA-VOS on the MOSE-long test set and on the DAVIS validation set.

Metrics. To measure the segmentation quality of the predicted masks, we use both the intersection-over-union $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ [57]. For this, we follow [8] and use the curve of $\mathcal{J}\&\mathcal{F}$ vs time. We also report the annotation time in hours at $\mathcal{J}\&\mathcal{F} = \{0.75, 0.80, 0.85\}$ (different levels of human annotation agreement for instance segmentation [6, 26, 36, 87]), and the average $\mathcal{J}\&\mathcal{F}$ up to 200 hours of annotation time. We consider 80 sec for drawing an object mask [42] and 1.5 sec for each click plus 1 sec of overhead for the annotator to locate the object [4, 6, 54].

Implementation details. QNet consists of two ResNet-18 [27]. We train it using SGD with lr=10^{-5}, batch size 64, 30 epochs, with B=20. The frame branch of the RL

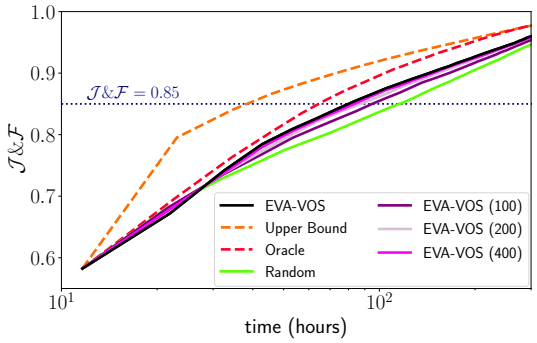

Figure 4. **Ablation study** on the number of training videos in the frame selection stage.

agent is the image encoder of SAM [37] while the mask branch is a ResNet-18 [27]. The RL agent is trained using Adam [35] and lr=10^{-5} for 50K iterations. For the video ranking, we estimate the hyper-parameters of Eq. (3) in the MOSE-long validation set. For all experiments, we use γ to 0.7 and c to -0.04. For the VOS module, we use a modified version of MiVOS [15], where we discard the original interaction module [15] (as it only works with scribbles) and replace the propagation module with STCN [16] for faster propagation. To better examine the effect of frame and annotation selections, we pre-train this modified version only on YouTubeVOS [80] and not on DAVIS [58]. In our experiments, we consider two annotation types: *'mask drawing'* and *'corrective clicks'* [6]. For *'corrective clicks'*, the annotator clicks 3 times to improve the given segmentation and determines the number of positive and negative clicks. For simplicity, for the remainder of this work, we denote *'mask drawing'* as Mask and *'corrective clicks'* as Clicks.

Human annotator simulation. In this work, we only perform experiments by simulating the human intervention. Given $M^t_{f_*}$ and the ground-truth mask of f_* m_g, we simulate positive and negative clicks to prompt SAM [37] similar to how a human would. Initially, we identify all false-negative and false-positive pixels between m_g and $M^t_{f_*}$. Then, we determine the connected components of each error region, and the center of the largest component is selected as the click location, whether positive or negative.

5. Experimental results

In all cases, we use figures for evaluation and show the segmentation quality ($\mathcal{J}\&\mathcal{F}$) of the predicted masks for all methods in relation to the annotation time on a log axis. We first evaluate the frame selection and annotation selection (Sec. 5.2) individually and show the results in Fig. 3(a) and (b), respectively. Then, we analyze the results of our full pipeline (Sec. 5.3) and report results in Fig. 3(c). Finally, in Sec. 5.4 we examine the generalization ability of EVA-VOS, by performing a cross-dataset examination (Fig. 6).

Table 1. **Comparison of annotation methods on MOSE [22].** We report the human annotation time in hours for each method to reach different $\mathcal{J}\&\mathcal{F}$ values (0.75, 0.8, 0.85). We also report the average $\mathcal{J}\&\mathcal{F}$ up to 200 hours. At the top of the table, we report the oracle performance of oracle approaches for frame and/or annotation selection. **Bold** is the overall best-performing model, while Underline is the best-performing frame selection approach that uses Mask-only as an annotation type.

Annotation Selection	Frame Selection	Hours at $\mathcal{J}\&\mathcal{F}$ = 0.75	0.80	0.85 ↓	Avg $\mathcal{J}\&\mathcal{F}$ ↑
Mask-only	Oracle*	34.42	45.62	67.64	0.83
Clicks-only	Oracle*	14.05	15.65	22.85	0.87
Oracle*	Oracle*	12.96	14.13	17.63	0.92
Mask-only	IVOS-W [86]	39.37	94.33	192.26	0.78
Mask-only	IVOS-W++	40.53	59.81	113.93	0.79
Mask-only	L2-ResNet50	40.55	59.92	109.42	0.80
Mask-only	Random	40.55	69.60	107.40	0.80
Mask-only	EVA-VOS	32.55	53.26	80.71	0.82
Random	Random	24.08	36.10	65.84	0.85
Clicks-only	Random	15.32	21.22	35.10	0.86
EVA-VOS	**EVA-VOS**	**14.24**	**17.25**	**29.80**	**0.87**

5.1. Frame selection evaluation

Here, we evaluate the frame selection (Sec. 3.2) and display the results in Fig. 3(a). For a fair comparison among all methods, we use only Mask as an annotation type.

Compared methods. We compare our method to a random baseline that selects frames randomly. We also compare EVA-VOS to IVOS-W [86], which is the state-of-the-art frame selection method for VOS. Note that IVOS-W was originally designed to work only under a scribble-based iVOS scenario. Therefore, we modify it here to work for different annotation types, and for a fair comparison, we train this modified version in MOSE-long. Furthermore, we implement and compare to it the IVOS-W++ in which we replace the RNN with a transformer [68] and the double Q-Learning [67] with PPO [63]. We also use powerful image encoders [23,27,51] pre-trained for image classification [62] to compute embeddings in Eq. (1) and we compare the results with QNet. We implement an oracle approach that selects the frame with the worst $\mathcal{J}\&\mathcal{F}$ and an upper bound approach that selects the frame that has the highest impact in the propagation stage after annotating it.

Comparison to the state of the art. Fig. 3(a) shows that most methods have a similar performance close to Random. Instead, EVA-VOS consistently stands out and yields higher $\mathcal{J}\&\mathcal{F}$ and in some cases almost identical to Oracle. We now analyze the results of all methods:

Random is shown as the green line in Fig. 3(a). We run all random baselines 15 times and report the average result.

EVA-VOS (Ours) is shown as the black line in Fig. 3(a).

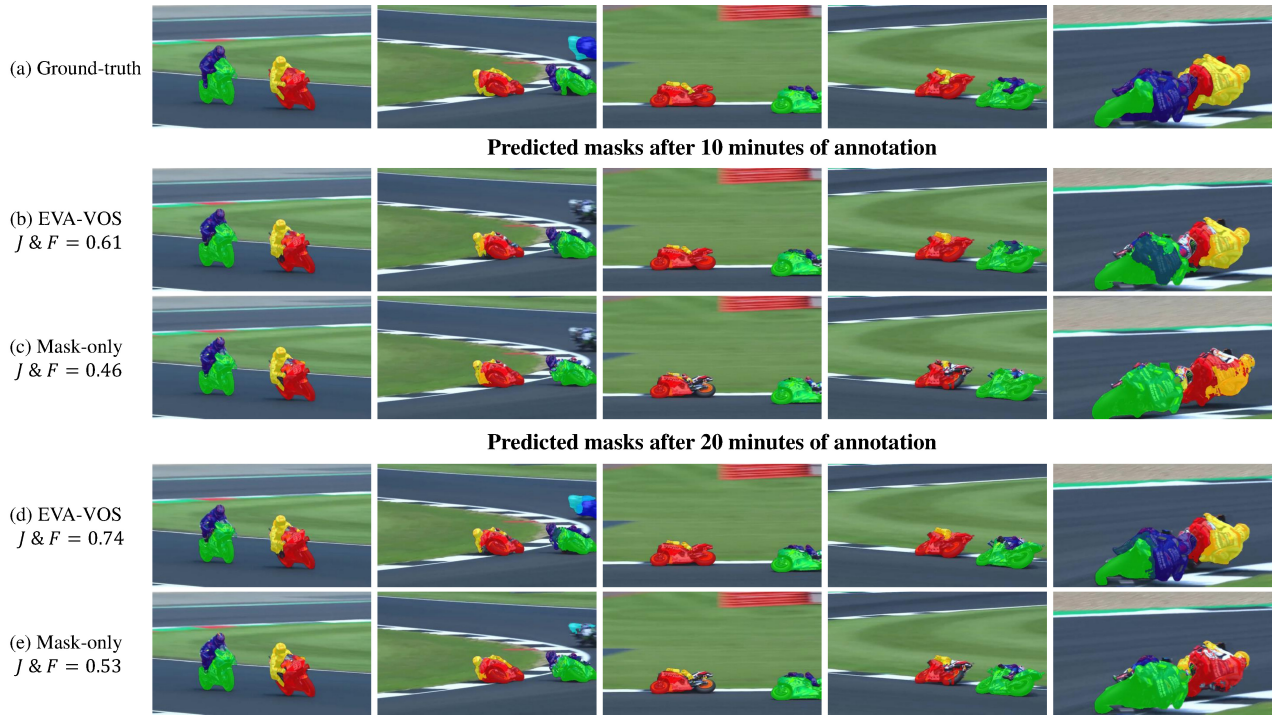

Figure 5. We compare EVA-VOS with Mask-only which resembles the traditional VOS annotation pipeline in one video from MOSE [22]. The row (a) shows the ground-truth masks on 5 frames of a video with six annotated objects (purple, green, yellow, red, blue, and cyan). The rows (b) and (c) show the predicted masks of EVA-VOS and Mask-only after 10 minutes of annotation while the rows (d) and (e) after 20 minutes. We observe that EVA-VOS consistently outperforms Mask-only at different annotation budgets. For example, in the second frame, EVA-VOS correctly segments the blue and cyan objects at 20 minutes, while Mask-only fails. Similarly, in the third and fourth frames, EVA-VOS correctly segments the red object at 10 minutes, while Mask-only fails even after 20 minutes.

Given the same annotation time, our framework consistently outperforms Random. For instance, we achieve $\mathcal{J}\&\mathcal{F}$=0.85 at 80.7 hours, 26.7 hours faster than Random.
State-of-the-art frame selection (IVOS-W [86]) performs significantly worse than our method. We observe that our method reaches $\mathcal{J}\&\mathcal{F}$ of 0.85 2.3× faster than IVOS-W.
IVOS-W++ performs better than IVOS-W but our method achieves $\mathcal{J}\&\mathcal{F}$ of 0.85 1.4× faster.
L2-Encoders yield approximately the same performance as random. This shows that our task-specific QNet learns much better representations and outperforms all pre-trained encoders that have even 28× more parameters.
Oracle is shown as the red dashed line in Fig. 3(a). Interestingly, we observe that for low budgets (up to 40 hours), our method yields almost identical $\mathcal{J}\&\mathcal{F}$.
Upper Bound consistently outperforms the oracle indicating that the frame with the worst $\mathcal{J}\&\mathcal{F}$ is not the most impactful one. Interestingly, the upper bound is only 2.1× faster than our method at $\mathcal{J}\&\mathcal{F} = 0.85$.

Ablation study on the number of training videos. QNet is trained on MOSE-long (800 videos). We retrain it using 100, 200, and 400 videos to examine the impact of the training data. In Fig. 4, we compare all of our frame selection models with the random method as it performs approximately the same as the L2-Encoders and IVOS-W++. We observe that all of our models outperform Random and even with a reduced training dataset, i.e, 400 videos, QNet showcases $\mathcal{J}\&\mathcal{F}$ that closely aligns with the initial model trained with a larger dataset of 800 videos. This reveals the robustness and effectiveness of our training process (Sec. 3.2).

5.2. Annotation selection evaluation

Here, we evaluate only our annotation selection stage. For a fair comparison, we set the frame selection for all approaches to oracle, i.e., the frame with the worst $\mathcal{J}\&\mathcal{F}$ is selected to be annotated at each iteration (results in Fig. 3(b)).
Compared methods. To examine the design choice of RL (Sec. 3.3), we implement two alternatives for annotation selection. The first one is AT-Improv (Annotation Type Improvement), which is trained to regress the improvement of each available annotation type. It selects the annotation type that maximizes Eq. (2). The second one is AT-CLF (Annotation Type Classification), and it is trained to classify each f_* into an annotation type. Furthermore, we compare against approaches that consider only one annotation type (Clicks or Mask). We also compare against a random base-

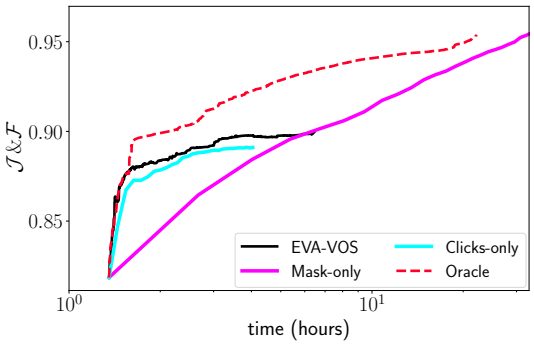

Figure 6. **EVA-VOS results on DAVIS 17.** We report the $\mathcal{J}\&\mathcal{F}$ accuracy as a function of annotation time in hours at log scale.

line that selects a_{f_*} randomly, and an oracle approach that selects a_{f_*} using Eq.(2), i.e., it selects the a_{f_*} that yields the maximum quality improvement normalized by the annotation cost. Moreover, the oracle approach ranks the videos using Eq. (2), while our approach uses Eq. (3).

Comparison to annotation selection methods. In Fig. 3(b) we observe the impact of annotation selection since Clicks-only plateaus at lower $\mathcal{J}\&\mathcal{F}$ while Mask-only is significantly slower. Furthermore, methods that do not select the annotation type wisely, perform worse than Mask-only. We now analyze the results of all methods:

EVA-VOS (Ours) is shown as the black line in Fig. 3(b). It reaches $\mathcal{J}\&\mathcal{F} = 0.85$ in only 29.8 hours.

AT-Improv, AT-CLF (blue and brown lines in Fig. 3(b)) perform significantly worse than EVA-VOS which is trained using RL instead of supervised learning.

Random is shown as the green line in Fig. 3(b). Even though it reaches $\mathcal{J}\&\mathcal{F} = 0.9$ at a similar time as our method, it performs significantly worse at lower budgets (e.g., we yield $\mathcal{J}\&\mathcal{F} = 0.85$ 1.9× faster). *Mask-only* performs consistently worse than random at all budgets, indicating that the traditional way of manually drawing object mask [57, 59, 80] is not a good approach.

Clicks-only performs on par with our method at low annotation budgets. However, it plateaus quickly at lower $\mathcal{J}\&\mathcal{F}$ values and it is not able to reach $\mathcal{J}\&\mathcal{F} = 0.9$, whereas our method can yield higher $\mathcal{J}\&\mathcal{F}$ at larger budgets.

Oracle (red dashed line in Fig. 3(b)) performs on par with our method at low budgets. Oracle performs better in very high annotation budgets that reach high $\mathcal{J}\&\mathcal{F}$ above 0.85.

5.3. Frame and Annotation selection evaluation

We evaluate here our full pipeline, showing the effect of both selection modules (Fig. 3(c) and Tab. 1).

Compared methods. Similar to Sec. 5.2, we compare our method to Clicks-only and Masks-only which select a random frame and consider only one annotation type. Additionally, we compare against Oracle, which uses both oracle frame selection and annotation selection.

Comparison of annotation methods. In Fig. 3(c), we examine the impact of both selection modules. Overall, we observe that a large amount of annotation time can be saved when annotating with more than one type and wisely selecting frames. We now present the results of all methods:

EVA-VOS (Ours) is shown as the black line in Fig. 3(c) and yield a $\mathcal{J}\&\mathcal{F}$ of 0.85 in 29.8 hours (also, last row in Tab. 1).

Oracle uses both oracle frame selection and annotation selection and shows the trade-off that EVA-VOS could achieve with an ideal oracle training scenario.

Mask-Only resembles the traditional way of annotating videos with segmentation masks [20, 22, 59, 77, 80]. Our method performs significantly better and achieves a 3.5× speed up compared to Mask-only at $\mathcal{J}\&\mathcal{F} = 0.85$ (Tab. 1).

Click-Only performs similarly to EVA-VOS at 50 hours but has a worse trade-off for either lower or higher budgets.

Tab. 1 compares EVA-VOS with the best frame and annotation selection approaches presented in Fig. 3. We quantify performance using the human annotation time in hours for each method to reach different $\mathcal{J}\&\mathcal{F}$ values and the average $\mathcal{J}\&\mathcal{F}$ up to 200 hours. Similar to Fig. 3, we observe that EVA-VOS overall outperforms all approaches, thus supporting our hypothesis that selecting frames and annotation type leads to both performance and time gains.

Qualitative results. In Fig. 5, we qualitatively compare EVA-VOS to Mask-only. Specifically, we illustrate the predicted masks of each method at different annotation budgets. We observe that EVA-VOS predicts more accurate masks faster than Mask-only.

5.4. EVA-VOS generalization ability

We now evaluate EVA-VOS in DAVIS 17 [58] without training any of our components on it. Similar to Sec. 5.3, we compare our method to Clicks-only and Masks-only. Fig. 6 illustrates the results, where we observe that EVA-VOS performs on par with Clicks-only and significantly outperforms Masks-only in lower annotation budgets.

6. Conclusions

We presented an alternative and efficient way to annotate objects in videos with segmentation masks. Our EVA-VOS framework shows significant gains in terms of annotation time (3.5× speed up) compared to the traditional, manual way of annotating objects in videos. Our experiments, especially on the challenging MOSE dataset, show that our framework reduces the total human annotation time while leading to high-quality segmentation masks for the videos.

Acknowledgements. D. Papadopoulos was supported by the DFF Sapere Aude Starting Grant "ACHILLES". V. Kalogeiton was supported by a Hi! PARIS grant and the ANR-22-CE23-0007. We would like to thank P. Pegios, J. Parslov, E. Riise, and Y. Benigmim for proofreading.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018. 2

[2] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, 2018. 2

[3] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *NeurIPS*, 2022. 1

[4] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 5

[5] Maksym Bekuzarov, Ariana Bermudez, Joon-Young Lee, and Hao Li. Xmem++: Production-level video segmentation from few annotated frames. In *ICCV*, 2023. 1, 3

[6] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019. 2, 5, 6

[7] 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. 1, 2

[8] 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. *arXiv:1803.00557*, 2018. 2, 5

[9] Arantxa Casanova, Pedro O. Pinheiro, Negar Rostamzadeh, and Christopher J. Pal. Reinforced active learning for image segmentation. In *ICLR*, 2020. 4

[10] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a Polygon-RNN. In *CVPR*, 2017. 2

[11] Bowen Chen, Huan Ling, Xiaohui Zeng, Jun Gao, Ziyue Xu, and Sanja Fidler. Scribblebox: Interactive annotation framework for video object segmentation. In *ECCV*, 2020. 2

[12] 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. 2

[13] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 2

[14] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 1, 2, 3

[15] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021. 1, 3, 5, 6

[16] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021. 1, 2, 6

[17] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 2

[18] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 2

[19] Kenan Dai, Jie Zhao, Lijun Wang, Dong Wang, Jianhua Li, Huchuan Lu, Xuesheng Qian, and Xiaoyun Yang. Video annotation for visual tracking via selection and refinement. In *ICCV*, 2021. 2

[20] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *NeurIPS*, 2022. 1, 2, 8

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[22] 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. In *ICCV*, 2023. 1, 2, 6, 7, 8

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 6

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

[25] Pinxue Guo, Tony Huang, Peiyang He, Xuefeng Liu, Tianjun Xiao, Zhaoyu Chen, and Wenqiang Zhang. Openvis: Open-vocabulary video instance segmentation. *arXiv preprint arXiv:2305.16835*, 2023. 2

[26] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 5

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4, 5, 6

[28] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Guided interactive video object segmentation using reliability-based attention maps. In *CVPR*, 2021. 2, 3

[29] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 1, 2

[30] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. In *CVPR*, 2017. 1, 2

[31] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *CVPR*, 2017. 2

[32] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. 2021. 1

[33] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 5

[34] Anna Khoreva, Anna Rohrbach, and Brent Schiele. Video object segmentation with referring expressions. In *ECCV Workshops*, pages 0–0, 2018. 2

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2017. 6

[36] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 5

[37] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 3, 4, 5, 6

[38] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In *NeurIPS*, 2017. 4

[39] Yuheng Li, Mingzhe Hu, and Xiaofeng Yang. Polyp-sam: Transfer sam for polyp segmentation. *arXiv preprint arXiv:2305.00293*, 2023. 2

[40] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018. 2

[41] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 5

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2, 5

[43] 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 *CVPR*, 2022. 1, 2

[44] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019. 2

[45] Jun Ma and Bo Wang. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*, 2023. 2

[46] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *PAMI*, 2018. 2

[47] Tim Meinhardt and Laura Leal-Taixé. Make one-shot video object segmentation efficient again. *NeurIPS*, 2020. 2

[48] Shentong Mo and Yapeng Tian. Av-sam: Segment anything model meets audio-visual localization and segmentation. *arXiv preprint arXiv:2305.01836*, 2023. 2

[49] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 1, 2

[50] OpenAI. Gpt-4 technical report, 2023. 4

[51] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 6

[52] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 4

[53] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, 2017. 2

[54] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Training object class detectors with click supervision. In *CVPR*, 2017. 5

[55] Dim P Papadopoulos, Ethan Weber, and Antonio Torralba. Scaling up instance annotation via label propagation. In *ICCV*, 2021. 2

[56] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2

[57] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 1, 2, 5, 8

[58] 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. 1, 2, 6, 8

[59] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *IJCV*, 2022. 1, 2, 5, 8

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *PMLR*, 2021. 5

[61] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. *arXiv:2307.01197*, 2023. 2

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 6

[63] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *PMLR*, 2017. 4, 6

[64] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *EECV*, 2020. 2

[65] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017. 2

[66] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the" object" in video object segmentation. *arXiv preprint arXiv:2212.06200*, 2022. 1, 2

[67] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016. 6

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 6

[69] 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. 1, 2

[70] 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. 1, 2

[71] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *BMCV*, 2017. 2

[72] Stephane Vujasinovic, Sebastian Bullinger, Stefan Becker, Norbert Scherer-Negenborn, Michael Arens, and Rainer Stiefelhagen. Revisiting click-based interactive video object segmentation. In *ICIP*, 2022. 2

[73] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 2

[74] Teng Wang, Jinrui Zhang, Junjie Fei, Yixiao Ge, Hao Zheng, Yunlong Tang, Zhe Li, Mingqi Gao, Shanshan Zhao, Ying Shan, et al. Caption anything: Interactive image description with diverse multimodal controls. *arXiv preprint arXiv:2305.02677*, 2023. 2

[75] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *NeurIPS*, 2019. 1

[76] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 1

[77] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 1, 2, 8

[78] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, 2018. 2

[79] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, 2021. 2

[80] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 1, 2, 6, 8

[81] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023. 2, 3

[82] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 1, 2

[83] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *NeurIPS*, 2021. 1, 2

[84] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. *NeurIPS*, 2022. 1, 2

[85] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition, 2022. 1

[86] 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. In *CVPR*, 2021. 3, 6, 7

[87] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. 5