

Segment Anything Meets Point Tracking

Frano Rajić^{1,3} Lei Ke^{1,2} Yu-Wing Tai² Chi-Keung Tang² Martin Danelljan¹ Fisher Yu¹
¹ETH Zürich ²HKUST ³EPFL

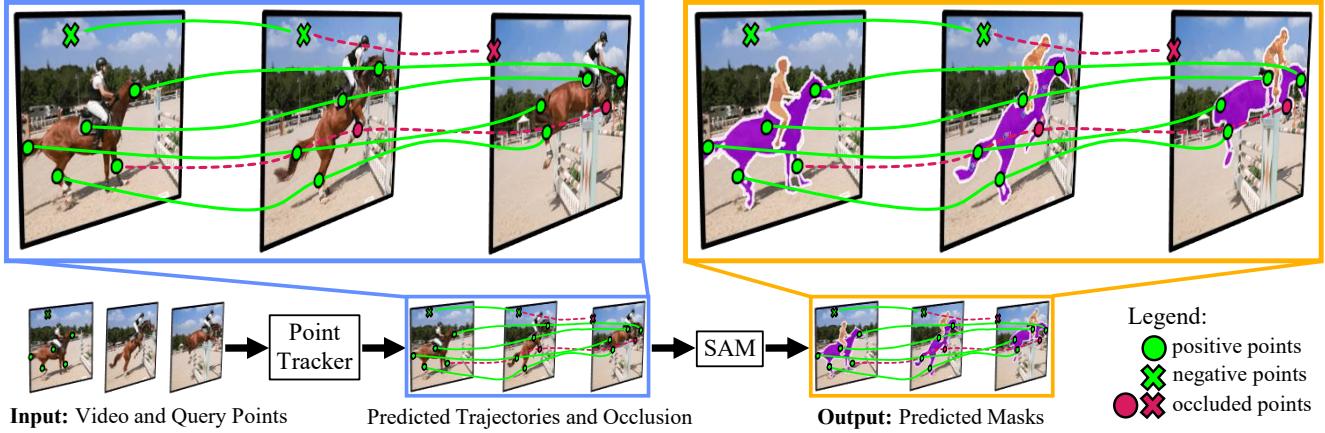


Figure 1. Segment Anything Meets Point Tracking (SAM-PT). SAM-PT is a *point-centric* method that utilizes sparse point propagation for interactive video segmentation, enabling easier interaction and faster annotation. We extend SAM [21] with long-term point trackers to effectively operate on videos in a *zero-shot* manner. SAM-PT takes user clicks as “query points” which either denote the target object (positive points) or designate non-target segments (negative points). The points are tracked throughout the video using point trackers that propagate the query points to all video frames, producing trajectory predictions and occlusion scores. SAM is subsequently prompted with the non-occluded points in the trajectories as to output a segmentation mask for each video frame independently. The propagated points can be further edited for accurate segmentation and tracking.

Abstract

The Segment Anything Model (SAM) has established itself as a powerful zero-shot image segmentation model, enabled by efficient point-centric annotation and prompt-based models. While click and brush interactions are both well explored in interactive image segmentation, the existing methods on videos focus on mask annotation and propagation. This paper presents SAM-PT, a novel method for point-centric interactive video segmentation, empowered by SAM and long-term point tracking. SAM-PT leverages robust and sparse point selection and propagation techniques for mask generation. Compared to traditional object-centric mask propagation strategies, we uniquely use point propagation to exploit local structure information agnostic to object semantics. We highlight the merits of point-based tracking through direct evaluation on the zero-shot open-world Unidentified Video Objects (UVO) benchmark. Our experiments on popular video object segmentation and multi-object segmentation tracking benchmarks, including DAVIS, YouTube-VOS, and BDD100K, suggest that a point-based segmentation tracker yields better zero-shot performance and efficient interactions. We release our code that

integrates different point trackers and video segmentation benchmarks at <https://github.com/SysCV/sam-pt>.

1. Introduction

Object segmentation and tracking in videos are central pillars for a myriad of applications, including autonomous driving, robotics, and video editing. Despite significant progress made in the past few years with deep neural networks [5, 7, 43, 48], we still need to rely on expensive labels for model supervision to achieve high accuracies. Therefore, many efforts have been made on interactive video segmentation to accelerate the data labeling process and benefit artistic video editing. Those methods are usually evaluated on a simplified semi-supervised segmentation setup as a proxy for full interactive video segmentation.

The prevailing methods [5, 7] in semi-supervised Video Object Segmentation (VOS) and Video Instance Segmentation (VIS) exhibit performance gaps when dealing with unseen data, particularly in a zero-shot setting, *i.e.*, when these models are transferred to video domains they have not been trained or that encompass object categories falling outside

of the training distribution.

Further, generalizable models usually require large amounts of training data. The existing interactive video segmentation methods assume the mask of an object is given on the first frame of the testing video. While getting accurate mask is laborious, recent works [6, 21] on training foundation image segmentation models show that point-based annotation in combination with mask editing tools is a scalable approach to label exceedingly large amounts of data. Despite its success on images in terms of labeling efficiency and accuracy, point-centric interactive segmentation has received scant attention in the video domains.

In this paper, we aim to achieve both domain generalizability and labeling efficiency for interactive video segmentation. Our insights are two-fold. First, foundation models in image segmentation are available, such as Segment Anything Model (SAM) [21]. SAM, trained on 11 million images and 1 billion object masks, has impressive zero-shot generalization capabilities. The model also supports point prompts as additional inputs for interactive image segmentation and produces high-quality masks. Second, we witnessed significant recent progress in point tracking [14–16, 19, 39, 53]. Those tracking methods, once trained, can propagate points across video frames on diverse domains.

Therefore, we introduce SAM-PT (Segment Anything Meets Point Tracking), depicted in Fig. 1. This is the first method to utilize sparse point tracking combined with SAM for video segmentation, offering a new perspective on solving the problem. Instead of employing object-centric dense feature matching or mask propagation, we propose a point-centric approach that capitalizes on tracking points using rich local structure information embedded in videos. It only requires sparse points annotation to denote the target object in the first frame and provides better generalization to unseen objects. This approach also helps preserve the inherent flexibility of SAM while extending its capabilities effectively to video segmentation. Similar to the data annotation process in SAM, our point-centric approach can be potentially integrated with the existing mask-based approaches in real-world applications.

SAM-PT prompts SAM with sparse point trajectories predicted using state-of-the-art point trackers, such as Co-Tracker [19], harnessing their versatility for video segmentation. We identified that initializing points to track using K-Medoids cluster centers from a mask label was the strategy most compatible with prompting SAM. Tracking both positive and negative points enables the clear delineation of target objects from their background. To further refine the output masks, we propose multiple mask decoding passes that integrate both types of points. In addition, we devised a point reinitialization strategy that increases tracking accuracy over time. This approach involves discarding points that have become unreliable or occluded, and adding points from object parts or segments that become visible in later frames, such as when the object rotates.

We evaluate SAM-PT on multiple setups including semi-supervised, open-world, and fully interactive video segmentation. Our method achieves stronger performance than existing zero-shot methods by up to 5.0% on DAVIS, 2.0% on YouTube-VOS, and 7.3% on BDD100K, while also surpassing a fully-supervised VIS method [46] on UVO by 6.7 points. We also set up a new benchmark for interactive point-based video segmentation to simulate the process of manually labeling the whole video. In this setup, SAM-PT significantly reduces annotation effort, approaching the performance of fully supervised approaches and underscoring its practicality. This comes without the need for any video segmentation data during training, underscoring the robustness and adaptability of our approach, and indicating its potential to enhance progress in video segmentation tasks, particularly in zero-shot scenarios.

2. Related Work

Point Tracking for Video Segmentation. Classical feature extraction and tracking methods such as Lucas-Kanade [26], Tomasi-Kanade [36], Shi-Tomasi [34], SIFT [25], and SURF [1], as well as newer methods such as LIFT [49], SuperPoint [12], and SuperGlue [33], have all demonstrated proficiency in identifying or tracking sparse features and establishing long-range correspondences. Nonetheless, these techniques often falter in dynamic, non-rigid environments. While flow-based approaches such as RAFT [35] offer improvements, they too struggle with maintaining long-term point accuracy due to error accumulation and occlusions. Addressing these shortcomings, recent innovations such as PIPS [16], PIPS++ [53], OmniMotion [39], TAPIR [15], and the state-of-the-art CoTracker [19], optimize for robust long-term trajectories and effectively manage occlusions. Our work is unique in applying these methods to guide image segmentation models for video segmentation tasks.

Segment and Track Anything Models. SAM [21] is a foundation model for image segmentation that showcases impressive zero-shot capabilities. Its extension, HQ-SAM [20], improves mask quality for complex objects but is not designed for video tasks. TAM [46] and SAM-Track [11] attempt to extend SAM to video segmentation by integrating the state-of-the-art fully-supervised XMem [7] and DeAOT [47] mask trackers, respectively, yet they lack in zero-shot scenarios.

Zero-Shot VOS / VIS. Generalist models such as Painter [41] apply visual prompting to various tasks but demonstrate limited performance in video segmentation. On the other hand, SegGPT [42] also uses visual prompting and competes closely with our method on some datasets. Other approaches, such as STC [18] and DINO [4], perform VOS through feature matching. Our approach distinguishes itself by taking the point-centric approach to enhance performance on VOS benchmarks in a zero-shot setting.

Interactive VOS. Interactive VOS has shifted from labor-

intensive manual annotations to more user-friendly interaction methods, such as scribbles, clicks, and drawings, enabling rapid and intuitive video editing [3, 17, 27, 28, 37]. Among these, MiVOS [8] stands out for its modular design that decouples mask generation from propagation, effectively incorporating user interactions from diverse interaction modalities. Unlike MiVOS and other fully-supervised methods, SAM-PT is the first to use point propagation instead of mask propagation and thus operates effectively in zero-shot settings. Our interactive point-based video segmentation study emphasizes the simplicity and efficacy of point interactions and differs from common scribble-based benchmarking [2] or in-distribution user studies [8].

3. Method

We propose SAM-PT for addressing video segmentation tasks in a zero-shot setting. SAM-PT combines the strengths of the Segment Anything Model (SAM), a foundation model for image segmentation, and prominent point trackers, such as PIPS [16] and CoTracker [19], to enable interactive tracking of anything in videos. Sec. 3.1 briefly describes the background knowledge about SAM. Sec. 3.2 then introduces our SAM-PT method with its four constituent steps. Finally, Sec. 3.3 analyzes and highlights the method’s novelty as the first point-centric interactive video segmentation method compared to existing works.

3.1. Preliminaries: SAM

Whereas in computer vision “zero-shot (learning)” usually refers to the study of generalization to unseen object categories in image classification [22], we follow prior work [21, 31] and rather employ the term in a broader sense and explore generalization to unseen datasets.

The Segment Anything Model (SAM) [21] is a novel vision foundation model designed for promptable image segmentation. SAM is trained on the large-scale SA-1B dataset, which contains 11 million images and over 1 billion masks. SA-1B has 400 times more masks than any prior segmentation dataset. This extensive training set facilitates SAM’s impressive zero-shot generalization capabilities. SAM has showcased its ability to produce high-quality masks from a single foreground point and has demonstrated robust generalization capacity on a variety of downstream tasks under a zero-shot transfer protocol using prompt engineering. These tasks include, but are not limited to, edge detection, object proposal generation, and instance segmentation.

SAM comprises three main components: an image encoder, a flexible prompt encoder, and a fast mask decoder. The image encoder is a Vision Transformer (ViT) backbone and processes high-resolution 1024×1024 images to generate an image embedding of 64×64 spatial size. The prompt encoder takes sparse prompts as input, including points, boxes, and text, or dense prompts such as masks, and translates these prompts into c -dimensional tokens. The

lightweight mask decoder then integrates the image and prompt embeddings to predict segmentation masks in real-time, allowing SAM to adapt to diverse prompts with minimal computational overhead.

3.2. Ours: SAM-PT

While SAM shows impressive capabilities in image segmentation, it is inherently limited in handling video segmentation tasks. Our Segment Anything Meets Point Tracking (SAM-PT) approach effectively extends SAM to videos, offering robust video segmentation without requiring training on any video segmentation data.

SAM-PT is illustrated in Fig. 2 and is primarily composed of four steps: **1)** selecting query points for the first frame; **2)** propagating these points to all video frames using point trackers; **3)** using SAM to generate per-frame segmentation masks based on the propagated points; **4)** optionally reinitializing the process by sampling query points from the predicted masks. We next elaborate on these four steps.

1) Query Points Selection. The process begins with defining query points in the first video frame, which either denote the target object (positive points) or designate the background and non-target objects (negative points). Users can manually and interactively provide query points, or they may be derived from a ground truth mask. For example, in the case of semi-supervised video object segmentation, the ground truth mask is provided for the first frame where the object appears. We derive the query points from ground truth masks using different point sampling techniques by considering their geometrical locations or feature dissimilarities, as depicted in Fig. 3. These sampling techniques are:

- **Random Sampling:** An intuitive approach where query points are randomly selected from the ground truth mask.
- **K-Medoids Sampling:** This technique takes the cluster centers of K-Medoids clustering [29] as query points to ensure good coverage of different parts of the object and robustness to noise and outliers.
- **Shi-Tomasi Sampling:** This method extracts Shi-Tomasi corner points from the image under the mask as they have been shown to be good features to track [34].
- **Mixed Sampling:** A hybrid method combining the above techniques since it might benefit from the unique strengths of each.

While each method contributes distinct characteristics that influence the model’s performance, our ablation study reveals that K-Medoids sampling yields the best results with good coverage of various segments of the complete object. Shi-Tomasi sampling follows closely, indicating their respective strengths in this context. The selection and arrangement of these points considerably affect the overall video segmentation performance, thus determining the optimal method is crucial.

2) Point Tracking. Initiated with the query points, we employ robust point trackers to propagate the points across

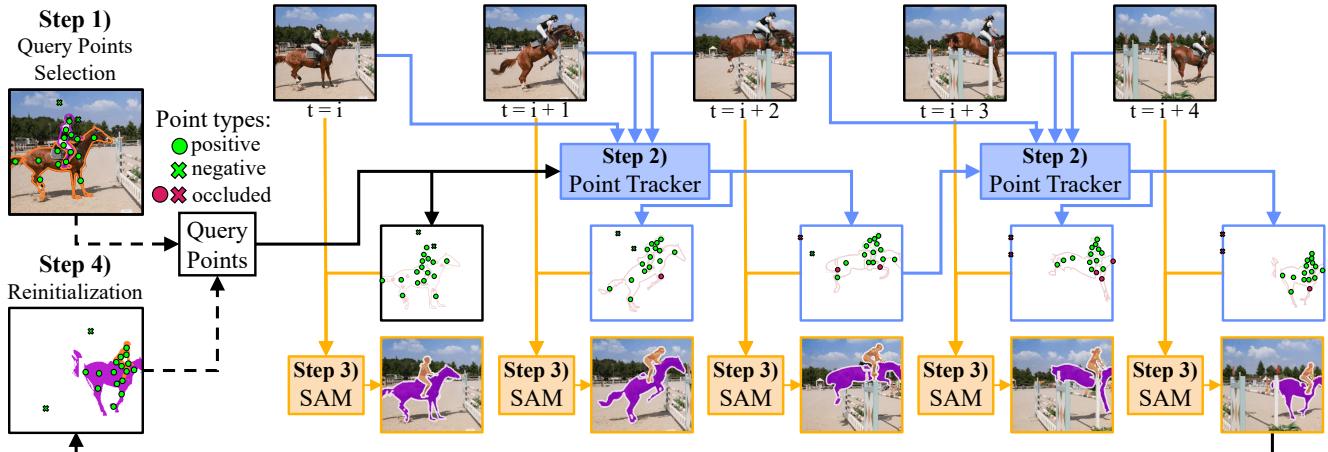


Figure 2. Segment Anything Meets Point Tracking (SAM-PT) overview. The essence of SAM-PT is to extend image segmentation foundation models to effectively operate on videos. SAM-PT has four steps: **1) Query Points Selection.** It starts with first-frame query points which denote the target object (positive points) or designate non-target segments (negative points). These points are provided by the user or derived from a ground truth mask. **2) Point Tracking.** Initiated with the query points, our approach leverages point trackers to propagate the points across video frames, predicting point trajectories and occlusion scores. **3) Segmentation.** The trajectories are then used to prompt the Segment Anything Model (SAM) and output per-frame mask predictions. **4) Point Tracking Reinitialization.** Optionally, the predicted masks are used to reinitialize the query points and restart the process when reaching a prediction horizon h . Reinitialization helps by getting rid of unreliable points and adding points to object segments that become visible in later frames.

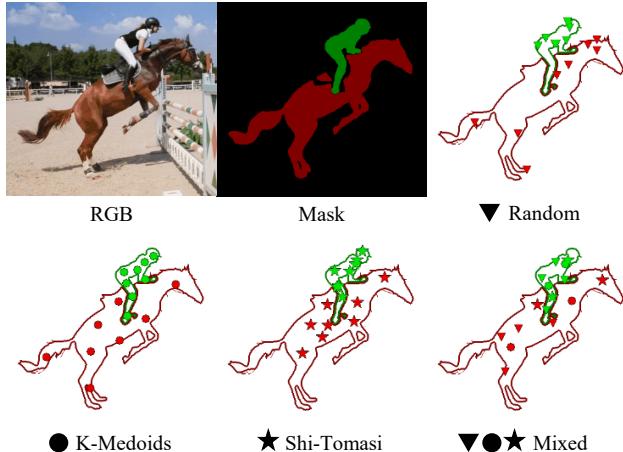


Figure 3. Positive Point Sampling. For an image paired with either a ground truth or predicted segmentation mask, positive points are sampled from within the mask area using one of the following point sampling methods: Random, K-Medoids [29], Shi-Tomasi [34], or Mixed. Notably, Random Sampling and K-Medoids Sampling only require the segmentation mask for input, not the corresponding input image. For negative points, we always use Mixed Sampling on the target object’s background mask.

all frames in the video, resulting in point trajectories and occlusion scores. We adopt point trackers such as PIPS [16] and the state-of-the-art CoTracker [19] to propagate the points as they show moderate robustness toward long-term tracking challenges such as object occlusion and re-appearance. Long-term point trackers are also shown more effective than methods such as chained optical flow propagation or first-frame correspondences in our experiments.

3) Segmentation. In the predicted trajectories, the non-

occluded points serve as indicators of where the target object is throughout the video. This allows us to use the non-occluded points to prompt SAM, as illustrated in Fig. 4, and leverage its inherent generalization ability to output per-frame segmentation mask predictions. Unlike conventional tracking methods that require training or fine-tuning on video segmentation data, our approach excels in zero-shot video segmentation tasks.

We combine positive and negative points by calling SAM in two passes. In the initial pass, we prompt SAM exclusively with positive points to define the object’s initial localization. Subsequently, in the second pass, we prompt SAM with both positive and negative points along with the previous mask prediction. Negative points provide a more nuanced distinction between the object and the background and help by removing wrongly segmented areas.

Lastly, we execute a variable number of mask refinement iterations by repeating the second pass. This utilizes SAM’s capacity to refine vague masks into more precise ones. Based on our ablation study, this step notably improves video object segmentation performance.

4) Point Tracking Reinitialization. We optionally execute a reinitialization of the query points using the predicted masks once a prediction horizon of $h = 8$ frames is reached. Upon reaching this horizon, we have h predicted masks and will take the last one to sample new points. At this stage, all previous points are discarded and substituted with the newly sampled points. Following this, steps 1) through 4) are repeated with the new points, starting from the horizon timestep where reinitialization occurs. The steps are iteratively executed until the entire video is processed. The reinitialization process serves to enhance tracking accuracy

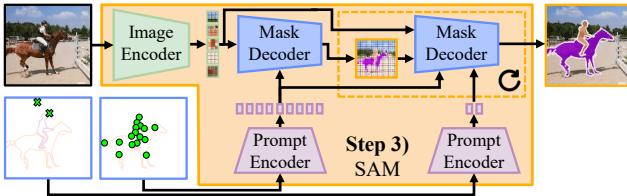


Figure 4. Interacting with SAM in SAM-PT. In the first pass, SAM is prompted exclusively with positive points to define the object’s initial localization. In the second pass, both positive and negative points along with the previous mask prediction are fed to the same mask decoder for further mask refinement. The negative points remove segments from the background and neighboring objects and notably help in cases when the point tracker mistakenly predicts positive points off the target object. The second pass is repeated iteratively to get a refined segmentation mask.

over time by discarding unreliable or occluded points while incorporating points from object segments that become visible later in the video. Other reinitialization variants are discussed in our Supplementary Material.

3.3. SAM-PT vs. Object-centric Mask Propagation

With sparse point tracking combined with prompting SAM, SAM-PT distinguishes itself from traditional video segmentation methods that depend on dense object mask propagation, as noted in Tab. 1. To propagate the first-frame GT label to the remaining video frames, traditional techniques commonly use feature matching with masks cached to a mask memory [7, 11, 46, 48], frame-by-frame feature matching [4, 18], optical flow [45], and, recently, in-context visual prompting [41, 42]. In contrast, SAM-PT introduces a unique approach to video object segmentation, employing the robust combination of point tracking with SAM, which is inherently designed to operate on sparse point prompts.

The point propagation strategy of SAM-PT offers several advantages over traditional object-centric tracking methods. First, point propagation exploits local structure context that is agnostic to global object semantics. This enhances our model’s capability for zero-shot generalization, an advantage that, coupled with SAM’s inherent generalization power, allows for tracking diverse objects in diverse environments, such as on the UVO benchmark. Second, SAM-PT allows for a more compact object representation with sparse points, capturing enough information to characterize the object’s segments/parts effectively. Finally, the use of points is naturally compatible with SAM, an image segmentation foundation model trained to operate on sparse point prompts, offering an integrated solution that aligns well with the intrinsic capacities of the underlying model.

Comparing SAM-PT with conventional methods in Tab. 1, SAM-PT emerges as superior or comparable to methods that refrain from utilizing video segmentation data during training. However, there is a performance gap that exists between such methods and those that leverage video segmentation training data in the same domain, such as XMem [7] or DeAOT [48]. Further, the potential of our

Table 1. Comparative analysis of semi-supervised Video Object Segmentation methods. Our approach, SAM-PT, introduces *sparse point propagation*, a compact mask representation that uses local structure information agnostic to object semantics. It outperforms other non-video-data-dependent methods, achieving top \mathcal{J} & \mathcal{F} scores on DAVIS 2016 and 2017, and the highest \mathcal{G} score on YouTube-VOS 2018. The comparison considers the reliance on video mask data during training, zero-shot learning setting, initial frame label requirements, and label propagation techniques used.

Method	Video Mask	Zero-Shot	Frame Init.	Propagation	DAVIS 2016	DAVIS 2017	YTVOS 2018
SiamMask [38]	✓	✗	Box	Feature Correlation	69.8	56.4	-
QMRA [24]	✓	✗	Box	Feature Correlation	85.9	71.9	-
TAM [46]	✓	✗	Points	Feature Matching	88.4	-	-
SAM-Track [11]	✓	✗	Points	Feature Matching	92.0	-	-
DEVA [10]	✓	✗	Mask	Feature Matching	-	87.6	-
XMem [7]	✓	✗	Mask	Feature Matching	92.0	87.7	86.1
DeAOT [48]	✓	✗	Mask	Feature Matching	92.9	86.2	86.2
Painter [41]	✗	✓	Mask	Mask Prompting	-	34.6	24.1
STC [18]	✗	✓	Mask	Feature Matching	-	67.6	-
DINO [4]	✗	✓	Mask	Feature Matching	-	71.4	-
SegGPT [42]	✗	✓	Mask	Mask Prompting	82.3	75.6	74.7
SAM-PT (ours)	✗	✓	Points	Points Prompting	84.3	79.4	76.2

model extends beyond video object segmentation to other tasks, such as Video Instance Segmentation (VIS), thanks to the inherent flexibility of our point propagation strategy.

4. Experiments

4.1. Datasets

We evaluate our method on four VOS datasets: DAVIS 2016, DAVIS 2017 [30], YouTube-VOS 2018 [44], and MOSE 2023 [13]. DAVIS 2017 is also used in our interactive point-based video segmentation study. We additionally devise a VOS dataset from BDD100K [50]. For VIS, We evaluate our method on the class-agnostic dense video instance segmentation task of the UVO v1.0 [40] dataset. UVO v1.0 is a VIS dataset aiming for open-world segmentation, where objects of any category, including those unseen in training, are identified and segmented.

4.2. Implementation Details

Training Data. For our experiments, we use pre-trained checkpoints provided by the respective authors for the point trackers (PIPS [32], CoTracker [19], etc.) and SAM. PIPS and CoTracker have been trained exclusively on synthetic data, PIPS on FlyingThings++ [16] and CoTracker on TAP-Vid-Kubric [14]. SAM has been trained on the large-scale SA-1B dataset, the largest image segmentation dataset to date. HQ-SAM is further trained on the HQ-Seg-44k [20]. Noteworthy, none of these datasets contain video segmentation data, nor do they intersect with any datasets we use for evaluation, situating our model within a zero-shot setting.

Interactive Point-Based Video Segmentation. To assess the interactive capabilities of SAM-PT, we simulate user refinement of video segmentation results through point additions and removals. We compare three methods: a non-tracking approach using SAM alone, an online method making one pass through the video with a target IoU quality, and an offline method employing multiple passes that

Table 2. We report the mean performance and standard deviation across eight runs on the validation subset of DAVIS 2017 to study the impact of different point trackers.

Point Tracker	DAVIS 2017 Validation [30]		
	$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$
SuperGlue [33]	28.4 \pm 3.1	24.7 \pm 2.4	32.0 \pm 3.8
TapNet [14]	60.9 \pm 0.2	58.2 \pm 0.3	63.5 \pm 0.2
RAFT [35]	63.0 \pm 0.6	60.7 \pm 0.6	65.4 \pm 0.5
PIPS++ [53]	73.2 \pm 0.5	69.9 \pm 0.5	76.6 \pm 0.5
PIPS [16]	76.3 \pm 0.6	73.6 \pm 0.6	78.9 \pm 0.6
TAPIR [15]	76.7 \pm 0.3	73.8 \pm 0.4	79.7 \pm 0.3
CoTracker [19]	77.6\pm0.7	74.8\pm0.7	80.4\pm0.7

progressively aim at higher IoU quality. These methods are detailed in the Supplementary Material, which includes pseudocode for each simulation variant.

4.3. Ablation Study

Our ablation experiments on the DAVIS 2017 validation subset assessed different aspects of SAM-PT’s design. Despite the valuable insights, we acknowledge the dataset’s limited scope in representing diverse and complex segmentation challenges, such as occlusions and varying environmental conditions, may constrain the generalizability of our ablation’s findings. Future investigations could benefit from a more varied validation set, potentially sourced from the YouTube-VOS 2018 training dataset, to enhance robustness.

Our findings, detailed in Tab. 2, underscore SAM-PT’s adaptability across leading long-term point trackers – PIPS [16], TAPIR [15], and notably CoTracker [19], which excelled due to its precise point tracking and reliable occlusion predictions. In contrast, PIPS++ [53] lagged despite being a more recent iteration of PIPS[16], due to the lack of occlusion prediction which is important for the effective use of point tracking for segmentation. TapNet [14] struggled due to less effective temporal consistency and high-resolution inputs. Traditional methods such as SuperGlue [33] and RAFT [35], which, although proficient in their respective domains, either struggle with the dynamic and deformable aspects of video scenes or cannot handle occlusion, highlighting the specialized efficacy of long-term trackers in the video segmentation landscape.

In Tab. 3, we tested SAM-PT with various settings using PIPS as the tracker. We found that using eight positive points per object instead of just one improved our scores significantly by 33.4 points because one point often wasn’t enough for unambiguously prompting SAM. Selecting points with K-Medoids was slightly better than random and matched Shi-Tomasi, giving a boost of 1.8 points. Incorporating negative points besides positive points helped when trackers made mistakes, such as losing track of an object, by improving scores by another 1.8 points. Adding iterative refinement smoothed out mask quality and fixed some errors, adding another 2.2 points to our performance. We tried filtering out unreliable points with patch similarity, but this did not work well as it ended up removing too many points. Finally, although reinitializing points did not

Table 3. Ablation study on the validation subset of DAVIS 2017 on the impact of different SAM-PT configurations using PIPS [16] as the point tracker. PSM: point selection method. PP: positive points per mask. NP: negative points per mask. IRI: iterative refinement iterations. PS: point similarity filtering. RV: reinitialization variant.

PSM	SAM-PT Configuration (using PIPS)					$\mathcal{J} \& \mathcal{F} \uparrow$	Gain
	PP	NP	IRI	PS	RV		
Random	1	0	0	\times	\times	37.1 \pm 21.7	
Random	8	0	0	\times	\times	70.5 \pm 1.4	+33.4
K-Medoids	8	0	0	\times	\times	72.3 \pm 1.2	+1.8
Shi-Tomasi	8	0	0	\times	\times	72.0 \pm 0.3	
Mixed	8	0	0	\times	\times	70.6 \pm 0.8	
K-Medoids	8	1	0	\times	\times	74.1 \pm 0.7	+1.8
K-Medoids	8	1	12	\times	\times	76.3 \pm 0.6	+2.2
K-Medoids	8	1	12	\checkmark	\times	72.7 \pm 2.0	none
K-Medoids	8	72	12	\times	A	76.8 \pm 0.7	+0.5
K-Medoids	8	1	12	\times	B	76.1 \pm 0.4	
K-Medoids	8	1	0	\times	C	75.5 \pm 0.7	
K-Medoids	8	1	12	\times	D	76.4 \pm 0.3	

help significantly in the initial tests, it did show benefits on other datasets such as MOSE and UVO, helping us recover from tracker errors by discarding incorrect and adding fresh points as well as detecting that the object has disappeared and the tracking should be halted.

Extended ablation experiments and discussions can be found in the Supplementary Material, including complete ablation results for PIPS and CoTracker, the choice of the SAM backbone, and SAM’s lightweight variants that reveal trade-offs between performance and inference speed.

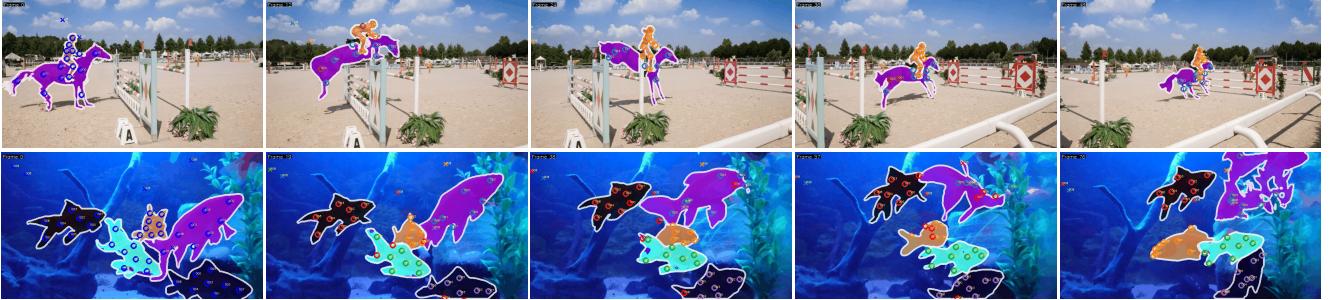
4.4. Video Object Segmentation

Performance Overview. Our SAM-PT method, utilizing HQ-SAM and CoTracker, sets a new standard in zero-shot video object segmentation on the DAVIS 2017 dataset with a mean $\mathcal{J} \& \mathcal{F}$ score of 79.4, outperforming SegGPT’s 75.6, DINO’s 71.4, and Painter’s 34.6 as shown in Tab. 4. On the easier DAVIS 2016 validation set, our method achieves 84.3, surpassing SegGPT’s 82.3, showcasing the strength of our approach even in less complex scenarios, as detailed in the Supplementary Material.

For the YouTube-VOS 2018 validation set, we achieve the highest performance among zero-shot methods with 76.2 against SegGPT’s 74.7 and Painter’s 24.1, indicating robust generalizability across various video segmentation benchmarks (Tab. 5). In the semi-supervised VOS on BDD100K’s validation set, our method outperforms SegGPT for non-transient objects but also surpasses the fully-supervised XMem across nearly all object visibility durations. The detailed breakdown is provided in Tab. 6.

On the MOSE 2023 validation set, our performance remains competitive with SegGPT, with exact figures available in the Supplementary Material.

Qualitative Analysis. Our visualizations in Fig. 5a and Fig. 5b demonstrate successful segmentation on the DAVIS 2017 dataset and underscore our method’s ability to perform zero-shot video segmentation on unseen content, such



(a) Successful segmentation cases for SAM-PT using 8 positive and 1 negative point.



(b) Successful segmentation cases for SAM-PT with 8 positive and 72 negative points and reinitialization enabled.



(c) Failure cases for SAM-PT where challenges such as occlusions and thin object structures lead to tracking errors.

Figure 5. Visualization of SAM-PT on DAVIS 2017 [30]. The method shows its capability to segment and track objects using the initial masks from the first frame, with circles denoting positive points and crosses negative points. Red symbols indicate occlusion prediction.



Figure 6. Successful segmentation using SAM-PT on short clips from “Avatar: The Last Airbender”. Although our method has never seen data from Avatar, an anime-influenced animated television series, it segments and tracks various objects in short clips.

as clips from the anime-influenced series “Avatar: The Last Airbender” in Fig. 6. These examples highlight the versatility and adaptability of SAM-PT.

Limitations and Challenges. Our method excels in zero-shot video object segmentation but faces challenges with point tracker reliability in complex scenarios, such as occlusions and fast-moving objects, as shown in Fig. 5c. While point reinitialization and negative point strategies offer some improvement, interactive use significantly bridges the performance gap with trained methods. This interactive potential is successfully demonstrated in Sec. 4.6, where user intervention enhances segmentation accuracy.

4.5. Video Instance Segmentation

Given the same mask proposals, SAM-PT outperforms TAM [46] significantly, as shown in Tab. 7, even though SAM-PT was not trained on any video segmentation data. TAM is a concurrent approach combining SAM and XMem [7], where XMem was pre-trained on BL30K [9] and trained on DAVIS and YouTube-VOS, but not on UVO. On the other hand, SAM-PT combines SAM with the PIPS or CoTracker point tracking method, both of which have not been trained on video segmentation tasks.

4.6. Interactive Point-Based Video Segmentation

Building upon our method’s strengths observed in standard benchmarks, this study evaluates the responsiveness of SAM-PT to human input, aiming to understand its practi-

Table 4. Quantitative results on the DAVIS 2017 [30] validation set for semi-supervised VOS. Performance is reported for different methods, including our SAM-PT and HQ-SAM-PT, with and without the reinitialization strategy (Reinit) and using different point trackers. Our method outperforms other zero-shot methods.

Method	Tracker	Reinit	DAVIS 2017 Validation [30]		
			$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$
(a) trained on video segmentation data					
MiVOS [8]	-	-	84.5	81.7	87.4
DeAOT [48]	-	-	86.2	83.1	89.2
DEVA [10]	-	-	87.6	84.2	91.0
XMem [7]	-	-	87.7	84.0	91.4
(b) not trained on video segmentation data (<i>zero-shot</i>)					
Painter [41]	-	-	34.6	28.5	40.8
DINO [4]	-	-	71.4	67.9	74.9
SegGPT [42]	-	-	75.6	72.5	78.6
SAM-PT (ours)	PIPS [16] PIPS [16] CoTracker [19] CoTracker [19]	✗ ✓ ✗ ✓	76.3 \pm 0.6 76.6 \pm 0.7 77.6 \pm 0.7 77.4 \pm 1.0	73.6 \pm 0.6 74.4 \pm 0.8 74.8 \pm 0.7 74.5 \pm 1.0	78.9 \pm 0.6 78.9 \pm 0.6 80.4 \pm 0.7 80.3 \pm 1.1
HQ-SAM-PT (ours)	PIPS [16] PIPS [16] CoTracker [19] CoTracker [19]	✗ ✓ ✗ ✓	77.2 \pm 0.5 77.0 \pm 0.7 79.4 \pm 0.6 77.7 \pm 0.8	74.7 \pm 0.5 74.8 \pm 0.8 76.5 \pm 0.6 74.6 \pm 0.9	79.8 \pm 0.4 79.2 \pm 0.6 82.3 \pm 0.5 80.8 \pm 0.7

Table 5. Quantitative results in semi-supervised VOS on the validation subset of YouTube-VOS 2018. Metrics are reported separately for “seen” and “unseen” classes, with \mathcal{G} being the overall average score over the metrics.

Method	Tracker	Reinit	YouTube-VOS 2018 Validation [44]				
			\mathcal{G}	\mathcal{J}_s	\mathcal{F}_s	\mathcal{J}_u	\mathcal{F}_u
(a) trained on video segmentation data							
XMem [7]	-	-	86.1	85.1	89.8	80.3	89.2
DeAOT [48]	-	-	86.2	85.6	90.6	80.0	88.4
(b) not trained on video segmentation data (<i>zero-shot</i>)							
Painter [41]	-	-	24.1	27.6	35.8	14.3	18.7
SegGPT [42]	-	-	74.7	75.1	80.2	67.4	75.9
SAM-PT (ours)	PIPS [16] PIPS [16] CoTracker [19] CoTracker [19]	✗ ✓ ✗ ✓	67.0 \pm 0.3 67.5 \pm 0.2 74.0 \pm 0.3 71.5 \pm 0.4	68.6 \pm 0.2 69.0 \pm 0.4 73.3 \pm 0.2 71.0 \pm 0.3	71.2 \pm 0.1 69.9 \pm 0.3 76.0 \pm 0.2 72.8 \pm 0.3	61.0 \pm 0.5 63.2 \pm 0.4 70.0 \pm 0.4 68.3 \pm 0.7	67.4 \pm 0.4 67.8 \pm 0.5 76.7 \pm 0.4 73.9 \pm 0.7
HQ-SAM-PT (ours)	CoTracker [19]	✗	76.2 \pm 0.1	75.3 \pm 0.1	78.4 \pm 0.2	72.1 \pm 0.2	79.0 \pm 0.2

cal utility for interactive video annotation tasks. We benchmarked SAM-PT’s interactive performance against a baseline SAM approach that does not utilize point tracking. The results, visualized in Fig. 7, show that SAM-PT significantly outperforms the baseline, particularly when employing the offline checkpoint strategy. Although this offline method initially starts at a lower performance due to setup costs, it quickly exceeds the online method’s performance by benefiting from a cumulative optimization approach.

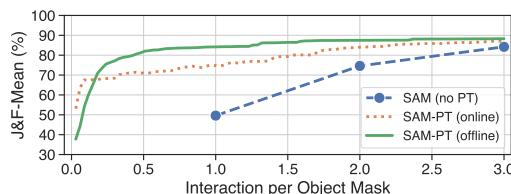


Figure 7. Interactive segmentation performance on the DAVIS 2017 [30] validation set. SAM-PT’s offline strategy notably outperforms the baseline SAM, demonstrating efficient video annotation with minimal user intervention.

Table 6. This table shows the semi-supervised VOS performance on BDD100K’s validation set, comparing our HQ-SAM-PT method using CoTracker as the point tracker (without reinitialization) against the zero-shot SegGPT and the fully-supervised XMem. Performance metrics include the average $\mathcal{J} \& \mathcal{F}$ measure for object visibility durations categorized as short (1-5 frames), medium (6-30 frames), and long (31+ frames). Our approach demonstrates superior results over SegGPT for non-transient objects and over XMem across all visibility durations except for long-term object tracking.

BDD100K VOS Validation [50]						
Method	$\mathcal{J} \& \mathcal{F}$	\mathcal{J}	\mathcal{F}	$\mathcal{J} \& \mathcal{F}$	$\mathcal{J} \& \mathcal{F}$	$\mathcal{J} \& \mathcal{F}$
				Short	Medium	Long
(a) trained on video segmentation data, but not on BDD100K						
XMem [7]	76.6	74.5	78.7	79.3	78.6	63.7
(b) not trained on video segmentation data (<i>zero-shot</i>)						
SegGPT [42]	81.5	81.2	81.8	96.1	78.6	52.0
HQ-SAM-PT (ours)	<u>81.0</u>	<u>80.1</u>	<u>81.8</u>	<u>91.8</u>	79.9	<u>55.8</u>

Table 7. Results on the validation split of UVOD [40] VideoDenseSet v1.0. SAM-PT outperforms TAM [46] even though the former was not trained on any video segmentation data. TAM is a concurrent approach combining SAM [21] and XMem [7], where XMem was pre-trained on BL30K [9] and trained on DAVIS [30] and YouTube-VOS [44], but not on UVOD. On the other hand, SAM-PT combines SAM with point trackers, both of which have not been trained on any video segmentation tasks.

Method	Tracker	Reinit	AR100	ARs	ARm	ARI	AP
			(a) trained on video segmentation data, including UVOD’s training subset				
Mask2Former	-	-	35.4	—	—	—	27.3
VIS [51]	-	-	41.2	—	—	—	32.7
ROVIS [51]	-	-	—	—	—	—	—
(b) trained on video segmentation data							
TAM [46]	-	-	24.1	21.1	32.9	31.1	1.7
(c) not trained on video segmentation data (<i>zero-shot</i>)							
SAM-PT (ours)	PIPS [16] PIPS [16] CoTracker [19] CoTracker [19]	✗ ✓ ✗ ✓	28.8 30.8 29.5 29.8	23.3 25.1 25.3 25.1	40.8 44.1 39.0 40.6	48.3 49.2 44.1 45.6	6.7 6.5 5.8 6.2

These results suggest that SAM-PT substantially reduces the effort required for high-quality video annotation, bringing its performance closer to fully-supervised methods and highlighting its practical utility.

5. Conclusion

SAM-PT introduces a point-centric approach for interactive video segmentation by combining the generalization capabilities of the Segment Anything Model (SAM) and long-term point tracking. Our work fills in the gap that the point-centric approach is scarcely explored in the literature. In our experiments, SAM-PT achieves strong performance across video segmentation tasks including semi-supervised, open-world, and fully interactive video segmentation. While our method has limitations such as difficulty handling occlusions, small objects, motion blur, and inconsistencies in mask predictions, it contributes a new perspective to video object segmentation research.

References

- [1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. [2](#)
- [2] Sergi Caelles, Alberto Montes, Kevins-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. In *arXiv:1803.00557*, 2018. [3](#)
- [3] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevins-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. In *arXiv:1905.00737*, 2019. [3](#)
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. [2, 5, 8](#)
- [5] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G. Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv: 2112.10764*, 2021. [1](#)
- [6] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. In *CVPR*, pages 2617–2626, 2022. [2](#)
- [7] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. [1, 2, 5, 7, 8, 11](#)
- [8] 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. [3, 8](#)
- [9] 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. [7, 8](#)
- [10] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with de-coupled video segmentation. In *ICCV*, 2023. [5, 8, 11](#)
- [11] 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, 5](#)
- [12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. [2](#)
- [13] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip H. S. Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. *arXiv preprint arXiv: 2302.01872*, 2023. [5, 11](#)
- [14] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *NeurIPS*, 2022. [2, 5, 6](#)
- [15] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. *ICCV*, 2023. [2, 6, 11](#)
- [16] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022. [2, 3, 4, 5, 6, 8, 11, 12](#)
- [17] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Interactive video object segmentation using global and local transfer modules. In *ECCV*, 2020. [3](#)
- [18] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. [2, 5](#)
- [19] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Co-tracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. [2, 3, 4, 5, 6, 8, 11, 12](#)
- [20] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023. [2, 5, 11, 12](#)
- [21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, pages 4015–4026, 2023. [1, 2, 3, 8, 12](#)
- [22] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958. IEEE, 2009. [3](#)
- [23] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *CVPR*, 2022. [11](#)
- [24] Fanchao Lin, Hongtao Xie, Yan Li, and Yongdong Zhang. Query-memory re-aggregation for weakly-supervised video object segmentation. In *AAAI*, 2021. [5](#)
- [25] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. [2](#)
- [26] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. [2](#)
- [27] Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *CVPR*, 2020. [3](#)
- [28] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Fast user-guided video object segmentation by interaction-and-propagation networks. In *CVPR*, 2019. [3](#)
- [29] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2, Part 2):3336–3341, 2009. [3, 4](#)
- [30] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. [5, 6, 7, 8, 11, 12, 15](#)
- [31] 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. 2021. [3](#)
- [32] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80:72–91, 2008. [5](#)
- [33] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. [2, 6](#)
- [34] Jianbo Shi and Tomasi. Good features to track. In *CVPR*, 1994. [2, 3, 4](#)

- [35] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 6
- [36] Carlo Tomasi and Takeo Kanade. Detection and tracking of point. *IJCV*, 9:137–154, 1991. 2
- [37] Jue Wang, Pravin Bhat, R Alex Colburn, Maneesh Agrawala, and Michael F Cohen. Interactive video cutout. In *ToG*, 2005. 3
- [38] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 5
- [39] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *ICCV*, 2023. 2
- [40] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 5, 8
- [41] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, 2023. 2, 5, 8, 11
- [42] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. In *ICCV*, 2023. 2, 5, 8, 11
- [43] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. In *ECCV*, 2022. 1
- [44] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark, 2018. 5, 8
- [45] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021. 5
- [46] 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, 5, 7, 8, 13
- [47] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *NeurIPS*, 2022. 2
- [48] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *NeurIPS*, 2022. 1, 5, 8, 11
- [49] K. M. Yi, Eduard Trulls, Vincent Lepetit, and P. Fua. Lift: Learned invariant feature transform. *ECCV*, 2016. 2
- [50] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 8
- [51] Zitong Zhan, Daniel McKee, and Svetlana Lazebnik. Robust online video instance segmentation with track queries. *arXiv preprint arXiv: 2211.09108*, 2022. 8
- [52] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 11, 12
- [53] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 2, 6

Segment Anything Meets Point Tracking

Supplementary Material

In this supplementary, we first report experimental results on additional datasets and subsets (Appendices A to C). Then we extend and detail our ablation and report more qualitative results (Appendices D to H). Lastly, we detail on our evaluation protocols (Appendices I to L).

A. MOSE 2023

MOSE 2023 [13] is a recently introduced dataset that focuses on multi-object segmentation and tracking in complex scenes, replete with challenges such as occlusions, transient visibility of objects, extensive occlusion, etc. Our results on the validation subset suggest that SAM-PT achieves performance competitive with SegGPT [42], as shown in Tab. 8.

Table 8. Quantitative results on the MOSE 2023 validation set for semi-supervised VOS. Our method achieves performance comparable to the state-of-the-art zero-shot learning method. Note that SegGPT and SAM-PT adopt completely different training data.

Method	Tracker	Reinit	MOSE 2023 Validation [13]		
			$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$
(a) trained on video segmentation data					
RDE [23]	-	-	48.8	44.6	52.9
XMem [7]	-	-	57.6	53.3	62.0
DeAOT [48]	-	-	59.4	55.1	63.8
DEVA [10]	-	-	66.5	62.3	70.8
(b) not trained on video segmentation data (zero-shot)					
Painter [41]	-	-	14.5	10.4	18.5
SegGPT [42]	-	-	45.1	42.2	48.0
SAM-PT (ours)	PIPS [16]	✗	38.5±0.2	34.9±0.3	42.1±0.2
	PIPS [16]	✓	41.0±0.5	38.5±0.5	43.5±0.5
	CoTracker [19]	✗	41.8±0.2	38.3±0.2	45.2±0.3
	CoTracker [19]	✓	40.1±0.5	36.0±0.5	44.1±0.4
HQ-SAM-PT (ours)	TAPIR [15]	✗	42.9±0.2	38.3±0.2	47.6±0.1
	CoTracker [19]	✗	42.4±0.3	39.0±0.3	45.8±0.3
	TAPIR [15]	✗	42.1±0.1	37.6±0.1	46.7±0.1

B. DAVIS 2016

DAVIS 2016 offers a single-object VOS benchmark across 20 diverse sequences. We report results on the DAVIS 2016 validation subset in Tab. 9, in which our method achieves 84.3 points, surpassing SegGPT’s 82.3 points.

Table 9. Quantitative results on the DAVIS 2016 validation set for semi-supervised VOS. Our method achieves higher performance compared to SegGPT, both of which are zero-shot methods.

Method	Tracker	Reinit	DAVIS 2016 Validation [30]		
			$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$
SegGPT [42]	-	-	82.3	81.8	82.8
SAM-PT	PIPS [16]	✗	83.1±1.5	83.0±0.8	83.0±1.1
	PIPS [16]	✓	80.2±0.6	80.3±0.6	80.0±0.6
	CoTracker [19]	✗	83.1±0.6	83.2±0.7	82.9±0.6
	CoTracker [19]	✓	82.6±0.8	83.0±1.0	82.2±0.9
HQ-SAM-PT	CoTracker [19]	✗	84.3 ±0.9	84.9 ±1.0	83.7 ±0.9

C. DAVIS 2017 Test-dev Subset

DAVIS 2017 is a multi-object extension of its 2016 version. The video scenarios within this dataset are small but diverse. In addition to the results on the validation subset in the main manuscript, we report the performance of SAM-PT on the DAVIS 2017 test-dev subset in Tab. 10.

Table 10. Quantitative results on the DAVIS 2017 test-dev subset for semi-supervised VOS.

Method	Tracker	Reinit	DAVIS 2017 Test-dev [30]		
			$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$
PIPS [16]	✗		62.7±0.5	59.4±0.6	66.1±0.4
PIPS [16]	✓		61.5±1.1	59.3±1.0	63.8±1.2
SAM-PT	CoTracker [19]	✗	65.7±0.7	62.8±0.7	68.5±0.7
	CoTracker [19]	✓	62.0±1.2	58.8±1.2	65.1±1.1
	TAPIR [15]	✗	69.0	66.0	72.0
	TAPIR [15]	✓	64.1	61.3	66.9
HQ-SAM-PT	CoTracker [19]	✗	64.8±0.5	61.9±0.5	67.7±0.5

D. Different SAM Backbones

The SAM model’s backbone plays an important role in determining its performance and inference speed. In this experiment, we evaluated SAM with different ViT backbones: ViT-Huge (used throughout the work), ViT-Large, and ViT-Base. The results, as measured on the validation subset of DAVIS 2017 are shown in Tab. 11. Replacing ViT-Huge with ViT-Large results in only a non-significant loss in performance, SAM-PT’s overall performance nevertheless remains non-real-time. Note that the ViT-Huge number in this experiment is slightly different from the one presented in the main manuscript due to the use of different seed values.

Table 11. Performance of SAM-PT with different backbones and their inference speed in semi-supervised VOS on the validation subset of DAVIS 2017, when using PIPS as the point tracker.

SAM Backbone	$\mathcal{J} \& \mathcal{F}$	FPS
ViT-Huge	76.7 ± 0.6	1.4
ViT-Large	76.4 ± 0.6	1.8
ViT-Base	72.2 ± 0.5	2.6

E. Different SAM Variants

To cater to scenarios where inference speed is crucial, we explored lightweight variants of SAM: Light HQ-SAM [20] and MobileSAM [52]. The performance and speed trade-offs for these variants are summarized in table Tab. 12. Using the HQ-SAM [20] variant of SAM results in the highest performance of 77.64 points, whereas MobileSAM has the highest inference speed of 5.5 FPS. Using the lightweight

variants doesn't achieve real-time performance as the bottleneck of the pipeline moves to the point tracker.

Table 12. Performance of lightweight SAM variants and their inference speed in semi-supervised VOS on the validation subset of DAVIS 2017 when using PIPS [16] as the point tracker.

SAM Variant	Backbone	$\mathcal{J} \& \mathcal{F}$	FPS
HQ-SAM [20]	ViT-Huge	77.64	1.3
SAM [21]	ViT-Huge	76.65	1.4
Light HQ-SAM [20]	ViT-Tiny	71.30	4.8
MobileSAM [52]	ViT-Tiny	71.07	5.5

F. Detailed Ablation Results

We report detailed experimental results of our ablation studies for configurations using PIPS [16] as the point tracker in Tab. 14 and using CoTracker [19] in Tab. 13.

Table 13. Ablation study on the validation subset of DAVIS 2017 on the impact of different SAM-PT configurations using CoTracker [19] as the point tracker. PSM: point selection method. PP: positive points per mask. NP: negative points per mask. IRI: iterative refinement iterations. PS: point similarity filtering.

SAM-PT Config. (using CoTracker)					DAVIS 2017 Validation [30]		
PSM	PP	NP	IRI	PS	$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$
K-Medoids	16	1	12	✗	77.6 \pm 0.7	74.8 \pm 0.7	80.4 \pm 0.7
Shi-Tomasi	16	1	12	✗	74.3 \pm 0.3	72.1 \pm 0.3	76.5 \pm 0.3
Random	16	1	12	✗	76.4 \pm 1.1	73.3 \pm 1.1	79.4 \pm 1.0
Mixed	16	1	12	✗	76.4 \pm 0.6	73.7 \pm 0.5	79.2 \pm 0.6
K-Medoids	1	1	12	✗	39.0 \pm 0.9	36.1 \pm 0.9	42.0 \pm 1.0
K-Medoids	16	0	12	✗	76.8 \pm 0.7	74.1 \pm 0.7	79.6 \pm 0.7
K-Medoids	16	1	0	✗	75.3 \pm 0.6	73.2 \pm 0.5	77.3 \pm 0.6
K-Medoids	16	1	1	✗	76.6 \pm 0.6	74.3 \pm 0.6	78.9 \pm 0.6
K-Medoids	16	1	100	✗	77.5 \pm 0.7	74.7 \pm 0.7	80.3 \pm 0.7
K-Medoids	16	1	12	✓	73.8 \pm 0.8	71.1 \pm 0.8	76.5 \pm 0.8

G. Point Tracking Reinitialization

In our method, we introduce an optional reinitialization strategy. Here, the point tracker begins anew every h frames, where h represents a pre-set tracking *horizon* (e.g., 8 frames), or is dynamically determined based on SAM's mask predictions for each timestep within the horizon (e.g., using most-similar-mask-area heuristics). Upon reaching this horizon, the query points given to the tracker are reinitialized according to the mask prediction SAM outputted at the horizon frame. While this method may increase the computational load, it shows performance improvement when using the PIPS [16] point tracker in demanding video sequences, such as those in the MOSE dataset. However, our studies also suggested that the proposed reinitialization strategies hurt performance when using CoTracker as the point tracker. We primarily designed the reinitialization variants to address some of the failure cases of PIPS such as the common case of points being wrongly predicted to be

Table 14. Detailed ablation study on the validation subset of DAVIS 2017 on the impact of different SAM-PT configurations using PIPS [16] as the point tracker. PSM: point selection method. PP: positive points per mask. NP: negative points per mask. IRI: iterative refinement iterations. PS: point similarity filtering. RV: reinitialization variant.

SAM-PT Configuration (using PIPS)						DAVIS 2017 Validation [30]			
PSM	PP	NP	IRI	PS	RV	$\mathcal{J} \& \mathcal{F} \uparrow$	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$	Gain
(a) point selection method and positive points per mask									
Random	1	0	0	✗	✗	37.1 \pm 21.7	34.3 \pm 22.0	40.0 \pm 21.5	
Random	8	0	0	✗	✗	70.5 \pm 1.4	68.5 \pm 1.4	72.6 \pm 1.5	
Random	16	0	0	✗	✗	70.0 \pm 1.1	68.2 \pm 1.0	71.8 \pm 1.2	
Random	72	0	0	✗	✗	62.6 \pm 0.4	62.3 \pm 0.3	62.8 \pm 0.5	
Shi-Tomasi	1	0	0	✗	✗	20.3 \pm 0.1	18.3 \pm 0.1	22.3 \pm 0.2	
Shi-Tomasi	8	0	0	✗	✗	72.0 \pm 0.3	70.3 \pm 0.3	73.7 \pm 0.4	
Shi-Tomasi	16	0	0	✗	✗	66.6 \pm 0.4	65.7 \pm 0.5	67.6 \pm 0.4	
Shi-Tomasi	72	0	0	✗	✗	54.4 \pm 0.3	54.8 \pm 0.2	54.0 \pm 0.4	
K-Medoids	1	0	0	✗	✗	32.2 \pm 0.7	30.4 \pm 0.8	34.0 \pm 0.7	
K-Medoids	8	0	0	✗	✗	72.3 \pm 1.2	70.4 \pm 1.3	74.3 \pm 1.1	
K-Medoids	16	0	0	✗	✗	71.4 \pm 0.2	69.8 \pm 0.3	73.1 \pm 0.2	
K-Medoids	72	0	0	✗	✗	58.0 \pm 0.2	57.3 \pm 0.2	58.7 \pm 0.3	
Mixed	1	0	0	✗	✗	29.9 \pm 0.9	26.6 \pm 0.8	33.2 \pm 1.4	
Mixed	8	0	0	✗	✗	70.6 \pm 0.8	68.6 \pm 0.8	72.5 \pm 0.8	
Mixed	16	0	0	✗	✗	70.0 \pm 0.7	68.2 \pm 0.6	71.9 \pm 0.7	
Mixed	72	0	0	✗	✗	62.8 \pm 0.5	62.4 \pm 0.5	63.2 \pm 0.6	
(b) negative points per mask									
K-Medoids	8	0	0	✗	✗	72.3 \pm 1.2	70.4 \pm 1.3	74.3 \pm 1.1	
K-Medoids	8	1	0	✗	✗	74.1 \pm 0.7	72.1 \pm 0.6	76.1 \pm 0.7	+1.8
K-Medoids	8	8	0	✗	✗	74.0 \pm 0.8	71.9 \pm 0.8	76.0 \pm 0.9	
K-Medoids	8	16	0	✗	✗	73.4 \pm 0.6	71.4 \pm 0.6	75.3 \pm 0.6	
K-Medoids	8	72	0	✗	✗	72.2 \pm 0.4	70.3 \pm 0.4	74.0 \pm 0.4	
(c) iterative refinement iterations									
K-Medoids	8	1	0	✗	✗	74.1 \pm 0.7	72.1 \pm 0.6	76.1 \pm 0.7	
K-Medoids	8	1	1	✗	✗	75.7 \pm 0.7	73.4 \pm 0.7	78.1 \pm 0.6	
K-Medoids	8	1	3	✗	✗	76.0 \pm 0.6	73.4 \pm 0.7	78.6 \pm 0.7	
K-Medoids	8	1	12	✗	✗	76.3 \pm 0.6	73.6 \pm 0.6	78.9 \pm 0.6	+2.2
(d) patch similarity filtering									
K-Medoids	8	1	12	✗	✗	76.3 \pm 0.6	73.6 \pm 0.6	78.9 \pm 0.6	
K-Medoids	8	1	12	0.002	✗	72.7 \pm 2.0	70.2 \pm 1.8	75.2 \pm 2.1	
K-Medoids	8	1	12	0.01	✗	70.7 \pm 2.0	68.3 \pm 1.8	73.2 \pm 2.1	
(e) point reinitialization									
K-Medoids	8	1	0	✗	A	75.7 \pm 0.7	73.7 \pm 0.6	77.7 \pm 0.8	
K-Medoids	8	1	0	✗	B	75.8 \pm 0.6	73.5 \pm 0.9	78.1 \pm 0.3	
K-Medoids	8	1	0	✗	C	75.5 \pm 0.7	73.2 \pm 0.8	77.8 \pm 0.7	
K-Medoids	8	1	0	✗	D	75.4 \pm 0.2	73.3 \pm 0.2	77.5 \pm 0.3	
K-Medoids	8	1	12	✗	A	76.6 \pm 0.8	74.0 \pm 0.8	79.1 \pm 0.8	
K-Medoids	8	1	12	✗	B	76.1 \pm 0.4	73.5 \pm 0.5	78.6 \pm 0.3	
K-Medoids	8	1	12	✗	C	75.4 \pm 0.6	72.8 \pm 0.7	78.0 \pm 0.5	
K-Medoids	8	1	12	✗	D	76.4 \pm 0.3	74.0 \pm 0.4	78.8 \pm 0.3	
K-Medoids	8	1	12	✗	✗	76.3 \pm 0.6	73.6 \pm 0.6	78.9 \pm 0.6	
K-Medoids	8	72	0	✗	A	74.9 \pm 0.9	73.2 \pm 0.8	76.6 \pm 1.0	
K-Medoids	8	72	0	✗	B	76.0 \pm 1.1	73.9 \pm 1.1	78.1 \pm 1.1	
K-Medoids	8	72	0	✗	C	75.1 \pm 0.6	72.9 \pm 0.5	77.2 \pm 0.7	
K-Medoids	8	72	0	✗	D	75.6 \pm 1.5	73.8 \pm 1.5	77.3 \pm 1.6	
K-Medoids	8	72	12	✗	A	76.8 \pm 0.7	74.5 \pm 0.8	79.0 \pm 0.6	+0.5
K-Medoids	8	72	12	✗	B	74.8 \pm 0.8	72.1 \pm 0.9	77.6 \pm 0.7	
K-Medoids	8	72	12	✗	C	75.0 \pm 0.4	72.1 \pm 0.4	77.8 \pm 0.5	
K-Medoids	8	72	12	✗	D	75.2 \pm 1.1	72.7 \pm 1.1	77.6 \pm 1.1	

off the target object, being predicted on the background instead, but this does not work as well for other point trackers such as CoTracker that are more robust to such failures.

We explored four reinitialization strategies, each varying in how they compute the value of h :

(A) Reinit-on-Horizon-and-Sync-Masks: This straightforward variant reinitializes points after a fixed number of frames (e.g., every 8 frames). However, it may stumble if the mask is absent at the reinitialization timestep.

(B) Reinit-at-Median-of-Area-Diff: In this variant, the tracker outputs trajectory points for each frame within the horizon, and SAM predicts masks based on these trajectories. Reinitialization happens at the frame

within the horizon that has the mean mask area among the non-empty masks predicted by SAM.

(C) **Reinit-on-Similar-Mask-Area:** This method triggers reinitialization when the mask area is similar to the initial mask area.

(D) **Reinit-on-Similar-Mask-Area-and-Sync-Masks:**

This variant reinitializes when the mask area for all masks in the batch is similar to the initial mask areas, synchronizing the masks to be tracked from the same timestep. This synchronization allows for the use of negative points from other masks when querying SAM.

From our ablation investigations, we found the **(A) Reinit-on-Horizon-and-Sync-Masks** strategy to be effective with PIPS as the point tracker and **(B) Reinit-at-Median-of-Area-Diff** with CoTracker. Besides the point tracker used, the choice of reinitialization method may depend on the specific validation subset and the degree of hyperparameter tuning involved. Note that we always use reinitialization along with negative points.

H. Additional Qualitative Results

We show additional visualizations on DAVIS videos in Fig. 8. We show failure cases on clips from the anime-influenced series “Avatar: The Last Airbender” in Fig. 9.

I. VOS Evaluation Details

When evaluating on VOS, we use the provided ground truth mask for the first frame to sample the query points required by our method. Then, we give only the sampled points as input to our method, not the mask. For all datasets, we use the full-resolution data and resize it to the longest side of 1024 to match SAM’s input resolution.

J. VIS Evaluation Details

For evaluating our method on the VIS task, we leverage SAM’s automatic mask generation capacity to generate up to 100 mask proposals for the initial frame. We use the same initial masks to assess TAM [46] for a fair comparison. Our current approach does not generate new proposals in subsequent frames, thus it cannot detect new objects appearing after the first frame, unlike fully-fledged VIS methods. However, this setup allows for a straightforward comparison of zero-shot capabilities in mask propagation from the initial frame.

K. BDD100K VOS Dataset Creation

BDD100K is a large open driving video dataset with 100K videos and 10 tasks to evaluate the progress of image recognition algorithms in autonomous driving. It includes a variety of geographic, environmental, and weather conditions. We convert its annotations from the Multi-Object Tracking and Segmentation (MOTS) section into semi-supervised

VOS annotations. This section has 154, 32, and 37 videos for train, validation, and test sets, totaling 25K instances and 480K masks.

To convert the annotations, we take the ground truth mask of the first appearance of each object in the videos. This mask will be given as input to semi-supervised VOS methods which then need to predict the masks for the remaining video frames. In converting the validation subset of 32 videos and 4566 object tracks, we create 61 semi-supervised VOS datapoints of up to 100 objects per video. We limit the number of objects per video for implementation simplicity since most VOS methods expect a small number of objects per video. For example, in the DAVIS validation subset, the maximum number of objects per video is 5. During the conversion, we additionally remove the instances marked as “ignored” or “crowd” in the MOTS annotations.

L. Interactive Point-Based Video Segmentation Details

Interactive point-based video segmentation aims to refine segmentation masks with minimal user input while optimizing the Intersection over Union (IoU) with the ground truth. To evaluate the responsiveness of SAM-PT to simulated human input, we benchmark against a SAM-only baseline (Algorithm 1) and compare it with both an online (Algorithm 2) and offline (Algorithm 3) interactive SAM-PT method:

- 1. Non-tracking method (SAM only):** As a baseline, this method mimics user interactions by selecting points on each frame without any point tracking, effectively emulating the process of manual, frame-by-frame annotation using the standalone SAM model.
- 2. Online method:** This approach models a user going through the video sequentially a single time, making corrective per-frame interactions to achieve an IoU of at least 95% with the ground truth mask for each frame. These corrections include the addition or removal of points and are propagated to subsequent frames via point tracking. The user may opt to skip frames that already appear to be well-annotated or that cannot be annotated sufficiently well within the available interaction budget.
- 3. Offline method (Checkpoint Strategy):** This method takes a multi-pass approach, incrementally aiming for higher IoU thresholds with each pass through the video. Checkpoints are saved after each pass, and the latest checkpoint given the interaction budget is returned, allowing for a more global optimization approach.

Interactions are defined as the act of adding or removing a point and are executed as described in Algorithm 4. To maintain simplicity in our evaluation, skipping frames while progressing through a video is not counted as an interaction.

Algorithm 1 Non-tracking (SAM only) Method.

```
1: Input: rgbs, gt_masks, int_per_frame
2: Output: pred_masks
3: point_memory ← emptyPointMemory()
4: for i in frames do
5:   pmf ← emptyFramePointMemory()
6:   for j in int_per_frame do
7:     m ← predMask(rgbs[i], pmf)
8:     performInt(i, m, gt_masks[i], pmf)
9:   end for
10:  addToPointMemory(point_memory, i, pmf)
11: end for
12: return predAllMasks(rgbs, point_memory)
```

Algorithm 2 Online Method.

```
1: Input: rgbs, gt_masks
2: Output: pred_masks
3: max_int ← 300
4: max_int_per_frame ← 3
5: threshold ← 0.95
6: point_memory ← selectFirstPoint(gt_masks[0])
7: max_int ← max_int - 1
8: for i in frames do
9:   m ← predMask(rgbs[i], point_memory)
10:  IoU ← calculateIoU(m, gt_masks[i])
11:  if IoU ≥ threshold then
12:    continue
13:  end if
14:  performInt(i, m, gt_masks[i], point_memory)
15:  max_int ← max_int - 1
16:  if max_int ≤ 0 then
17:    break
18:  end if
19: end for
20: return predAllMasks(rgbs, point_memory)
```

Algorithm 3 Offline Method (Checkpoint Strategy).

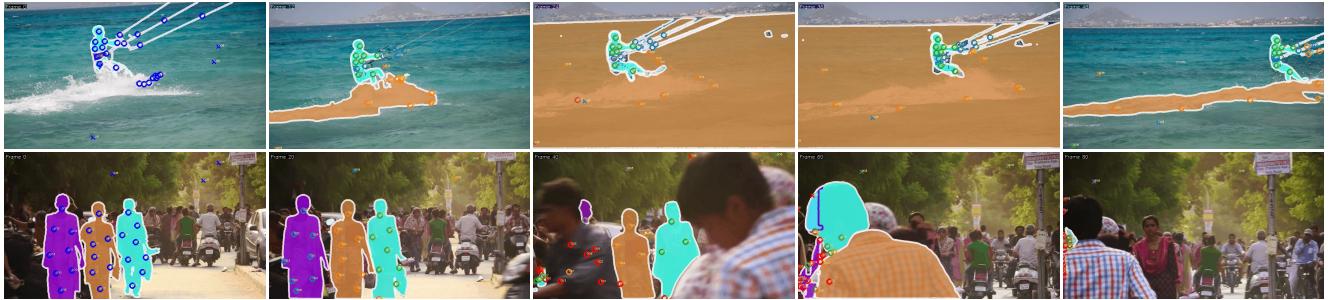
```
1: Input: rgbs, gt_masks
2: Output: pred_masks
3: max_int ← 300
4: max_int_per_frame ← 3
5: int_iou_thresholds ← [0.10, 0.20, ..., 0.95]
6: point_memory ← selectFirstPoint(gt_masks[0])
7: max_int ← max_int - 1
8: best_ckpt ← predAllMasks(rgbs, point_memory)
9: for threshold in int_iou_thresholds do
10:  for i in frames do
11:    for j in max_int_per_frame do
12:      m ← predMask(rgbs[i], point_memory)
13:      IoU ← calculateIoU(m, gt_masks[i])
14:      if IoU ≥ threshold then
15:        break
16:      end if
17:      performInt(i, m, gt_masks[i], point_memory)
18:      max_int ← max_int - 1
19:    end for
20:  end for
21:  if max_int ≥ 0 then
22:    best_ckpt ← predAllMasks(rgbs, point_memory)
23:  else
24:    break
25:  end if
26: end for
27: return best_ckpt
```

Algorithm 4 Perform Interaction.

```
1: Input: frame_idx, m, gt_m, point_memory
2: i ← frame_idx
3: tp_mask ← m ∧ gt_m
4: tn_mask ← ¬m ∧ ¬gt_m
5: fp_mask ← m ∧ ¬gt_m
6: fn_mask ← ¬m ∧ gt_m
7: pos_points ← getPosPoints(point_memory, i)
8: neg_points ← getNegPoints(point_memory, i)
9: if any neg_points in fn_mask then
10:  p ← firstIncorrectNegPoint(fn_mask, neg_points)
11:  removePointAndItsFuture(point_memory, i, p)
12: else if any pos_points in fp_mask then
13:  p ← firstIncorrectPosPoint(fp_mask, pos_points)
14:  removePointAndItsFuture(point_memory, i, p)
15: else
16:  if sum(fn_mask) > sum(fp_mask) then
17:    [x, y] ← extractPoint(fn_mask)
18:    lbl ← positive
19:  else
20:    [x, y] ← extractPoint(fp_mask)
21:    lbl ← negative
22:  end if
23:  addToFrameAndFuture(point_memory, i, lbl, x, y)
24: end if
```



(a) Successful cases for SAM-PT.



(b) Failure cases for SAM-PT.

Figure 8. Additional visualization of SAM-PT on videos from the DAVIS 2017 [30] validation subset, including (a) successful cases and (b) failure cases. Circles denote positive points and crosses denote negative points. Red symbols indicate occlusion prediction.



Figure 9. Challenging scenarios for SAM-PT on short clips from “Avatar: The Last Airbender”. These cases illustrate instances where our model struggles when faced with point tracking failures that are the result of incorrectly predicting the point at a similar-looking segment or when faced with object occlusions and disappearing objects.