This ACCV 2024 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2024
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

Strike the Balance: On-the-Fly Uncertainty based User Interactions for Long-Term Video Object Segmentation

Stéphane Vujasinović[1], Stefan Becker[1], Sebastian Bullinger[1], Norbert Scherer-Negenborn[1], Michael Arens[1], and Rainer Stiefelhagen[2]

[1] Fraunhofer IOSB, Ettlingen, Germany
{name}.{surname}@iosb.fraunhofer.de
[2] Karlsruhe Institut of Technology, Germany
{name}.{surname}@kit.edu

Abstract. In this paper, we introduce a variant of video object segmentation (VOS) that bridges interactive and semi-automatic approaches, termed Lazy Video Object Segmentation (ziVOS). In contrast, to both tasks, which handle video object segmentation in an off-line manner (*i.e.*, pre-recorded sequences), we propose through ziVOS to target online recorded sequences. Here, we strive to strike a balance between performance and robustness for long-term scenarios by soliciting user feedback's on-the-fly during the segmentation process. Hence, we aim to maximize the tracking duration of an object of interest, while requiring minimal user corrections to maintain tracking over an extended period. We propose Lazy-XMem as a competitive baseline, that estimates the uncertainty of the tracking state to determine whether a user interaction is necessary to refine the model's prediction. We introduce complementary metrics alongside those already established in the field, to quantitatively assess the performance of our method and the user's workload. We evaluate our approach using the recently introduced LVOS dataset, which offers numerous long-term videos. Our code is available at https://github.com/Vujas-Eteph/LazyXMem.

Keywords: Video Object Segmentation · Interactive

1 Introduction

Video Object Segmentation (VOS) is a fundamental challenge involving various tasks, including semi-automatic Video Object Segmentation (sVOS) and interactive Video Object Segmentation (iVOS) [83]. In sVOS, given an initial segmentation mask for the first video frame, methods classify each pixel in the subsequent video frames as a part of the object of interest (*i.e.*, foreground) or the background. Here, a user only interacts at the start of the sequence by providing the corresponding annotation mask to indicate which object to segment in the video. In contrast, iVOS methods incorporate user interactions in a multi-round scheme, where the user interacts with the method before each round, to improve

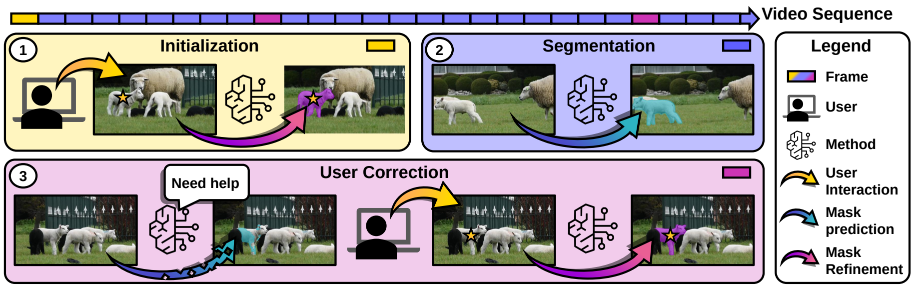


Fig. 1: Visual representation of the ziVOS framework. (1) The user initiates the segmentation by clicking to identify the object of interest in the video, (2) thus indicating which object to segment. Only when requested by the method (3) does the user provides corrective clicks on-the-fly.

the segmentation quality on the subsequent rounds. Both applications are suited for pre-recorded sequences, *i.e.*, offline segmentation, as sVOS methods assumes that the user has unlimited time to annotate the initial frame with utmost accuracy, whereas iVOS approaches expect the user to inspect the segmentation quality of the previous round, and interact for multiple rounds until the desired segmentation quality is achieved. However, while sVOS methods demonstrate impressive performances on short-term datasets [49, 69], their applicability to long-term sequences remains under-explored [2, 6, 8, 22, 36, 38, 59, 74] and yet to be addressed by iVOS methods.

This underscores a gap in methodologies suited for prolonged sequences, where maintaining error-free segmentation under challenging conditions becomes increasingly difficult. In this paper, we explore a hybrid framework, named lazy interactive Video Object Segmentation (ziVOS) (depicted in Fig. 1), that bridges the methodologies of sVOS and iVOS, focusing on maintaining robust object tracking with minimal user interactions. Unlike iVOS, we discard the round based scheme and integrate user corrections on-the-fly, refining the model's prediction as needed, while the method segments the video. Moreover, distinct from sVOS, in ziVOS the object of interest is indicated with a user interaction (*i.e.*, click). To achieve this, we only allow one interaction per frame and per object, and solely rely on click-based interactions, as pointing an object is the quickest, most intuitive and predictable interaction type for humans [14, 18]. Hence, we propose ziVOS to emulate a human-in-the-loop process when segmenting a video in an online fashion, that is better suited for dynamic applications, when user engagement is feasible and where maintaining consistent object tracking in challenging conditions is more critical than achieving segmentation accuracy. Concretely, our objective shifts from segmenting an object with high accuracy to maximizing the number of frames in which the object is segmented above a minimal alignment ratio (*i.e.*, Intersection over Union (IoU)), denoted as τ_{iou}, by integrating user corrections on-the-fly (only at critical events), while simultaneously reducing the user's workload.

We propose Lazy-XMem, as a baseline for future works addressing ziVOS. Lazy-XMem assess the uncertainty of a predicted object mask on-the-fly and refines it accordingly (through SAM-HQ [29]), if the uncertainty is too high, through either pseudo-corrections or user-corrections. In our approach, the Shannon entropy [53] serves as a proxy to estimate the performance of the tracking state – alignment (*i.e.*, IoU) ratio between the predicted mask and a hypothetical ground-truth. Similarly, recent studies in iVOS [15, 75] evaluate which frame to interact with at the end of a round. They compare the embeddings of each frame in the video sequence against all other frame embeddings to determine which frame to suggest to the user for new interactions, limiting this strategy to only pre-recorded videos. In contrast, our criterion is solely defined w.r.t. the tracker's state, how accurate the prediction is for the current observed frame, allowing us to work with non-prerecorded sequences. To our knowledge, only QDMN by Liu *et al.* [38] also estimates the tracking state in an online fashion, by predicting a quality score through a second head (following a similar design to [24]). However, unlike prior works [15, 38, 75], we estimate the tracker's state on pixel level in a *post-hoc* fashion, removing the need to train an auxiliary network. An additional benefit, to computing the uncertainty on pixel-level is that we can visually indicate ambiguous regions to the user where an interaction might be the most helpful. Moreover, depending on the confidence of the predicted mask, Lazy-XMem decides whether the current predicted mask will be stored in the memory. Hence, by selectively refining masks based on entropy-driven uncertainty estimation, we aim to maintain a balance between robustness and user-workload in ziVOS, specifically in long-term scenarios. Thus, while on-the-fly interactions may suggest constant user supervision, our approach minimizes this need by prompting the user (ideally) only at critical events – when Lazy-XMem is uncertain about its prediction. Our goal is to reduce the cognitive load by allowing the user to intervene only when necessary, enabling them to focus on other tasks simultaneously.

In this context the paper presents the following contributions: (1) Online (on-the-fly) assessment of the tracking state quality, by leverage entropy, to minimize the user's monitoring by providing interactions only at critical events (*e.g.*, occlusions, distractors). (2) A scheme to integrate pseudo-interactions, into our interactive feed-back loop to reduce the user's workload. We generate pseudo-interactions based on the original mask and the corresponding uncertainty. (3) Suitable metrics to evaluate the robustness of our method, and the user's workload w.r.t. the standard $\mathcal{J}\&\mathcal{F}$ metric proposed by Perazzi *et al.* [47]. (4) Evaluation on long-term [22] sequences to highlight the suitability of our method to maintain robust tracks.

2 Related Work

2.1 Semi-Automatic Video Object Segmentation

Early deep learning methods in sVOS follow an *online fine-tuning approache* [4, 39, 42, 58, 64, 65], which adapts the network's parameters on-the-fly while segmenting the objects of interest in the video sequence. This results, in slow inference times and poor generalization capabilities [83]. Concurrently, *propagation-based methods* [23, 26, 28, 48, 80] propagate the masks from the previous adjacent frame to the current one for segmentation, but they are prone to error accumulation and often fail during occlusions [83]. *Matching-based methods* [13, 57, 71–74, 77] leverage features from the initial and previous adjacent frame to segment the current frame. The leading methods in the field further integrate features from in-between frames (previously processed) into an external memory [9, 10, 37, 45, 46, 51, 52, 66, 72, 74], using cross-attention to link features from previous frames to the current frame to segment. However, these methods are limited in real-world applications due to their expanding memory requirements, making long-term segmentation on consumer-grade GPUs challenging.

Recent works address this bottleneck by selectively integrating frame representations into the external memory [33, 38, 59] or by generating compact representations to summarize similar features together [2, 6, 8, 34, 36]. These methods effectively manage the memory footprint, enabling more efficient sVOS on long-term videos. Newer methods [6] also explore improved ways to differentiate similar objects (distractors) from each other. Additionally, new datasets have recently been introduced [1, 16, 22] to provide an alternative to the classical DAVIS [49] and YouTube-VOS [69] datasets, with some targeted specifically for long-term video segmentation [22, 36] and tracking [32].

Contemporary works [11, 70, 84] leverage Segment Anything Model (SAM) [30] or a variant [29, 78, 82] to refine the original mask predicted by an sVOS baseline [8, 74]. However, in contrast to our framework, they refine every n-th mask predicted by the sVOS backbone with a SAM based approach [29, 30, 78, 82], and require continuous user monitoring to identify when interventions are needed. Furthermore, they diminish the influence to enhance the predictive accuracy for subsequent frame as they do not update the memory with the refined mask.

2.2 Interactive Image Segmentation

In interactive Image Object Segmentation (iIOS), methods predict a mask for an object of interest based on user interactions for a single image. These approaches aim to reduce the user's workload by replacing densely annotated mask for sparse annotations (*e.g.*, clicks [19, 27, 31, 35, 54, 55, 68], extreme points [17, 40, 81], or bounding boxes [50, 63, 67]). Most notable approach is f-BRS [54] which optimizes internal auxiliary features of the segmentation network to align its prediction's at the clicked position with the user annotated label. A follow up work by Sofiiuk *et al.* [55], replaces the previous f-BRS backbone with an HRNet [61] + OCR [76] network, to maintain high quality features through out the network to obtain a preciser segmentation mask. Since the introduction of SAM [30], a plethora of SAM-based methods have been proposed to solve the task in medical imaging [41] and natural images [79]. For instance, SAM-HQ [29] improves upon SAM by better handling complex shapes, such as thinner structures and objects with holes. Additionally, faster approaches like FastSAM [82] and MobileFast [78] have been developed to enhance performance and efficiency.

2.3 Interactive Video Object Segmentation

Originally intended to reduce the user's workload during video annotations [5], iVOS methods integrate user interactions in a round-based process. Most approaches follow the design introduced by Benard *et al.* [3], which combines sVOS and iIOS pipelines. The blueprint process for the iVOS task is as follow: (1) Firstly, the method predicts a segmentation mask for each frame in the video (through a sVOS baseline), based on an initial mask provided for a frame. (2) Next, a user scrolls through the resulting masks and selects a frame to interact with (*e.g.*, through clicks [25, 60, 62], scribbles [9, 20, 21, 43, 44]). Based on the provided interaction, a new mask is predicted for the annotated frame, serving as a new starting point when repeating step (1). Steps (1) and (2) are repeated one after the other, until the user is satisfied with the final results.

A persistent bottleneck is determining which frame to annotate for the next round. Recent approaches address this by identifying a quartet of candidate frames for the user to annotate [21], estimating which frame would yield the most improvement [75], or using a weakly supervised method to indicate the frame and type of interaction [15] to the user. To determine which frame or set of frames to annotate, these methods map each frame in the sequence into an embedding space, restricting them to short videos, as it requires storing the embedding of every frame. Here, each embedding encode the frame's representation and the quality of the corresponding predicted mask. The best candidate frame is selected by comparing each embedding w.r.t. others and against those of annotated frames, either through an agent [75] or by choosing the embedding that is furthest from any annotated embedding [15]. In contrast, our approach introduces corrections on-the-fly by directly assessing the tracking state during segmentation, thereby proposing an online methodology that is also not restricted to short sequences.

2.4 Uncertainty Estimation in Video Object Segmentation

Uncertainty estimations is essential to improve the reliability and explainability of a model, however estimating the uncertainty of Deep Neural Networks (DNNs), remains a challenging topic. To our knowledge, only the work by Liu *et al.* [38] incorporates a confidence score to asses the tracking state on-the-fly for the sVOS task by leveraging an auxiliary head (*i.e.*, QAM module), predicting a confidence score on how likely the predicted mask would align with a ground-truth annotation. Similar to our approach, QDMN manages its memory updates based on a threshold value that determines whether a predicted mask, given its confidence level, is reliable enough to be stored in the external memory. However, as the QAM module only predicts a single score per object, it is unable to guide the user during an interaction, as to where a correction might be the most valuable. In contrast, we explore uncertainty estimation through information theory [53] (*i.e.*, Shannon entropy) and update the memory with the refined mask.

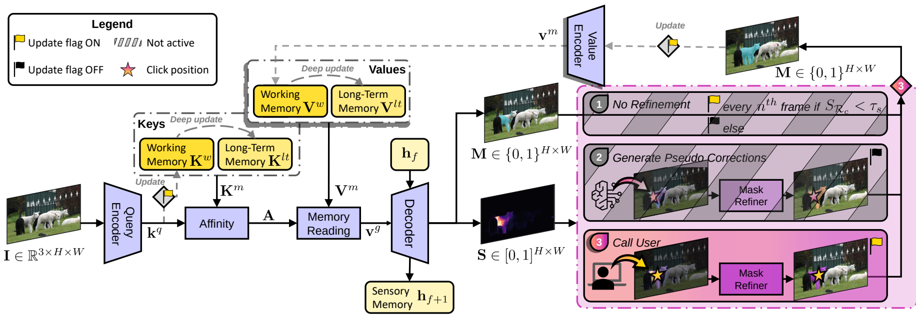


Fig. 2: Overview of Lazy-XMem for Lazy Video Object Segmentation. Our method is relies on an sVOS baseline (*i.e.*, XMem [8]). We leverage the entropy to estimate on-the-fly the tracking state. Based on the tracking state's, the method either uses the original mask of the sVOS baseline, or refine the original mask by generating pseudo-interactions, or requesting user interaction.

3 Method

We present Lazy-XMem, depicted in Fig. 2, as a baseline for future works targeting ziVOS. Lazy-XMem comprises the following key components: (1) An sVOS baseline, to predict object masks; (2) An uncertainty assessment component; (3) A mask refiner, to refine the original prediction from the sVOS baseline; (4) An interaction-issuer, to issue either pseudo- or user-corrections and (5) a memory update mechanism.

3.1 XMem as Baseline

We rely on XMem [8] as our sVOS baseline. Initialized with an object mask at the beginning, the network predicts masks for subsequent frames. For simplicity, we assume the network segments a single object. The key components are:

Convolutional Blocks: (1) A *query encoder*, that extracts query key features $\mathbf{k}^q \in \mathbb{R}^{C_k \times \frac{H}{16}\frac{W}{16}}$ from the current image to segment. (2) A *decoder*, which predicts an object mask $\mathbf{M} \in \{0,1\}^{H \times W}$ for a query frame $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$. (3) Lastly, a *value encoder*, that extracts value features $\mathbf{v}^m \in \mathbb{R}^{C_v \times \frac{H}{16}\frac{W}{16}}$ based on the current image $\mathbf{I}$ and the predicted mask $\mathbf{M}$.

Memories: Unlike previous works [9,10,45], XMem [8] employs three distinct memories: a *working memory*, a *long-term memory*, and a *sensory memory*.

(1) The *working memory*, is updated every n-th frame with query and value representations $\mathbf{K}^w \in \mathbb{R}^{C_k \times t\frac{H}{16}\frac{W}{16}}$ and $\mathbf{V}^w \in \mathbb{R}^{C_v \times t\frac{H}{16}\frac{W}{16}}$, until it reaches a capacity of $t = T_{max}$.

(2) When the working memory is full, it is distilled into l prototype features $\mathbf{k}^p \in \mathbb{R}^{C_k \times l}$ and $\mathbf{v}^p \in \mathbb{R}^{C_v \times l}$, based on usage frequency during memory reads. These prototypes are added to the long-term memory $\mathbf{K}^{lt} \in \mathbb{R}^{C_k \times L}$ and $\mathbf{V}^{lt} \in \mathbb{R}^{C_v \times L}$, with least-frequent-usage (LFU) filtering to remove obsolete features.

(3) The *sensory memory*, uses GRU cells [12] to update a hidden representation $\mathbf{h}_f \in \mathbb{R}^{C_h \times \frac{H}{16} \times \frac{W}{16}}$ every frame, encoding prior information like position [8].

Memory Reading: During the memory read operation, feature representations from both working and long-term memories are used, totaling $N = T\frac{H}{16}\frac{W}{16} + L$ elements. The model computes the similarity between memory keys and query keys using an anisotropic ℓ_2-similarity function [8], resulting in a similarity matrix $\mathbf{W}(\mathbf{K}^m, \mathbf{k}^q) \in \mathbb{R}^{N \times \frac{H}{16}\frac{W}{16}}$. Applying a softmax along the rows yields the affinity matrix $\mathbf{A}$. A new value $\mathbf{v}^g \in \mathbb{R}^{C_v \times \frac{H}{16}\frac{W}{16}}$ is then generated for the decoder through

$$\mathbf{v}^g = (\mathbf{V}^m)^\intercal \mathbf{A}(\mathbf{K}^m, \mathbf{k}^q), \tag{1}$$

where $\mathbf{K}^m = \mathbf{K}^w \oplus \mathbf{K}^{lt}$ and $\mathbf{V}^m = \mathbf{V}^w \oplus \mathbf{V}^{lt}$ are the concatenated working and long-term memories. The keys provide robust semantic information for matching, while the values encode boundary and texture cues needed for decoding [45]. For more detailed information, we refer readers to the original paper [8].

3.2 Uncertainty Estimation via Entropy

To estimate the uncertainty of the tracking state (*i.e.*, the predicted segmentation mask), we leverage the Shannon entropy [53], denoted as S. We consider pixels as discrete random variables whose classes c belong to a set $\mathcal{C}$, contains every object observed in a given video, including the background (*i.e.*, $c = 0$). We use the output values of the softmax layer as a approximation of a probability mass functions $p_{\mathcal{C}}(c \mid x_{h,w})$ for each pixel located x at (h, w). Here, $c \in \mathcal{C}$ denotes the class of the pixel, and (h, w) specifies the pixel's location in terms of height h and width w within a mask $\mathbf{M} \in \{0, \dots, |\mathcal{C}|\}^{H \times W}$.

However, as the number of classes $|\mathcal{C}|$ can vary over time (*i.e.*, from one video to another, or even from one frame to another in the same video), we normalize the entropy for consistency and comparability. Hence, we express the entropy of a pixel $x_{h,w}$ by

$$\mathbf{S}_{h,w} = -\frac{\sum_{c \in \mathcal{C}} p_{\mathcal{C}}(c \mid x_{h,w}) \log\left(p_{\mathcal{C}}(c \mid x_{h,w})\right)}{\log(|\mathcal{C}|)}, \tag{2}$$

where $\mathbf{S} \in [0, 1]^{H \times W}$ denotes the corresponding entropy map of the current frame to segment. To compute the entropy (*i.e.*, uncertainty) for a specific object class c, we use a dilated mask $\mathbf{M}_c^d$ based on the original object mask $\mathbf{M}_c$, such that the dilated mask allows us to exclude the background noise, while still considering the uncertainty around the predicted object's edges. We compute the dilated mask $\mathbf{M}_c^d$ through

$$\mathbf{M}_c^d(h, w) = \max_{(i,j) \in \mathbf{K}} \mathbf{M}_c(h + i, w + j), \tag{3}$$

where $\mathbf{K}$, a circular dilation kernel, determines the increase of $\mathbf{M}_c^d$ relative to $\mathbf{M}_c$ by a specified ratio. The variables i and j represent the coordinates within $\mathbf{K}$. The supplementary material contains an empirical evaluation of suitable values for this hyper-parameter. Otherwise, using directly the predicted mask $\mathbf{M}_c$ might truncate the aleatoric uncertainty (especially near the edges of the object). Hence, we compute the total uncertainty via the joint entropy $S_{\mathcal{R}_c}$ of a considered object region $\mathcal{R}_c = \{(h, w) \mid \mathbf{M}_c^d(h, w) = 1\}$, through

$$S_{\mathcal{R}_c} = \sum_{r \in \mathcal{R}_c} \mathbf{S}_r \left(x_r \mid x_{r-1}, \ldots, x_1 \right) \approx \sum_{r \in \mathcal{R}_c} \mathbf{S}_r, \tag{4}$$

by essentially summing the conditional entropies of each random variable within the considered region. This approach captures the inter-dependencies among all variables, reflecting their collective impact on $\mathcal{R}_c$. However, computing the joint entropy is impractical as the network does not provide any joint or conditional distributions for a formal evaluation. Additionally, the computational cost would grow exponentially with respect to the number of classes $\mathcal{C}$ and the size of the region $\mathcal{R}_c$ (*i.e.*, $O(|\mathcal{C}|^{|\mathcal{R}_c|})$ time complexity). To reduce the computational complexity, we assume zero mutual information between the predicted probability distributions of pixels in the region $\mathcal{R}_c$. This allows us to sum the entropy of each pixel $\mathbf{S}_{h,w}$ belonging to the region of interest $\mathbf{M}_c^d$ (refer to Eq. (4)), allowing us to significantly reduce the computational cost (*i.e.*, to $O(|\mathcal{C}| \times |\mathcal{R}_c|)$ complexity).

Additionally, considering that the object size may vary from one image to another, we divide $S_{\mathcal{R}_c}$ by the size of the corresponding region $|\mathcal{R}_c|$. This dampens the fluctuation of $S_{\mathcal{R}_c}$ due to object size variations.

3.3 Mask Refinement

For the mask-refinement component, we rely on SAM-HQ [29], which extends SAM [30] to segment intricate object structures in more details, while preserving its zero-shot capabilities and flexibility. SAM-HQ [29] introduces two additional components on top of SAM [30]: (1) An *HQ-output token* to correct the original SAM's mask. (2) A *global-local features fusion*, which fuses early features with later ones (*i.e.*, after the first and last global attention block respectively) to enrich the features used by the mask decoder. For more details about SAM [30] and SAM-HQ [29] we refer the reader to the original sources.

3.4 Issuing Corrections

For a given object c, we record the corresponding masked entropy $S_{\mathcal{R}_c}$ at each frame, such that $\mathbf{s}_{\mathcal{R}_c} = [S_{\mathcal{R}_c}(f=0), \ldots, S_{\mathcal{R}_c}(f=F)]$, where f denotes the frame index and F the latest frame to segment. Let $\mathbf{s}'_{\mathcal{R}_c}$ denote the derivative of $\mathbf{s}_{\mathcal{R}_c}$, such that $\mathbf{s}'_{\mathcal{R}_c} = [\Delta S_{\mathcal{R}_c}(f=1), \ldots, \Delta S_{\mathcal{R}_c}(f=F)]$, where $\Delta S_{\mathcal{R}_c}(f) = S_{\mathcal{R}_c}(f) - S_{\mathcal{R}_c}(f-1)$. Depending on $\Delta S_{\mathcal{R}_c}(F)$ we either generate a pseudo- or request a user-correction.

Following the definition in iIOS, we denote a positive or negative click as a point-wise interaction to indicate a falsely classified region as either foreground or background. **User-Correction (U-C)**: We prompt a user correction whenever $\Delta S_{\mathcal{R}_c}(f) \geq \tau_u$, where τ_u denotes the threshold above which a user correction is requested. The user indicates a foreground or background region via a positive or negative click, which is then processed by the mask refiner to generate a new mask. The original mask is not used during refinement due to its high uncertainty. Note, that during a user-interactions, Lazy-XMem does not preselect candidates regions. Instead, our method overlays the entropy on the image to visually guide the user's interaction towards the most uncertain regions in the model's predictions. We illustrate this in the qualitative experiments of the supplementary material. **Pseudo-Correction (P-C)**: In addition to requesting on-the-fly user corrections, the model generates pseudo-corrections when $\tau_u > \Delta S_{\mathcal{R}_c}(f) \geq \tau_p$, with τ_p representing the lower bound for a pseudo-correction to be generated. A pseudo-correction p_f^c for object c, given frame f, is by

$$\mathbf{E}_c(h,w) = \min_{(h_r,w_r)\in\Omega} \sqrt{(h-h_r)^2 + (w-w_r)^2}, \tag{5}$$

$$p_f^c = \underset{(h,w)}{\operatorname{argmax}} \left(\mathbf{M}_c^d \odot \mathbf{E}_c \odot (\mathbf{1}_{H\times W} - \mathbf{S})\right), \tag{6}$$

where Ω denotes the set of pixel that belong to the boundaries of the object mask $\mathbf{M}_c$, $\mathbf{E}_c$ a distance field, and $\odot$ represents the Hadamard product.

3.5 Interaction and Uncertainty Driven Memory Updates

At each user-correction, we update the working memory of our sVOS baseline with the newly refined mask. This update strategy, termed *Interaction-Driven Update* (IDU), improves the method's robustness as the refined mask can influence the segmentation of the subsequent frames. An additional update mechanism, named *Uncertainty-Driven Update* (UDU), prevents updating the working memory with the original representation when the corresponding uncertainty $S_{\mathcal{R}_c}$, exceeds τ_m (similarly to QDMN [38]).

4 Metrics

Since we introduce ziVOS, we propose complementary metrics to the standard $\mathcal{J}\&\mathcal{F}$ presented by Perazzi *et al.* [47] to quantify the user's workload in providing on-the-fly corrections and to evaluate the robustness of a given method.

4.1 Robustness Metric

We take inspiration from Kristan *et al.* [32] and propose R@τ_{IoU} (higher is better) to measure the robustness of a method. Given a threshold value τ_{IoU}, we compute the ratio of frames, in which the predicted object mask attains an IoU above or equal to τ_{IoU} for all objects in a given dataset. More formally, let $\mathcal{O}$ be the set of objects in the dataset, where $\mathcal{F}_o$ is the set of frames in which object o is present, we define $R@\tau_{\text{IoU}}$, such that

$$R@\tau_{\text{IoU}} = \frac{1}{|\mathcal{O}|}\sum_{o\in\mathcal{O}} \frac{1}{|\mathcal{F}_o|}\sum_{f\in\mathcal{F}_o} \mathbb{1}_{[\text{IoU}(\mathbf{M}_f^o,\mathbf{GT}_f^o)\geq\tau_{IoU}]}, \tag{7}$$

where $\mathbf{M}_f^o$ and $\mathbf{GT}_f^o$ denote respectively the predicted mask and ground-truth annotation for object o at frame f. Like in [32], whenever the method correctly predicts the absence of an object we set $\mathbb{1}_{\mathrm{IoU}(\mathbf{M}_f^o, \mathbf{GT}_f^o) \geq \tau_{IoU}}$ to 1, otherwise to 0.

4.2 User-Workload Metrics

To quantitatively evaluate the workload for the user we introduce the following metrics: (1) Number of Correction (NoC) to denote the total number of user-corrections issued by the model to refine its current prediction. (2) Interaction Density Index (IDI) (higher is better), is introduced as an intuitive metric that reports the average time between two user-corrections reported in seconds. Note that some sequences might have no user interactions; however, to include every sequence in the evaluation, we consider the initialization and the end of a sequence as user-interactions. (3) As IDI does not reflect the underlying distribution of the interactions, we provide through Average Correction Interval (ACI) (which encapsulates both NoC and IDI) a score to indicate this distribution. In essence, we compute the cumulative count over user interactions and their respective distance to each other. Consequently, a low ACI score indicates more spread out user interactions, while a higher score indicates consecutive interactions more closer to each other within a short period. More formally, let $\mathcal{N}_o = \{f_{p=0}, \cdots, f_{p=P_o} \mid f_p \in \mathcal{F}_o\}$ denote the set containing the frame indexes f_p where a user prompt p is issued for object o. Hence, we define ACI, such that

$$\mathrm{ACI} = \sum_{o \in \mathcal{O}} \frac{1}{|\mathcal{F}_o|} \sum_{i=1}^{|\mathcal{F}_o|} \sum_{j=1}^{i} n_j, \tag{8}$$

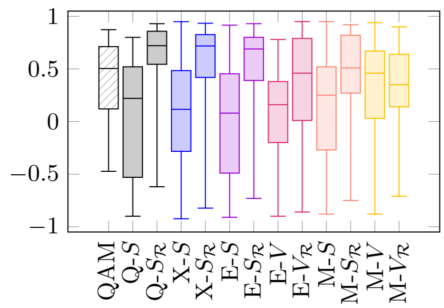
where $n_j = \sum_{f_p=1}^{\mathcal{N}_o} \mathbb{1}_{[j = f_p - f_{p-1}]}$ denotes the number of occurrences a user provided corrections at a distance of j frames from one prompt to the next.

5 Experiments

In Sec. 5.1, we assess the effectiveness of the proposed masked entropy $S_{\mathcal{R}_c}$ to estimate the tracking's state on-the-fly. We present the evaluation protocol for the ziVOS benchmark in Sec. 5.2, and present our results on the LVOS dataset [22] in Sec. 5.3. Lastly, an ablation study in Sec. 5.4 examines the impact of each design choice. We provide qualitative results in the supplementary material.

5.1 Entropy as a Proxy

To evaluate the effectiveness of the masked entropy $\mathcal{S}_{\mathcal{R}_c}$ to estimate the tracking's state, we compare against the following approaches: (1) Using the Quality-Aware Module (QAM) from QDMN [38], which predicts a confidence score through an auxiliary network. (2) Computing the entropy S and its masked version $S_{\mathcal{R}}$ for various models: single models denoted as Q and X respectively



(a) On DAVIS 2017 [49]

(b) On LVOS [22]

Fig. 3: Comparison of correlation coefficients across the DAVIS 2017 [49] and LVOS [22] datasets: i) the QAM module [38] and entropy based QDMN [38] (as Q-S and Q-$S_\mathcal{R}$), ii) entropy results for a single baseline (X-S, X-$S_\mathcal{R}$), an ensemble (E-S, E-$S_\mathcal{R}$), and Monte-Carlo methods (M-S, M-$S_\mathcal{R}$), and iii) epistemic uncertainty variants for ensemble and Monte-Carlo (E-V, E-$V_\mathcal{R}$, M-V, M-$V_\mathcal{R}$).

for the QDMN [38] and XMem [8] networks, an ensemble model denoted as E, and a Monte-Carlo dropout model denoted as M. We provide in the supplementary material details to the ensemble and Monte Carlo dropout approaches. (3) We also consider for the ensemble and Monte-Carlo dropout variants the epistemic uncertainty, denoted as V (and the masked version $V_\mathcal{R}$). For each method, we compute the Spearman coefficient [56] to measure the correlation between each variant's output for the tracking state w.r.t. the actual IoU. We conduct our evaluations on the DAVIS 2017 [49] and LVOS [22] validation sets, featuring short and long videos respectively.

Fig. 3 presents the distribution (*i.e.*, box-plots) of the correlation coefficients when computing the coefficient for every object present in a dataset. Aside from the QAM based methods, we expect an inverse correlation, however, to facilitate the comparison, we invert the correlation results for all methods except for the QAM version. Consequently, values closer to 1 indicate a higher correlation, suggesting a more accurate estimate of the tracking state. Across both the DAVIS 2017 [49] and LVOS [22] datasets, variants employing masked entropy (*i.e.*, $S_\mathcal{R}$) demonstrate notably stronger correlations. This highlights the effectiveness of isolating uncertainty at the object level using a mask. Among the different model variants – single (Q and X), ensemble (E), and Dropout (M) – the single models (Q and X) leveraging $S_\mathcal{R}$ outperform even the advanced learning-based QAM module [38].

Hence, by examining Fig. 3, the most effective method for estimating the tracking state on-the-fly appears to be the masked entropy approach, particularly the X-$S_\mathcal{R}$ variant, as its median value is closer to 1 and the distribution is notably narrower. This underscores the efficacy of masked entropy as a straightforward yet robust approach to estimate the tracking's state on-the-fly.

5.2 Evaluation Framework for ziVOS

As our goal is to improve the robustness of video object segmentation methods by incorporating user corrections on-the-fly, while mimizing the user's workload, we only allow one interaction per object per frame. Following standard practices only two form of interactions are possible, *i.e.*, positive and negative, to indicate foreground and background regions respectively. Moreover, we limit the type of interactions to only clicks, as pointing an object is the quickest and most intuitive interaction type for humans [14, 18].

To automatically evaluate ziVOS methods, we simulate a user interaction u_f^o at frame f for object o in the sequence using a *simulated agent*, whenever the ziVOS method requests a user correction. We simulate a user interaction u_f^o at the center of gravity of the largest misclassified region or from the ground-truth object mask (both approaches shown similar results, but for the remaining we rely on the latter approach for simplicity).

5.3 Quantitative Results (ziVOS)

In Tab. 1, we present quantitative results of Lazy-XMem compared to State-of-the-Art (SOTA) methods on the LVOS validation set [22] by following the ziVOS evaluation process outlined in Sec. 5.2. Hence, unlike sVOS, which relies on curated masks, we rely on click to indicate which object to track in the video. As a results, we employ imperfect masks generated by the mask refiner (*i.e.*, SAM-HQ [29]), which more closely resembles real-world scenarios. We provide additional results on LVOS [22] when following the protocol in sVOS in our supplementary material.

To allow for a better comparability, we also evaluate a modified version of QDMN [38], that adopts the same design as LazyXMem for integrating user and pseudo-corrections, with the notable exception that the tracking state estimation is based on the QAM module [38]. Moreover, we evaluate an alternative approach that simply requests user corrections at random intervals throughout the sequence (denoted as Rand-Lazy-XMem). In addition, we introduce a variant of Lazy-XMem (denoted by Lazy-XMem†), which operates without user corrections to facilitate the comparison w.r.t. SOTA sVOS methods. We report the popular $\mathcal{J}\&\mathcal{F}$ metric, alongside our complementary metrics (see Sec. 4).

As shown in Tab. 1, our proposed Lazy-XMem† achieves competitive results w.r.t. to the SOTA sVOS methods. However the robustness is still close to the original XMem [8] version, despite of the increase in accuracy. By incorporating user corrections (*i.e.*, Lazy-XMem), we manage to improve the robustness by 13 points on average over all robustness metrics, while requesting in total 325 interactions from the user for the entire datasets, averaging one interaction every 18.4 seconds. This corresponds to approximately 1.05% of the total number of frames in the LVOS validation set, which contains $30,876$ frames [22].

While Lazy-XMem incorporates user corrections on-the-fly to enhance its robustness, it requires a continuous participation of the user throughout the segmentation process. Therefore, we present Lazy-XMem as an alternative to sVOS

Table 1: Quantitative evaluation of ziVOS and sVOS methods on the LVOS validation set [22] following the ziVOS framework. Here, we initialize each methods with an imperfect mask, in contrast to sVOS, to indicate which object to segment in the sequence.

		Robustness			**User-Workload**		
Method	$\mathcal{J}\&\mathcal{F}$	R@0.1	R@0.25	R@0.5	ACI	NoC	IDI
sVOS Methods							
QDMN [38] (ECCV 2022)	44.2	47.8	45.5	36.2	-	-	-
XMem [8] (ECCV 2022)	52.8	57.0	55.0	49.0	-	-	-
DEVA [7] (ICCV 2023)	55.1	**63.6**	**59.3**	52.4	-	-	-
Cutie-base [6] (CVPR 2024)	57.0	59.2	57.8	52.4	-	-	-
Cutie-small [6] (CVPR 2024)	**57.6**	58.6	57.0	**52.5**	-	-	-
Lazy-XMem† (ours)	56.4	58.8	56.8	50.6	-	-	-
ziVOS Methods							
Rand-Lazy-XMem	61.3	67.9	65.8	59.3	5.17	335	17.9
Lazy-QDMN	52.7	58.2	52.0	42.9	5.64	360	16.7
Lazy-XMem (ours)	**64.3**	**70.2**	**67.8**	**62.3**	5.02	**325**	18.4

and iVOS methods to segment offline and online videos, in scenarios where user engagement is feasible and where segmenting over an extended period with high reliability is the priority.

5.4 Ablations

To provide more insights into our pipeline, we detail the influence of each design choice in Tab. 2. Using the Uncertainty Driven Update (UDU), we achieve improvements over the baseline by selectively integrating memory predictions that present sufficiently low uncertainty. By soliciting user interactions to refine the initial mask predicted by the sVOS baseline (i.e., XMem [8]), we achieve slight improvements at the cost of 507 interactions across the dataset. While, storing the refined masks as references for future segmentation after a user correction through the Interaction Driven Update (IDU), we attain substantial improvements in both robustness and user workload. However, using the original mask from XMem [8], associated with a high uncertainty, as an additional prompt to the user's interaction for the mask refiner leads to a decrease in performance.

By generating pseudo-interactions following the strategy outlined in Sec. 3.4 to refine XMem's initial mask, we enhance the robustness even further while slightly reducing the user's workload. However, saving the resulting refined mask from a pseudo-interaction (pseudo-IDU), affect only marginally the robustness, but increases the user workload considerably. When discarding the user interactions and only relying on pseudo-corrections, *i.e.*, Lazy-XMem†, we obtain a similar setup to sVOS methods and manage to improve the results of the XMem [8] baseline, even attain competitive results against the current SOTA sVOS methods as shown in Tab. 1. Thus, our extension improves the baseline by: (1) dis-

Table 2: Ablation study for Lazy-XMem on the ziVOS framework. We initialize each method with an imperfect mask, to indicate which object to segment in the sequence.

Configuration						**Robustness**			**User-Workload**		
	Pseudo		*User*								
UDU	Corr.	IDU	Corr.	IDU	$\mathcal{J}\&\mathcal{F}$	R@0.1	R@0.25	R@0.5	ACI	NoC	IDI
-	-	-	-	-	52.8	57.0	55.0	49.0	-	-	-
✓	-	-	-	-	54.7	56.3	54.5	50.0	-	-	-
✓	✓	-	-	-	56.4	58.8	56.8	50.6	-	-	-
✓	✓	✓	-	-	53.1	57.0	55.1	49.6	-	-	-
✓	-	-	✓	-	55.6	58.2	56.4	51.8	7.80	507	12.6
✓	-	-	✓	✓	62.9	67.8	66.2	60.9	5.05	327	18.3
✓	✓	-	✓	✓	**64.3**	**70.2**	67.8	**62.3**	5.02	**325**	18.4
✓	✓	✓	✓	✓	**64.3**	70.1	**68.2**	62.1	5.91	352	17.3

carding non-confident predictions from being added to the memory; (2) issuing pseudo-corrections to prompt SAM-HQ [29], thereby refining the baseline's initial prediction when the method's uncertainty increases sharply (Sec. 3.4); (3) and requests user-corrections on-the-fly as needed to improve the robustness.

6 Conclusion

We introduce Lazy-XMem as a reference for future work tackling ziVOS, a hybrid combination of sVOS and iVOS, that emulates a human-in-the-loop process for online video segmentation. Through Lazy-XMem, we enhance the robustness (i.e., the ratio of frames segmented above a certain IoU threshold) w.r.t. the sVOS baseline (*i.e.*, XMem [8]) by integrating pseudo and user corrections on-the-fly. However, as we solicit user corrections, we also aim to reduce the user's workload, striking a balance between performance and user engagement by requesting help only during critical events, where the method is likely to fail. We estimate the tracking state (*i.e.*, confidence) of the method by leveraging entropy (from information theory) and demonstrate that our proposed approach is an effective means of estimating the tracking state on-the-fly. As we propose ziVOS, we also introduce complementary metrics to the popular $\mathcal{J}\&\mathcal{F}$ metrics [47], to evaluate the robustness of our approach and the user's workload. Our evaluation on the long-term LVOS dataset [22] shows that Lazy-XMem improves the robustness relative to the baseline, albeit at the cost of additional user interactions. Thus, we present Lazy-XMem as an alternative to sVOS and iVOS methods to segment online video, particularly when on-the-fly corrections by a user are possible and when maintaining the tracking for an extended period is preferred over the accuracy of a method.

References

1. Athar, A., Luiten, J., Voigtlaender, P., Khurana, T., Dave, A., Leibe, B., Ramanan, D.: Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In: Winter Conference on Applications of Computer Vision (WACV) (2023) 4
2. Bekuzarov, M., Bermudez, A., Lee, J.Y., Li, H.: Xmem++: Production-level video segmentation from few annotated frames (2023) 2, 4
3. Benard, A., Gygli, M.: Interactive video object segmentation in the wild. arXiv preprint arXiv:1801.00269 (2017) 5
4. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3
5. Caelles, S., Montes, A., Maninis, K.K., Chen, Y., Van Gool, L., Perazzi, F., Pont-Tuset, J.: The 2018 davis challenge on video object segmentation. arXiv:1803.00557 (2018) 5
6. Cheng, H.K., Oh, S.W., Price, B., Lee, J.Y., Schwing, A.: Putting the object back into video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 2, 4, 13
7. Cheng, H.K., Oh, S.W., Price, B., Schwing, A., Lee, J.Y.: Tracking anything with decoupled video segmentation. In: International Conference on Computer Vision (ICCV) (2023) 13
8. Cheng, H.K., Schwing, A.G.: XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In: European Conference on Computer Vision (ECCV) (2022) 2, 4, 6, 7, 11, 12, 13, 14
9. Cheng, H.K., Tai, Y.W., Tang, C.K.: Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4, 5, 6
10. Cheng, H.K., Tai, Y.W., Tang, C.K.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In: Neural Information Processing Systems (NeurIPS) (2021) 4, 6
11. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv preprint arXiv:2305.06558 (2023) 4
12. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014) 7
13. Cho, S., Lee, H., Lee, M., Park, C., Jang, S., Kim, M., Lee, S.: Tackling background distraction in video object segmentation. In: European Conference on Computer Vision (ECCV) (2022) 4
14. Clark, H.H.: Coordinating with each other in a material world. Discourse studies (2005) 2, 12
15. Delatolas, T., Kalogeiton, V., Papadopoulos, D.P.: Learning the what and how of annotation in video object segmentation. In: Winter Conference on Applications of Computer Vision (WACV) (2024) 3, 5
16. Ding, H., Liu, C., He, S., Jiang, X., Torr, P.H., Bai, S.: MOSE: A new dataset for video object segmentation in complex scenes. In: International Conference on Computer Vision (ICCV) (2023) 4
17. Dupont, C., Ouakrim, Y., Pham, Q.C.: Ucp-net: Unstructured contour points for instance segmentation. arXiv preprint arXiv:2109.07592 (2021) 4

18. Firestone, C., Scholl, B.J.: "please tap the shape, anywhere you like" shape skeletons in human vision revealed by an exceedingly simple measure. Psychological science (2014) 2, 12
19. Forte, M., Price, B., Cohen, S., Xu, N., Pitié, F.: Getting to 99% accuracy in interactive segmentation. arXiv preprint arXiv:2003.07932 (2020) 4
20. Heo, Y., Jun Koh, Y., Kim, C.S.: Interactive video object segmentation using global and local transfer modules. In: European Conference on Computer Vision (ECCV) (2020) 5
21. Heo, Y., Koh, Y.J., Kim, C.S.: Guided interactive video object segmentation using reliability-based attention maps. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 5
22. Hong, L., Chen, W., Liu, Z., Zhang, W., Guo, P., Chen, Z., Zhang, W.: Lvos: A benchmark for long-term video object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) 2, 3, 4, 10, 11, 12, 13, 14
23. Hu, P., Wang, G., Kong, X., Kuen, J., Tan, Y.P.: Motion-guided cascaded refinement network for video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
24. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask Scoring R-CNN. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3
25. Jain, S., Grauman, K.: Click carving: Segmenting objects in video with point clicks. In: Association for the Advancement of Artificial Intelligence (AAAI) (2016) 5
26. Jang, W.D., Kim, C.S.: Online video object segmentation via convolutional trident network. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4
27. Jang, W.D., Kim, C.S.: Interactive image segmentation via backpropagating refinement scheme. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4
28. Johnander, J., Danelljan, M., Brissman, E., Khan, F.S., Felsberg, M.: A generative appearance model for end-to-end video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4
29. Ke, L., Ye, M., Danelljan, M., Liu, Y., Tai, Y.W., Tang, C.K., Yu, F.: Segment anything in high quality. In: Neural Information Processing Systems (NeurIPS) (2023) 3, 4, 8, 12, 14
30. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023) 4, 8
31. Kontogianni, T., Gygli, M., Uijlings, J., Ferrari, V.: Continuous adaptation for interactive object segmentation by learning from corrections. In: European Conference on Computer Vision (ECCV) (2020) 4
32. Kristan, M., et al.: The first visual object tracking segmentation vots2023 challenge results. In: International Conference on Computer Vision Workshops (ICCVW) (2023) 4, 9, 10
33. Li, M., Hu, L., Xiong, Z., Zhang, B., Pan, P., Liu, D.: Recurrent dynamic embedding for video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4
34. Li, Y., Shen, Z., Shan, Y.: Fast video object segmentation using the global context module. In: European Conference on Computer Vision (ECCV) (2020) 4
35. Li, Z., Chen, Q., Koltun, V.: Interactive image segmentation with latent diversity. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4

36. Liang, Y., Li, X., Jafari, N., Chen, J.: Video object segmentation with adaptive feature bank and uncertain-region refinement. In: Neural Information Processing Systems (NeurIPS) (2020) 2, 4
37. Liu, Y., Yu, R., Wang, J., Zhao, X., Wang, Y., Tang, Y., Yang, Y.: Global spectral filter memory network for video object segmentation. In: European Conference on Computer Vision (ECCV) (2022) 4
38. Liu, Y., Yu, R., Yin, F., Zhao, X., Zhao, W., Xia, W., Yang, Y.: Learning quality-aware dynamic memory for video object segmentation. In: European Conference on Computer Vision (ECCV) (2022) 2, 3, 4, 5, 9, 10, 11, 12, 13
39. Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: Video object segmentation without temporal information. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2018) 3
40. Maninis, K.K., Caelles, S., Pont-Tuset, J., Van Gool, L.: Deep extreme cut: From extreme points to object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
41. Marinov, Z., Jäger, P.F., Egger, J., Kleesiek, J., Stiefelhagen, R.: Deep interactive segmentation of medical images: A systematic review and taxonomy. arXiv preprint arXiv:2311.13964 (2023) 4
42. Meinhardt, T., Leal-Taixé, L.: Make one-shot video object segmentation efficient again. Neural Information Processing Systems (NeurIPS) (2020) 3
43. Miao, J., Wei, Y., Yang, Y.: Memory aggregation networks for efficient interactive video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 5
44. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Fast user-guided video object segmentation by interaction-and-propagation networks. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 5
45. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: International Conference on Computer Vision (ICCV) (2019) 4, 6, 7
46. Park, K., Woo, S., Oh, S.W., Kweon, I.S., Lee, J.Y.: Per-clip video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4
47. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 3, 9, 14
48. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4
49. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv:1704.00675 (2017) 2, 4, 11
50. Rother, C., Kolmogorov, V., Blake, A.: " grabcut" interactive foreground extraction using iterated graph cuts. ACM transactions on graphics (TOG) (2004) 4
51. Seong, H., Hyun, J., Kim, E.: Kernelized memory network for video object segmentation. In: European Conference on Computer Vision (ECCV) (2020) 4
52. Seong, H., Oh, S.W., Lee, J.Y., Lee, S., Lee, S., Kim, E.: Hierarchical memory matching network for video object segmentation. In: International Conference on Computer Vision (ICCV) (2021) 4
53. Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal (1948) 3, 5, 7

54. Sofiiuk, K., Petrov, I., Barinova, O., Konushin, A.: f-brs: Rethinking backpropagating refinement for interactive segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
55. Sofiiuk, K., Petrov, I.A., Konushin, A.: Reviving iterative training with mask guidance for interactive segmentation. In: International Conference on Image Processing (ICIP) (2022) 4
56. Spearman, C.: The proof and measurement of association between two things. American Journal of Psychology (1904) 11
57. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: Feelvos: Fast end-to-end embedding learning for video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4
58. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. British Machine Vision Conference (BMVC) (2017) 3
59. Vujasinovic, S., Bullinger, S., Becker, S., Scherer-Negenborn, N., Arens, M., Stiefelhagen, R.: Readmem: Robust embedding association for a diverse memory in unconstrained video object segmentation. In: British Machine Vision Conference (BMVC) (2023) 2, 4
60. Vujasinović, S., Bullinger, S., Becker, S., Scherer-Negenborn, N., Arens, M., Stiefelhagen, R.: Revisiting click-based interactive video object segmentation. In: International Conference on Image Processing (ICIP) (2022) 5
61. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020) 4
62. Wang, T., Han, B., Collomosse, J.P.: Touchcut: Fast image and video segmentation using single-touch interaction. Computer Vision and Image Understanding (CVIU) (2014) 5
63. Wu, J., Zhao, Y., Zhu, J.Y., Luo, S., Tu, Z.: Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2014) 4
64. Xiao, H., Feng, J., Lin, G., Liu, Y., Zhang, M.: Monet: Deep motion exploitation for video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 3
65. Xiao, H., Kang, B., Liu, Y., Zhang, M., Feng, J.: Online meta adaptation for fast video object segmentation. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2019) 3
66. Xie, H., Yao, H., Zhou, S., Zhang, S., Sun, W.: Efficient regional memory network for video object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4
67. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.: Deep grabcut for object selection. arXiv preprint arXiv:1707.00243 (2017) 4
68. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.S.: Deep interactive object selection. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 4
69. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018) 2, 4
70. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos (2023) 4

71. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by foreground-background integration. In: European Conference on Computer Vision (ECCV) (2020) 4
72. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. In: Neural Information Processing Systems (NeurIPS) (2021) 4
73. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by multi-scale foreground-background integration. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2021) 4
74. Yang, Z., Yang, Y.: Decoupling features in hierarchical propagation for video object segmentation. In: Neural Information Processing Systems (NeurIPS) (2022) 2, 4
75. Yin, Z., Zheng, J., Luo, W., Qian, S., Zhang, H., Gao, S.: Learning to recommend frame for interactive video object segmentation in the wild. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3, 5
76. Yuan, Y., Chen, X., Chen, X., Wang, J.: Segmentation transformer: Object-contextual representations for semantic segmentation. In: European Conference on Computer Vision (ECCV) (2021) 4
77. Yunyao, M., Ning, W., Wengang, Z., Houqiang, L.: Joint inductive and transductive learning for video object segmentation. In: International Conference on Computer Vision (ICCV) (2021) 4
78. Zhang, C., Han, D., Qiao, Y., Kim, J.U., Bae, S.H., Lee, S., Hong, C.S.: Faster segment anything: Towards lightweight sam for mobile applications. arXiv preprint arXiv:2306.14289 (2023) 4
79. Zhang, C., Liu, L., Cui, Y., Huang, G., Lin, W., Yang, Y., Hu, Y.: A comprehensive survey on segment anything model for vision and beyond. arXiv:2305.08196 (2023) 4
80. Zhang, L., Lin, Z., Zhang, J., Lu, H., He, Y.: Fast video object segmentation via dynamic targeting network. In: International Conference on Computer Vision (ICCV) (2019) 4
81. Zhang, S., Liew, J.H., Wei, Y., Wei, S., Zhao, Y.: Interactive object segmentation with inside-outside guidance. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
82. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., Wang, J.: Fast segment anything (2023) 4
83. Zhou, T., Porikli, F., Crandall, D.J., Van Gool, L., Wang, W.: A survey on deep learning technique for video segmentation. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2023) 1, 4
84. Zhu, J., Chen, Z., Hao, Z., Chang, S., Zhang, L., Wang, D., Lu, H., Luo, B., He, J.Y., Lan, J.P., Chen, H., Li, C.: Tracking anything in high quality (2023) 4