Segment Anything Across Shots: A Method and Benchmark

Hengrui Hu, Kaining Ying, Henghui Ding*

Fudan University
https://henghuiding.com/SAAS/

Abstract

This work focuses on multi-shot semi-supervised video object segmentation (MVOS), which aims at segmenting the target object indicated by an initial mask throughout a video with multiple shots. The existing VOS methods mainly focus on single-shot videos and struggle with shot discontinuities, thereby limiting their real-world applicability. We propose a transition mimicking data augmentation strategy (TMA) which enables cross-shot generalization with single-shot data to alleviate the severe annotated multi-shot data sparsity, and the Segment Anything Across Shots (SAAS) model, which can detect and comprehend shot transitions effectively. To support evaluation and future study in MVOS, we introduce Cut-VOS, a new MVOS benchmark with dense mask annotations, diverse object categories, and high-frequency transitions. Extensive experiments on YouMVOS and Cut-VOS demonstrate that the proposed SAAS achieves state-of-the-art performance by effectively mimicking, understanding, and segmenting across complex transitions. The code and datasets are released at https://henghuiding.com/SAAS/.

1 Introduction

Semi-supervised video object segmentation (VOS) (Caelles et al. 2017) aims to segment and track the target object throughout a video sequence, given its mask in the first frame as a prompt. This task has received increasing attention (Ravi et al. 2024) in the research community because of its broad applicability in human–robot interaction, video editing, autonomous driving, and annotation assistance, *etc*.

Despite notable progress, existing VOS methods predominantly focus on single-shot videos, overlooking the increasing prevalence of multi-shot videos (see Figure 1 (a)) in real-world Internet content. This oversight on **multi-shot video object segmentation** (**MVOS**) has led to a widening gap between academic research and practical deployment. The current representative VOS methods, *e.g*. XMem (Cheng and Schwing 2022), DEVA (Cheng et al. 2023), Cutie (Cheng et al. 2024), and SAM2 (Ravi et al. 2024) exhibit a notable performance degradation when exposed to complex shot transitions. As shown in Figure 1 (b), SAM2-B+ suffers a 21.4% $\mathcal{J}\&\mathcal{F}$ drop on the MVOS benchmark compared to MOSE (Ding et al. 2023b), highlighting their limitations in the applications of edited videos, multi-camera systems, and high-mobility platforms.

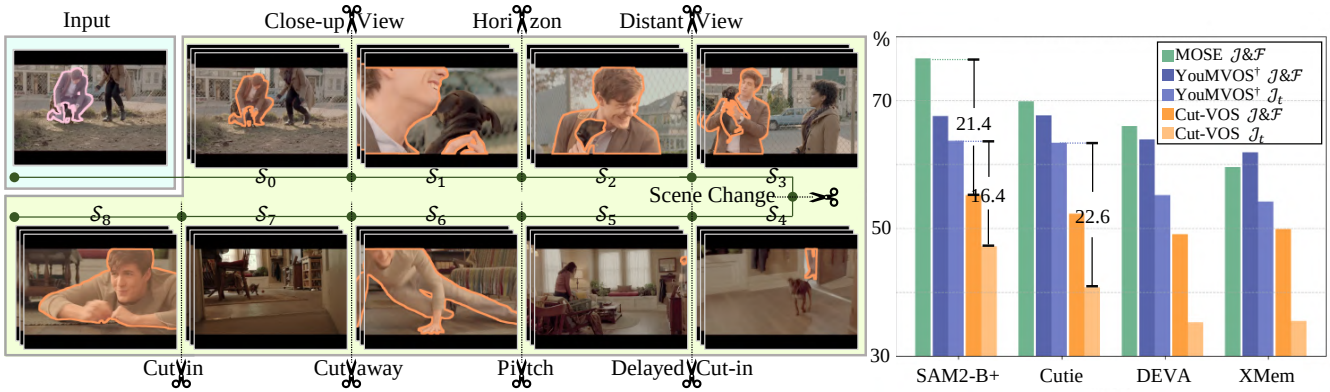
To our knowledge, YouMVOS (Wei et al. 2022) is currently the only dataset that supports MVOS. However, upon reviewing the playlists provided in their dataset, we find that the dataset falls short in fully reflecting the challenges of MVOS task. Specifically, the dataset contains only sparse shot transitions, exhibits a limited diversity of object categories with a predominant focus on humans, and lacks screening or categorization of transition types, as shown in Figure 2. Furthermore, the mask annotations of YouMVOS have not been open-sourced to date, making it unavailable for subsequent model development and training.

To address the lack of multi-shot training data, we propose the **T**ransition **M**imicking Data **A**ugmentation (**TMA**) strategy, which simulates diversiform shot transitions on single-shot datasets to enable effective multi-shot segmentation training without relying on native multi-shot annotations. Meanwhile, the deficiencies of previous methods in complex multi-shot videos, as shown in Figure 1 (b), prompt us to develop a specialized cross-shot segmentation method, **S**egment **A**nything **A**cross-**S**hot (**SAAS**), equipped with transition detection and comprehension modules. These modules jointly detect and interpret shot transitions using adjacent frames along with background context, guided by two auxiliary training objectives. Additionally, we introduce a training-free memory refinement mechanism through a local memory bank that stores fine-grained object features to enhance segmentation quality across transitions.

To fairly evaluate cross-shot segmentation performance and better reflect the complexity of real-world multi-shot videos, we introduce a new MVOS benchmark, **C**omplex M**u**lti-sho**t** **V**ideo **O**bject **S**egmentation (**Cut-VOS**), containing 10.2K instance masks for 174 unique objects in 100 videos. Compared to YouMVOS, the proposed Cut-VOS provides 1.6× higher shot transition frequency and 3× more object categories. The transition types are manually screened to ensure greater diversity and difficulty. For qualitative comparison, we build YouMVOS†[1] test split by

*Corresponding author (hhding@fudan.edu.cn).

[1]All experiments on YouMVOS in this paper are conducted on YouMVOS†, a manually annotated version constructed by us, as the original dataset does not release mask annotations.



(a) MVOS: Segment in videos with complex transitions **(b) Comparison of existing VOS models**

Figure 1: This work focuses on an underexplored task of multi-shot video object segmentation (MVOS). As shown in (a), the significant variations in object appearance, spatial location, and background across shots pose major challenges in MVOS. We introduce Cut-VOS, a challenging MVOS benchmark with high transition diversity to support this task. As shown in (b), on Cut-VOS, SAM2-B+ exhibits a 21.4% $\mathcal{J}\&\mathcal{F}$ drop compared to the challenging single-shot MOSE dataset and a 16.4% $\mathcal{J}_t$ drop compared to YouMVOS†, a sampled MVOS dataset YouMVOS annotated by our team strictly following its original protocol. The metric $\mathcal{J}_t$ specifically measures cross-shot segmentation performance, further highlighting the difficulty of Cut-VOS.

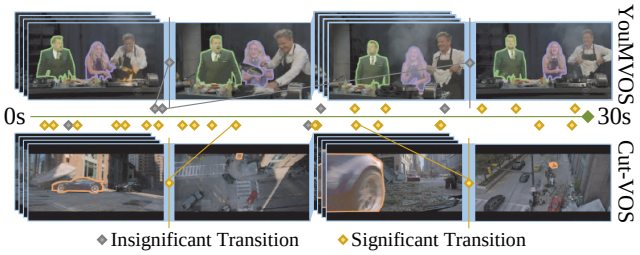


Figure 2: The comparison between YouMVOS and our proposed Cut-VOS benchmark. Cut-VOS is distinguished from YouMVOS by frequent, significant transitions and more variety in complex scenarios.

sampling and annotating 30 videos across 10 genres from the playlist, strictly following their announced protocol. Compared to YouMVOS, the models perform significantly worse on Cut-VOS, as shown in Figure 1 (b), indicating a substantial difficulty gap. Extensive experiments demonstrate that SAAS achieves consistent improvements across both YouMVOS and Cut-VOS.

Overall, the key contributions of this work are as follows:

- We introduce a new VOS training strategy, **T**ransition **M**imicking Data **A**ugmentation (TMA), to alleviate data sparsity by simulating shot transitions, thereby promoting the model's multi-shot segmentation capacity using only single-shot datasets.
- To the best of our knowledge, the proposed **SAAS** is the first semi-supervised VOS method specialized for multi-shot videos. It incorporates online transition detection, transition comprehension, and local visual cue encoding. Extensive experiments demonstrate its robustness and effectiveness in complex multi-shot scenarios.
- To facilitate future research in MVOS, we introduce **C**omplex **Mu**lti-sho**t** **V**ideo **O**bject **S**egmentation (**Cut-VOS**) dataset, which will become the first fully open-sourced MVOS benchmark with mask annotations upon publication. Cut-VOS provides diverse object categories and carefully curated transition types to evaluate cross-shot tracking performance.

2 Related Work

Video Object Segmentation. Video object segmentation (VOS) (Ding et al. 2023a, 2025c,a,b; Caelles et al. 2017; Ying, Hu, and Ding 2025; Lin, Qi, and Jia 2019; Huang et al. 2020; He et al. 2025; Liu et al. 2025) aims at tracking and segmenting the objects in a video sequence, given the mask in the first frame. Early methods (Xiao et al. 2018; Perazzi et al. 2017) are mostly fine-tuning-based. They model inter-frame correlations via fine-tuning during inference. Matching-based methods (Cheng et al. 2018; Duarte, Rawat, and Shah 2019; Duke et al. 2021) generate an object prototype embedding from the conditional frame, performing pixel-level matching to classify each pixel as foreground or background. Propagation-based methods (Han et al. 2018; Hu et al. 2018; Jabri, Owens, and Efros 2020; Wang et al. 2019) leverage the previous frames and predictions to guide the segmentation on the current frame. For better use of historical information, recent methods introduce a memory bank to compress and store previous frames. For example, XMem (Cheng and Schwing 2022) conducts multiple granularities of memories, while Cutie (Cheng et al. 2024) enriches the memory bank with object-specific queries. Most recently, SAM2 (Ravi et al. 2024) extends SAM (Kirillov et al. 2023) to the video domain, yielding a remarkable improvement via a robust memory architecture and large-scale training. However, these previous methods only focus on single-shot videos, lacking solid cross-shot tracking capacity, which leads to

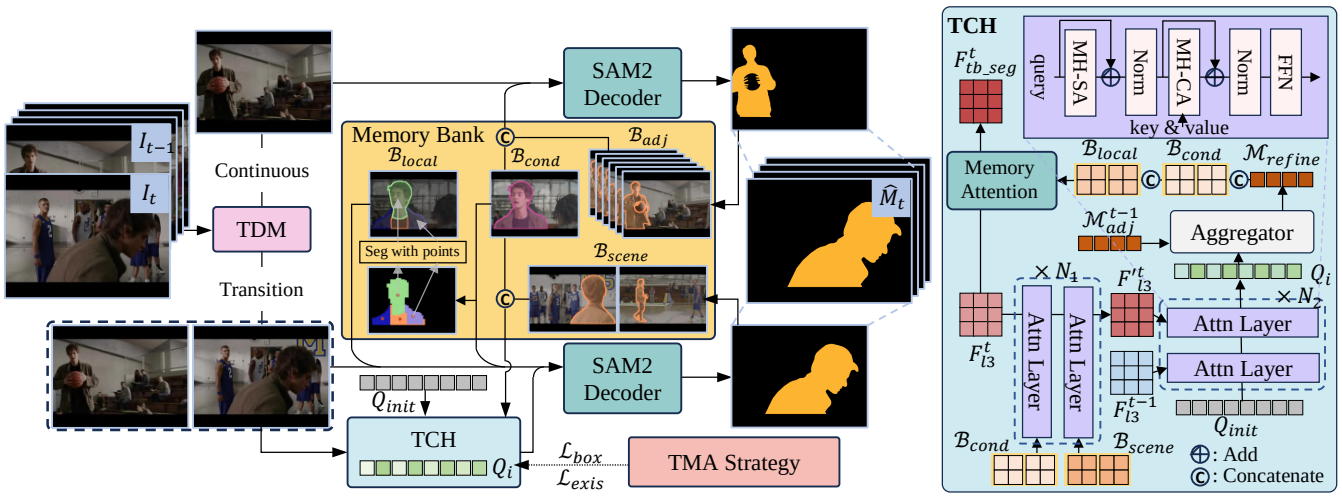


Figure 3: The overall pipeline of our proposed Segment Anything Across Shots (SAAS) method, consisting of three new components, Transition Detection Module (TDM), Transition Comprehension Module (TCH), and local memory bank $\mathcal{B}_{local}$. Transition Mimicking Augmentation (TMA) is employed to train the model by synthesizing high-quality multi-shot training samples using annotated single-shot videos.

their limited applications. This work aims to generalize VOS to multi-shot videos, bridging the gap between the current research and practical requirements.

Multi-shot Video Understanding. Multi-shot videos, which circulate on the internet at an increasingly large scale, have gradually attracted the attention of the computer vision community. Most early works (Canny 1986; Jacobs et al. 2004; Qian, Liu, and Su 2006) aim to detect the shot boundaries with manual features. With the development of deep learning, some methods (Hassanien et al. 2017; Soucek and Lokoc 2024; Bouyahi and Ayed 2020; Wang et al. 2021) adapt 3D-CNN (Ji et al. 2012) and dilated filter (Chen et al. 2017; Yu, Koltun, and Funkhouser 2017) to improve model accuracy. Meanwhile, some works collect multi-shot videos in their video captioning benchmarks (Xu et al. 2016; Krishna et al. 2017; Zhou, Xu, and Corso 2018), asking the model to generate video descriptions. Recently, Shot2Story (Han et al. 2023) and MMBench-Video (Fang et al. 2024) posed more fine-grained questions, requiring clip-wise understandings to answer. MUSES (Liu et al. 2021a) focuses on the multi-shot temporal event localization task which requests dense frame labels. However, these works still lack the exploration of pixel-level instance segmentations (Ying et al. 2022, 2023). This paper specifically targets fine-grained segmentation in multi-shot videos.

3 Methodology

3.1 Overview

Figure 3 shows an overview of our approach, which contains the proposed Transition Mimicking Data Augmentation (TMA) training strategy and a transition-aware method, Segment Anything Across Shots (SAAS), built upon the SAM2 to generalize VOS to multi-shot videos. Given a video $\mathcal{V} = \{I_t\}_{t=1}^{T}$ with T frames, and the first frame I_0 with ground truth mask M_0, SAAS firstly applies SAM2 image encoder to extract multi-level visual features $\{F^t_{li}\}_{i=1,2,3}$. At each timestep t, SAAS introduces the Transition Detection Module (TDM) to detect if a shot transition occurs and subsequently directs to diverse segmentation strategies. For the detected transitions, the following Transition Comprehension Module (TCH) further comprehends them, generates compressed transition state representation Q_i, thereby refining previous memories. To capture the local fine-grained features of objects, we also propose the local memory bank B_{local}, to partition the target and store corresponding information unsupervisedly. The conditional memories from B_{cond} and features stored in B_{local} are then concatenated to generate the features prepared to be segmented $F^t_{tb\_seg}$, used to finally predict $\hat{M}_t$ by the mask decoder. The entire architecture is trained via the TMA strategy, with two additional objectives.

3.2 Transition Mimicking Augmentation

One of the most critical challenges for MVOS is the lack of available training data. To address this issue, we propose **T**ransition **M**imicking Data **A**ugmentation (**TMA**), a new strategy which synthesizes quality-approved multi-shot training samples from annotated single-shot videos by simulating diverse transitions. TMA enables the effective MVOS training utilizing existing single-shot VOS datasets, significantly alleviating data scarcity.

We show some primary patterns involved in TMA in Figure 4. TMA maintains a conventional 8-frame continuous sampling strategy in previous VOS works with a probability $1 - p_{trans}$, otherwise performs a transition mimicking operation. Specifically, TMA conducts a single transition (as shown in (a), (b), and (d)) with a probability p_{once}, otherwise applies multiple transitions (as depicted in (c)). For each expected transition, TMA employs a well-defined framework with several control random variables to generate

Figure 4: Some visualization cases of our proposed TMA strategy. (a) Random strong transforms. (b) Single transition across different temporal segments from the same video. (c) Multiple transitions, conducting a case with *cut in* and *cut away*. (d) Single transition to another video, with random replication and gradual translations.

different transition patterns. For example, case (a) retains a continuous 8-frame sampling but applies strong transformations, including horizon flipping, random scaling, and random affine on posterior frames after the transition. This pattern simulates common view transitions, like *close-up view* or *distant view*. Case (b) cuts to a different segment from the same video, with a higher probability of sampling more further frames. The substantial temporal gap between the two clips often results in significant changes in object poses and camera viewpoints. Case (c) cuts to an unrelated video and cuts back later, like the *cut away* and *cut in* transitions. Case (d) cuts to an unrelated video while replicating the object with a random, gradual translation, simulating the *scene change* and the *delayed cut in* transitions effectively. TMA fully combines these patterns to preserve data richness while carefully avoiding ambiguous samples and anomalous noises. More details are offered in the appendix.

3.3 Transition Detection and Comprehension

Transition Detection Module. SAAS employs a lightweight **T**ransition **D**etection **M**odule (**TDM**) to detect different shot segments and occurring transitions in video sequences. Inspired by previous shot boundary detection methods (Tu et al. 2017; Soucek and Lokoc 2024), we conduct a dilated convolution pyramid (Chen et al. 2018, 2019) as TDM. At each timestep t, TDM predicts a probability score for current frame I_t, directing to different pipelines:

$$\hat{p}_{i,tr} = \text{Sigmoid}(\mathcal{F}_{\text{TDM}}(F^t, F^{t-i}_{i=1,2,...,N})), \quad (1)$$

where $\mathcal{F}_{\text{TDM}}$ indicates the main network of TDM which uses the adjacent N frames for detection. When $\hat{p}_{i,tr} < \tau_{tr}$, SAAS passes through the SAM2 segmentation pipeline (the upper part in Figure 3) directly, and only encodes the memory $\mathcal{M}_t$ into the bank $\mathcal{B}_{adj}$. Otherwise, SAAS recognizes the transition occurs and adopts a transition segmentation strategy instead (the down part). Extracted features F^t and F^{t-1}, along with few memory banks, are fed to the TCH. TCH compresses them to refine the memory tokens, followed by the segmentation head to achieve a cross-shot segmentation. Meanwhile, the memory $\mathcal{M}_t$ is encoded and stored in a special memory bank $\mathcal{B}_{scene}$ instead, used to establish a necessary spatial scene understanding in TCH.

Transition Comprehension Module. SAAS builds a **T**ransition **C**ompre**h**ension Module (**TCH**) to firstly associate stored scene information and then integrate adjacent frames to fully comprehend the occurring transition. Specifically, TCH reads out the background scene information from the banks $\mathcal{B}_{cond}$ and $\mathcal{B}_{scene}$. $\mathcal{B}_{scene}$ stores representative memories for the most closed N_s shots. These memories are used to build an entire scene understanding, subsequently integrated into F^t_{l3} via stacked attention layers, attaining F'^t_{l3}. Then, a trainable vector Q_{init} passes through the module, sufficiently interacts with the features of the previous frame and the current frame to comprehend the current transition:

$$Q^n_i = \text{Attn}(\text{Attn}(Q^{n-1}_i, F'^t_{l3}), F^{t-1}_{l3})), \quad (2)$$

where $Q^0_i = Q_{init}$, $n = \{1, 2, ..., N_2\}$. Attn represents a standard attention layer (Vaswani et al. 2017), consisting of a multi-head cross-attention, a multi-head self-attention, and a feed forward layer following previous ViT works (Dosovitskiy et al. 2020; Liu et al. 2021b) with a RoPE positional encoding (Su et al. 2024). To validate the process of transition state modeling, we incorporate two additional auxiliary objectives: presence prediction and bounding box regression. Presence prediction requires the model to predict the presence of the object on the next frame from the transition state representation Q_i, supervised by a BCE loss $\mathcal{L}_{exis}$. For the bounding box regression objective, the model learns a mapping from the previous bounding box and Q_i to the post-transition bounding box, adopting a MCE loss $\mathcal{L}_{box}$. Simple MLP stacks suffice for these objectives.

Subsequently, an attention-based aggregator is introduced to decode the transition state Q_i to refine the previous memory $\mathcal{M}^{t-1}_{adj}$. This decoding strategy ensures seamless compatibility with SAM2's well-trained segmentation head. The final refined memories are concatenated with memories from $\mathcal{B}_{cond}$ and $\mathcal{B}_{local}$ and fed to SAM2's memory attention module to prepare the features to be segmented $F^t_{tb\_seg}$.

3.4 Local Memory Bank

In a significant proportion of transitions, local object details can serve as critical segmentation cues, like the clothing of a person or the painted markings on a vehicle. Previous VOS methods struggle to actively capture and recognize such fine-grained features. Informed by such an observation, SAAS introduces a local memory bank $\mathcal{B}_{local}$ to capture and store the target's local details. Inspired by previous works (Song et al. 2019; Liang et al. 2022; Lyu, Zhong, and Zhao 2024), SAAS constructs a minimum spanning tree (MST) on the masked deepest feature map of the conditional frame $M_0 \odot F^0_{l3}$ to simultaneously preserve semantic clustering and spatial structural information. By pruning low-weight edges in the tree, the target is unsupervisedly

Dataset	#Videos	#Objects	#Masks	#Shots	Trans. Frequency	Obj. Categories	Available
YouMVOS	200	492	431.0K	13.4K	0.222/s*	4	✗
YouMVOS-test	30	78	64.6K*	2.4K	0.222/s*	4	✗
Cut-VOS (ours)	100	174	10.2K	648	0.346/s	11	✓

Table 1: The basic statistics for the Cut-VOS benchmark. * denotes the number is estimated via the corresponding description in the paper. Cut-VOS has 1.6× higher transition frequency and 3× more categories than the YouMVOS test split.

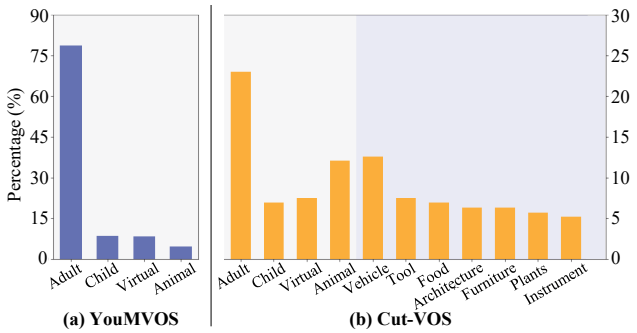


Figure 5: Comparison of object categories. Cut-VOS contains 4 categories in YouMVOS and 7 new categories.

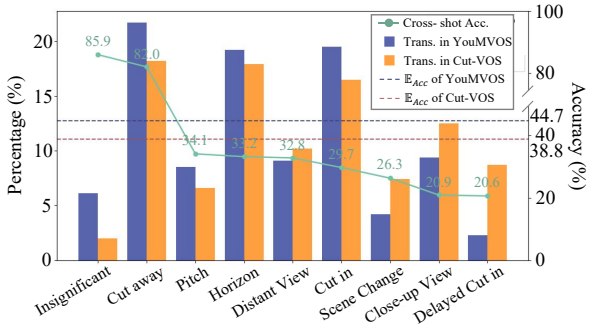


Figure 6: The average accuracies of different transition types on the SAM2-B+ model and their distribution across two benchmarks. The drop in expected accuracy shows Cut-VOS's more challenging nature.

partitioned into multiple semantically coherent sub-regions on a low-resolution map. SAAS further adopts the center point of each partition as a positive point prompt, the rest as the negative to segment these sub-regions and extract corresponding fine-grained features at a high resolution. These detailed features are compressed as complementary object pointers and preserved in the local memory bank $\mathcal{B}_{local}$, which is leveraged to guide the cross-shot segmentation when a transition is detected. Notably, we set a proportion threshold τ_p (2.5% in a common setting) to filter out too small objects, preventing over-partitioning them.

4 Cut-VOS Benchmark

4.1 Video Collection and Annotation

The new challenging multi-shot video object segmentation (MVOS) benchmark, Complex Multi-shot Video Object Segmentation (Cut-VOS), collects large amounts of high-quality multi-shot videos from mainstream community media. The videos and objects are carefully selected to ensure the data samples are unambiguous. The detailed object and transition distributions are shown in Figure 5 and Figure 6.

For mask annotation, our research team organizes and trains a cohort of highly responsible annotators and validators, establishing a robust annotation pipeline. Each annotated video undergoes a dual-review verification to ensure annotation quality assurance. For videos that are discovered with uncertain object correlations, we reconvened discussions to determine whether to keep or filter them.

4.2 Dataset Statistics

Overall, Cut-VOS contains 100 videos, 174 annotated objects, and 10.2K high-quality masks, as shown in Table 1. Cut-VOS outperforms the existing YouMVOS-test in three main aspects: 1) More videos and objects representing more diverse scenarios. 2) Carefully screened, multiple types of transitions with a 1.6 times higher frequency reaching 0.346/s, which makes the Cut-VOS more challenging. 3) 11 diversiform object categories which cover the YouMVOS as depicted in Figure 5, containing 62% actors and 38% static objects. These characteristics make the Cut-VOS benchmark more complex and better aligned with real-world scenarios.

4.3 Transition Analysis

To better analyze the latent challenges in the MVOS task, we classify all shot transitions into 9 different categories: *cut in, cut away, delayed cut in* as existence types, and *close up view, distant view, pitch transformation, horizon transformation, scene change, insignificance* as view types. In specific cases, we allow the coexistence of an existence type and a view type in one transition. Please refer to the appendix for detailed explanations and visualized examples.

We test the tracking accuracy on different transition types with the SAM2-B+ model to pinpoint existing bottlenecks. As shown in Figure 6, SAM2 performs well on *cut away* and *insignificance*, shows moderate competencies on *pitch* and *horizon* types, but drops ruinously on *delayed cut-in*, *close-up view*, and *scene change* types (lower than 27%). The observation indicates that previous methods can recognize the object disappearing, but struggle with matching targets with abrupt visual appearance and absolute position shifts. Cut-VOS filters out simple *insignificance* and long duration *cut away*, involving more difficult transitions to make the benchmark more challenging. Compared to YouMVOS, the significant decrease of $\mathbb{E}_{Acc}$ (44.7% to 38.8%) reflects the challenges brought by screened complex transitions.

Method	Venue	Param.(M)	FPS	YouMVOS				Cut-VOS			
				$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
XMem	ECCV'22	62.2	**45**	61.7	62.1	61.9	54.2	48.4	51.4	49.9	35.5
DEVA	ICCV'23	<u>61.2</u>	37	63.3	64.5	63.9	55.2	47.3	50.8	49.1	35.3
Cutie	CVPR'24	**35.0**	<u>40</u>	67.3	68.1	67.7	63.4	51.0	53.6	52.3	40.8
Cutie*	CVPR'24	**35.0**	<u>40</u>	67.9	68.8	68.4	64.7	50.0	52.7	51.4	40.0
SAM2-B+	ICLR'25	80.9	22	67.6	67.6	67.6	63.7	54.0	56.4	55.2	47.2
SAM2-L	ICLR'25	224.0	15	69.9	70.3	70.1	68.5	58.3	60.6	59.4	50.7
SAM2-B+*	ICLR'25	80.9	22	68.7	69.1	68.9	64.1	53.9	55.9	54.9	46.8
SAM2-L*	ICLR'25	224.0	15	69.7	70.7	70.2	68.4	57.6	60.3	58.9	50.4
Cutie+TMA	-	**35.0**	<u>40</u>	69.1	70.0	69.6	65.4	52.0	55.0	53.5	43.1
SAAS-B+ (Ours)	AAAI'26	92.5	21	<u>73.4</u>	<u>73.7</u>	<u>73.5</u>	<u>68.9</u>	<u>59.4</u>	<u>61.9</u>	<u>60.7</u>	<u>53.1</u>
SAAS-L (Ours)	AAAI'26	235.6	14	**74.0**	**74.4**	**74.2**	**69.6**	**60.5**	**63.6**	**62.0**	**54.0**

Table 2: Main results on YouMVOS and Cut-VOS benchmarks. * denotes the model is directly trained on the YTVOS dataset without extra data augmentation. Bold and underlined indicate the best and the second-best performance in the tested methods.

ID	$\mathcal{B}_{local}$	TMA	TCH	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
I	✗	✗	✗	55.2	47.2
II	✓	✗	✗	57.6	49.4
III	✗	✓	✗	58.0	50.7
IV	✓	✓	✗	58.8	52.0
V	✗	✓	✓	<u>60.1</u>	<u>52.8</u>
VI	✓	✓	✓	**60.7**	**53.1**

Table 3: The ablation study on different modules.

5 Experiments

Benchmark Setting. We benchmark the proposed SAAS and existing methods on Cut-VOS and YouMVOS under the semi-supervised VOS setting. Following previous works (Ding et al. 2023b; Ying et al. 2025), we compute $\mathcal{J}\&\mathcal{F}$ to quantify the region similarity and the contour accuracy of predictions. Besides, we additionally measure the cross-shot tracking capacity by computing region similarity $\mathcal{J}_t$ on post-transition frames. Given the ground truth shot set $\mathcal{S}$, for each shot $\mathcal{S}_i$ we calculate intersection over union (IoU) on the first frame I^i_{tr} and the frame where the object firstly appears I^i_{app} (defined as the first frame too if the object isn't present in the shot) separately, to accommodate different existence transitions, especially *delayed cut in*. Then $\mathcal{J}_t$ is defined as:

$$\mathcal{J}_t = \frac{1}{|\mathcal{S}|}\sum_{i\in|\mathcal{S}|}\frac{\text{IoU}(\hat{M}^i_{tr}, M^i_{tr}) + \text{IoU}(\hat{M}^i_{app}, M^i_{app})}{2}, \quad (3)$$

where M^i denotes the ground truth mask on I_i and $\hat{M}^i$ represents the predicted one. In all of the following experiments, we report both $\mathcal{J}\&\mathcal{F}$ and $\mathcal{J}_t$ as metrics.

Implementation Details. Our method is build upon SAM2 framework, with MAE-pretrained (He et al. 2022) Hiera (Bolya et al. 2023; Ryali et al. 2023) serving as image encoders. We initialize SAM2 original modules with their official weights, firstly freeze other parameters, and train our transition detection module on IACC.3 (Awad et al. 2017) and ClipShots (Tang et al. 2018) shot boundary detection datasets. In the following main training phase, we unfreeze all parameters and train the model for 30 epochs on YTVOS (Xu et al. 2018) with TMA enabled. We set the number of sampling frames as 8 for the base-plus setting and 6 for the large. We enable focal, dice, iou, and CE losses in original SAM2, along with our proposed $\mathcal{L}_{box}$ and $\mathcal{L}_{exis}$. The weights of $\mathcal{L}_{box}$ and $\mathcal{L}_{exis}$ are both set as 0.5. We employ AdamW as the optimizer, with the learning rate decaying from 5e-6 to 5e-7 during training. All experiments are conducted on 4 NVIDIA RTX-A6000 (48G) GPUs.

5.1 Main Results

As shown in Table 2, we conduct exhaustive experiments on existing VOS methods (Cheng and Schwing 2022; Cheng et al. 2023, 2024; Ravi et al. 2024) and our proposed SAAS on YouMVOS and Cut-VOS benchmarks. For SAM2 and SAAS, we test the base-plus setting and the large setting, respectively. The result shows that SAAS-B+ and SAAS-L outperform corresponding SAM2 methods and other existing VOS methods on two benchmarks across both $\mathcal{J}_t$ and $\mathcal{J}\&\mathcal{F}$ metrics, demonstrating its superiority. All reported data are calculated as the average of three runs.

From the table, we observe that training on YTVOS with TMA disabled (marked by *) results in a marginal improvement on YouMVOS (0.7% $\mathcal{J}\&\mathcal{F}$ on Cutie and 1.3% $\mathcal{J}\&\mathcal{F}$ on SAM2-B+). This strategy, however, suppressed methods' performance by 0.3% to 0.9% on Cut-VOS. The finding reveals that some videos from YouMVOS insufficiently represent MVOS difficulties, as they exhibit characteristics similar to conventional single-shot videos. In contrast, directly training on single-shot clips offers diminishing returns for Cut-VOS, which is specifically collected for MVOS.

The experimental result illustrates the effectiveness and robustness of the SAAS method. SAAS-B+ reaches 73.5% $\mathcal{J}\&\mathcal{F}$, 68.9% $\mathcal{J}_t$ on YouMVOS(vs. 67.6% and 63.7%) and 60.7% $\mathcal{J}\&\mathcal{F}$, 53.1% $\mathcal{J}_t$ on Cut-VOS(vs. 55.2% and 47.2%). Compared to SAM2-L, SAAS-L also attains consistent improvements of $\mathcal{J}\&\mathcal{F}$ (from 59.4% to 62.0%) and $\mathcal{J}_t$ (from 50.7% to 54.0%). Notably, SAAS has virtually no degradation in inference speed due to efficient designs. Cutie+TMA method, compared to Cutie and Cutie*, reaches 69.6% (vs. 68.4%) and 53.5% (vs. 52.3%) $\mathcal{J}\&\mathcal{F}$ on two benchmarks, showing great generalization of TMA strategy.

(a) **Delayed cut in** and **Position shift**

(b) **Crowded scene** and **Complex relations**

Figure 7: Qualitative comparison of some representative cases from Cut-VOS between the SAAS and the SAM2 methods. (a) shows a case with a delayed cut in transition and an abrupt position shift of target objects. (b) demonstrates SAAS's better capacity in a crowded scene with complex relations. SAAS coherently segments the target object among ten similar objects.

In the following ablation study, we offer more detailed data to further corroborate TMA and other modules' advantages.

5.2 Ablation Studies

We analyze the validation of our modules via rigorous ablation studies, shown in Table 3. The ablation studies maintain the same implementation as the main experiments, employing the base-plus setting and uniformly tested on the Cut-VOS benchmark. In Table 3, we mainly study the effectiveness of different modules. Compared to baseline model I, the local memory bank $\mathcal{B}_{local}$ and TMA (model II and III) improve $\mathcal{J}\&\mathcal{F}$ by 2.4% and 2.8% respectively, while TMA plus TCH (V) achieves a 4.9% $\mathcal{J}\&\mathcal{F}$ increase. For more ablation studies and hyperparameters analysis in detail, please refer to the appendix.

5.3 Qualitative Results

Figure 7 presents several representative visualized examples and corresponding segmentation results of SAM2 and SAAS models. Case (a) shows a delayed cut in transition, one of the most difficult types, and a classical abrupt position shift of the target object, with a similar appearance distractor appearing at the same position. SAM2 misses the target man (orange) when he reoccurs in shot 2, and incorrectly segments one another man with the same clothing (green) in shot 3, whereas our method successfully segments them. In case (b), we highlight a crowded scene with complex relations between multiple similar objects. SAM2 model struggles to match different instances correctly, leading to flickering predictions. In contrast, by effectively capturing detail cues and establishing scene understanding, SAAS predicts high-quality masks for the object of interest consistently. These examples demonstrate the superiority of our approach in complex multi-shot videos. A few more qualitative analyses are involved in the appendix.

6 Conclusion

We introduce **TMA**, a new training strategy that mitigates MVOS data sparsity by mimicking different transitions on single-shot datasets, and **SAAS**, a new MVOS method performing robust multi-shot segmentation capacity on complex edited videos. Meanwhile, we present a complex multi-shot benchmark, **Cut-VOS**, enabling evaluation and facilitating future research in MVOS. Extensive experiments demonstrate that our proposed strategy and method achieve state-of-the-art performance on MVOS benchmarks.

Limitations. Our method still struggles with extreme appearance changes of the target. For example, the same person with different clothing and hairstyles. The proposed TMA can't simulate this type effectively, and captured local cues may not help. This reflects one of the key challenges for MVOS: the model is required to both match unlike targets and distinguish similar distractors, demanding reducing the reliance on pure visual feature matching and requiring a stronger reasoning ability, which is to be further explored.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62472104.

References

Awad, G.; Butt, A. A.; Fiscus, J.; Joy, D.; Delgado, A.; Mcclinton, W.; Michel, M.; Smeaton, A. F.; Graham, Y.; Kraaij, W.; et al. 2017. Trecvid 2017: evaluating ad-hoc and instance video search, events detection, video captioning, and hyperlinking. In *TRECVID 2017*. NIST.

Bolya, D.; Ryali, C.; Hoffman, J.; and Feichtenhofer, C. 2023. Window attention is bugged: How not to interpolate position embeddings. arXiv:2311.05613.

Bouyahi, M.; and Ayed, Y. B. 2020. Video scenes segmentation based on multimodal genre prediction. *Procedia Computer Science*, 176: 10–21.

Caelles, S.; Maninis, K.-K.; Pont-Tuset, J.; Leal-Taixé, L.; Cremers, D.; and Van Gool, L. 2017. One-shot video object segmentation. In *CVPR 2017*, 221–230. IEEE.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6): 679–698.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2019. Rethinking atrous convolution for semantic image segmentation. arXiv 2017. arXiv:1706.05587.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV 2018*, 801–818. Springer.

Cheng, H. K.; Oh, S. W.; Price, B.; Lee, J.-Y.; and Schwing, A. 2024. Putting the object back into video object segmentation. In *CVPR 2024*, 3151–3161. IEEE.

Cheng, H. K.; Oh, S. W.; Price, B.; Schwing, A.; and Lee, J.-Y. 2023. Tracking anything with decoupled video segmentation. In *ICCV 2023*, 1316–1326. IEEE.

Cheng, H. K.; and Schwing, A. G. 2022. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV 2022*, 640–658. Springer.

Cheng, J.; Tsai, Y.-H.; Hung, W.-C.; Wang, S.; and Yang, M.-H. 2018. Fast and accurate online video object segmentation via tracking parts. In *CVPR 2018*, 7415–7424. IEEE.

Ding, H.; Liu, C.; He, S.; Jiang, X.; and Loy, C. C. 2023a. MeViS: A large-scale benchmark for video segmentation with motion expressions. In *ICCV 2023*, 2694–2703.

Ding, H.; Liu, C.; He, S.; Jiang, X.; Torr, P. H.; and Bai, S. 2023b. MOSE: A new dataset for video object segmentation in complex scenes. In *ICCV 2023*, 20224–20234. IEEE.

Ding, H.; Liu, C.; He, S.; Ying, K.; Jiang, X.; Loy, C. C.; and Jiang, Y.-G. 2025a. MeViS: A multi-modal dataset for referring motion expression video segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12): 11400–11416.

Ding, H.; Tang, S.; He, S.; Liu, C.; Wu, Z.; and Jiang, Y.-G. 2025b. Multimodal referring segmentation: A survey. *arXiv preprint arXiv:2508.00265*.

Ding, H.; Ying, K.; Liu, C.; He, S.; Jiang, X.; Jiang, Y.-G.; Torr, P. H.; and Bai, S. 2025c. MOSEv2: A more challenging dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2508.05630*.

Ding, S.; Qian, R.; Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Guo, Y.; Lin, D.; and Wang, J. 2024. Sam2long: Enhancing sam 2 for long video segmentation with a training-free memory tree. arXiv:2410.16268.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.

Duarte, K.; Rawat, Y. S.; and Shah, M. 2019. Capsulevos: Semi-supervised video object segmentation using capsule routing. In *ICCV 2019*, 8480–8489. IEEE.

Duke, B.; Ahmed, A.; Wolf, C.; Aarabi, P.; and Taylor, G. W. 2021. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR 2021*, 5912–5921. IEEE.

Fang, X.; Mao, K.; Duan, H.; Zhao, X.; Li, Y.; Lin, D.; and Chen, K. 2024. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *Advances in Neural Information Processing Systems*, 37: 89098–89124.

Han, J.; Yang, L.; Zhang, D.; Chang, X.; and Liang, X. 2018. Reinforcement cutting-agent learning for video object segmentation. In *CVPR 2018*, 9080–9089. IEEE.

Han, M.; Yang, L.; Chang, X.; and Wang, H. 2023. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. arXiv:2312.10300.

Hassanien, A.; Elgharib, M.; Selim, A.; Bae, S.-H.; Hefeeda, M.; and Matusik, W. 2017. Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. *arXiv preprint arXiv:1705.03281*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *CVPR 2022*, 16000–16009. IEEE.

He, S.; Ji, P.; Yang, Y.; Wang, C.; Ji, J.; Wang, Y.; and Ding, H. 2025. A survey on 3d gaussian splatting applications: Segmentation, editing, and generation. *arXiv preprint arXiv:2508.09977*.

Hong, L.; Liu, Z.; Chen, W.; Tan, C.; Feng, Y.; Zhou, X.; Guo, P.; Li, J.; Chen, Z.; Gao, S.; et al. 2024. LVOS: A Benchmark for Large-scale Long-term Video Object Segmentation. arXiv:2404.19326.

Hu, P.; Wang, G.; Kong, X.; Kuen, J.; and Tan, Y.-P. 2018. Motion-guided cascaded refinement network for video object segmentation. In *CVPR 2018*, 1400–1409. IEEE.

Huang, X.; Xu, J.; Tai, Y.-W.; and Tang, C.-K. 2020. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *CVPR 2020*, 8879–8889. IEEE.

Jabri, A.; Owens, A.; and Efros, A. 2020. Space-time correspondence as a contrastive random walk. *Advances in Neural Information Processing Systems*, 33: 19545–19560.

Jacobs, A.; Miene, A.; Ioannidis, G. T.; and Herzog, O. 2004. Automatic Shot Boundary Detection Combining Color, Edge, and Motion Features of Adjacent Frames. In *TRECVID 2004*, volume 2004, 197–206. NIST.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 221–231.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *ICCV 2023*, 4015–4026. IEEE.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *ICCV 2017*, 706–715. IEEE.

Liang, Z.; Wang, T.; Zhang, X.; Sun, J.; and Shen, J. 2022. Tree energy loss: Towards sparsely annotated semantic segmentation. In *CVPR 2022*, 16907–16916. IEEE.

Lin, H.; Qi, X.; and Jia, J. 2019. Agss-vos: Attention guided single-shot video object segmentation. In *ICCV 2019*, 3949–3957. IEEE.

Liu, C.; Ding, H.; Ying, K.; Hong, L.; Xu, N.; Yang, L.; Fan, Y.; Gao, M.; Chen, J.; Miao, Y.; et al. 2025. LSVOS 2025 Challenge Report: Recent Advances in Complex Video Object Segmentation. *arXiv preprint arXiv:2510.11063*.

Liu, X.; Hu, Y.; Bai, S.; Ding, F.; Bai, X.; and Torr, P. H. 2021a. Multi-shot temporal event localization: a benchmark. In *CVPR 2021*, 12596–12606. IEEE.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV 2021*, 10012–10022. IEEE.

Lyu, H.; Zhong, T.; and Zhao, S. 2024. Gtms: A gradient-driven tree-guided mask-free referring image segmentation method. In *ECCV 2024*, 288–304. Springer.

Perazzi, F.; Khoreva, A.; Benenson, R.; Schiele, B.; and Sorkine-Hornung, A. 2017. Learning video object segmentation from static images. In *CVPR 2017*, 2663–2672. IEEE.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 davis challenge on video object segmentation.

Qian, X.; Liu, G.; and Su, R. 2006. Effective fades and flashlight detection based on accumulating histogram difference. *IEEE Trans. Circuit Syst. Video Technol.*, 16(10): 1245–1258.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. arXiv:2408.00714.

Ryali, C.; Hu, Y.-T.; Bolya, D.; Wei, C.; Fan, H.; Huang, P.-Y.; Aggarwal, V.; Chowdhury, A.; Poursaeed, O.; Hoffman, J.; et al. 2023. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *ICML 2023*, 29441–29454. PMLR.

Song, L.; Li, Y.; Li, Z.; Yu, G.; Sun, H.; Sun, J.; and Zheng, N. 2019. Learnable tree filter for structure-preserving feature transform. *Advances in Neural Information Processing Systems*, 32.

Soucek, T.; and Lokoc, J. 2024. Transnet v2: An effective deep network architecture for fast shot transition detection. In *ACMMM 2024*, 11218–11221. ACM.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Tang, S.; Feng, L.; Kuang, Z.; Chen, Y.; and Zhang, W. 2018. Fast video shot transition localization with deep structured models. In *ACCV 2018*, 577–592. Springer.

Tu, C.; Zhang, Z.; Liu, Z.; and Sun, M. 2017. TransNet: Translation-Based Network Representation Learning for Social Relation Extraction. In *IJCAI 2017*, 2864–2870. IJCAI Inc.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Videnovic, J.; Lukezic, A.; and Kristan, M. 2025. A distractor-aware memory for visual object tracking with sam2. In *CVPR 2025*, 24255–24264. IEEE.

Wang, T.; Feng, N.; Yu, J.; He, Y.; Hu, Y.; and Chen, Y.-P. P. 2021. Shot boundary detection through multi-stage deep convolution neural network. In *MMM 2021*, 456–468. Springer.

Wang, Z.; Xu, J.; Liu, L.; Zhu, F.; and Shao, L. 2019. Ranet: Ranking attention network for fast video object segmentation. In *ICCV 2019*, 3978–3987. IEEE.

Wei, D.; Kharbanda, S.; Arora, S.; Roy, R.; Jain, N.; Palrecha, A.; Shah, T.; Mathur, S.; Mathur, R.; Kemkar, A.; et al. 2022. Youmvos: an actor-centric multi-shot video object segmentation dataset. In *CVPR 2022*, 21044–21053. IEEE.

Xiao, H.; Feng, J.; Lin, G.; Liu, Y.; and Zhang, M. 2018. Monet: Deep motion exploitation for video object segmentation. In *CVPR 2018*. IEEE.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR 2016*, 5288–5296. IEEE.

Xu, N.; Yang, L.; Fan, Y.; Yue, D.; Liang, Y.; Yang, J.; and Huang, T. 2018. Youtube-vos: A large-scale video object segmentation benchmark. arXiv:1809.03327.

Yang, C.-Y.; Huang, H.-W.; Chai, W.; Jiang, Z.; and Hwang, J.-N. 2024. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. arXiv:2411.11922.

Ying, K.; Ding, H.; Jie, G.; and Jiang, Y.-G. 2025. Towards omnimodal expressions and reasoning in referring audio-visual segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22575–22585.

Ying, K.; Hu, H.; and Ding, H. 2025. MOVE: Motion-guided few-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11632–11642.

Ying, K.; Wang, Z.; Bai, C.; and Zhou, P. 2022. Isda: Position-aware instance segmentation with deformable attention. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2619–2623. IEEE.

Ying, K.; Zhong, Q.; Mao, W.; Wang, Z.; Chen, H.; Wu, L. Y.; Liu, Y.; Fan, C.; Zhuge, Y.; and Shen, C. 2023. Ctvis: Consistent training for online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 899–908.

Yu, F.; Koltun, V.; and Funkhouser, T. 2017. Dilated residual networks. In *CVPR 2017*, 472–480. IEEE.

Zhou, L.; Xu, C.; and Corso, J. 2018. Towards automatic learning of procedures from web instructional videos. In *AAAI*. AAAI Press.

Appendix

A Cut-VOS Benchmark

A.1 Statistics

The proposed benchmark, Cut-VOS, contains 100 high-quality videos, 174 objects of 11 different categories, 7965 frames, and 10.2K valid masks. Counting the number of shots for each target, Cut-VOS has 1131 shots in total. Counting the number of shots for each video, Cut-VOS has 648 shots, with an average of 6.5 shots per video. With an average length of 15.9 seconds of each video, the transition frequency is calculated as 0.346/s.

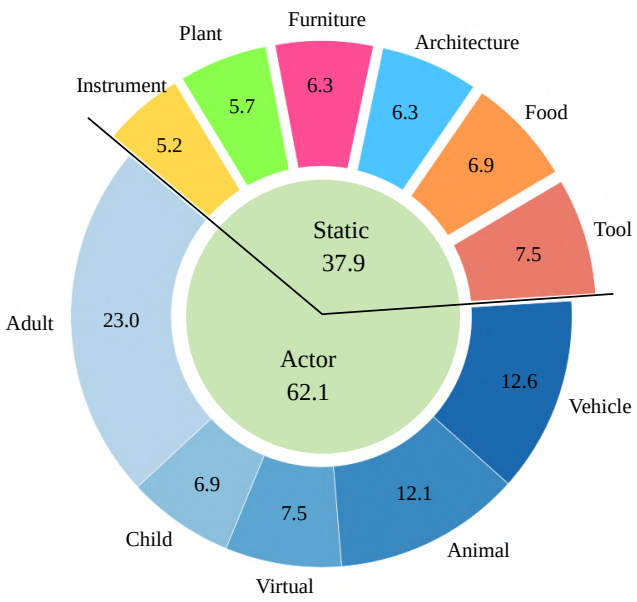


Figure 8: The object category distribution in Cut-VOS.

The distribution of object categories across 11 coarse categories is shown in Figure 8. Cut-VOS benchmark has 62.1% actors and 37.9% static objects overall.

egg	flag	chair	flower	warplane
gun	lion	fruit	tomato	sculpture
cup	sofa	horse	bottle	instrument
dog	meat	knife	insect	race car
car	ball	motor	laptop	tree branch
meat	bread	pizza	cheetah	other plants
ship	adult	table	penguin	remote control
stem	child	snake	building	other tool

Table 4: All fine-grained categories involved in Cut-VOS.

In a finer classification manner, Cut-VOS contains 40 object categories. We show all contained categories in Table 4. Except for adult and child, which account for about 30% of the total, the remaining categories show a relatively uniform distribution (3.2 instances per category on average).

A.2 Transition Analysis

In this section, we introduce different types of transitions classified by us in detail. Figure 9 shows some representative cases picked from Cut-VOS. Except for the *cut in* case we additionally show the conditional frame; we mainly show two preceding frames before the transition and two subsequent frames after the transition. With these visualization cases, we further explain the definition of these types:

Figure 9: Cases of different types of transitions.

1. *Cut away*. The target is present in the previous frame, but disappears in the next frame.
2. *Cut in*. The target isn't present in the previous frame, but reoccurs in the next frame.
3. *Delayed cut in*. The target may or may not be present in the previous frame. The target isn't present in the first frame of the next shot, but reoccurs in the shot with the movements of the camera view or the object itself.
4. *Horizon Transformation*. The camera rotates horizontally to reveal different aspects of the target.
5. *Pitch Transformation*. The camera's pitch angle changes to reveal the top or bottom surface of the target.
6. *Close-up view*. The viewpoint zooms in on the target, typically changing from the entire object to a partial.
7. *Distant View*. The viewpoint zooms out on the target, usually changing from a part of the object to the whole.

Algorithm 1: Transition Mimicking Data Augmentation

```
def Affine_M(frame, mask):
    affine = RandomAffine(
        ... ) # a moderate affine
    return affine(frame), affine(mask)
def Affine_S(frame, mask):
    affine = RandomAffine(
        ... ) # a strong affine
    return affine(frame), affine(mask)
def Hflip(frame, mask):
    return frame[:,:,::-1], mask[:,:,::-1]
def CopyFg(s_frame, frame, mask):
    s_frame[mask>0] = frame[mask>0]
    return s_frame
def GTranslation(frame, mask, i, t):
    tl = RandomTranslation(1, 1)
    tl = trans * (t - i) / t
    return tl(frame), tl(mask)
def TMA(frames, masks, video_lists):
    # input format:[T,H,W,3],[T,H,W],VOSDataset
    if rand() > p_trans:
        return frames, masks
    if_once = rand() < p_once
    s = T // 2 if if_once else T // 3
    e = T if if_once else T // 3 * 2 + 1
    if rand() < p_cut:
        if_same_video = rand() < p_same
        if_copy = rand() < p_copy
        if if_same_video:
            aug_frames, aug_masks =
                current_video.get_sample()
        # More possible to sample further segments
        else:
            aug_frames = video_lists.get_sample()
            aug_masks = np.zeros((T, H, W))
        aug_frames, aug_masks = Affine_M(
            aug_frames, aug_masks)
        for i in range(s, e):
            if not if_same_video and if_copy:
                frames[i], masks[i] = GTranslation(
                    frames[i], masks[i], i-s, T//2)
                aug_frames[i] = CopyFg(
                    aug_frames[i], frames[i], masks[i])
                aug_masks[i] = masks[i].copy()
            frames[i] = aug_frames[i]
            masks[i] = aug_masks[i]
    else:
        for i in range(s, e):
            frames[i], masks[i] = Affine_S(
                frames[i], masks[i])
    if rand() < p_hflip:
        for i in range(s, e):
            frames[i], masks[i] = Hflip(
                frames[i], masks[i])
    return frames, masks
```

8. *Scene change*. Abrupt changes in time or space. The background would be completely different, sometimes with great changes in targets' appearance as well.
9. *Insignificance*. A cut between two similar shots (like a short frame decimation).

All types of transitions can be further classified into two categories: *Cut away, cut in, and delayed cut in* as presence transitions, and the rest as view transitions. We allow the coexistence of one presence transition and one view transition when the target object reoccurs in the video, since the model is required to address the challenges arising from both types of transitions. It's difficult to conclusively determine which factor plays the predominant role in the model's failure. Also, multiple different view transitions may occur simultaneously (a 180-degree *horizon transformation* and a 30-degree *pitch transformation*, for example). We only marked the predominant one type (*horizon*) for these transitions.

We also further explain the accuracy computation experiments based on the definitions of these transition types, mentioned in Section 4 of the main paper. The experiment is conducted on the SAM2-B+ model. For a given transition, we mark it as a correct segmentation if the IoU between the predicted masks and ground truth masks exceeds 0.5 in both the preceding frames before the transition and the subsequent frames after the transition (for delayed cut in, add the first frame where the target reoccurs as well).

ID	Hyperparameters						$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
	v_1	v_2	v_3	v_4	v_5	v_6		
I	0.80	0.25	0.70	0.30	0.75	0.70	58.4	50.1
II	0.75	0.30	0.70	0.40	0.75	0.60	58.8	50.2
III	0.40	0.75	0.50	0.70	0.30	0.25	58.9	48.3
IV	0.50	0.70	0.60	0.60	0.45	0.30	58.6	48.9
V	0.50	0.50	0.60	0.50	0.50	0.45	<u>59.2</u>	<u>50.7</u>
VI	0.60	0.60	0.50	0.40	0.75	0.55	**59.5**	**51.6**

Table 5: Experiments on the impact of different probability settings in TMA. The shortened variables v_1 to v_6 represent p_{trans}, p_{once}, p_{cut}, p_{same}, p_{copy}, p_{hflip} respectively.

ID	Aggregator	Q_i	$\mathcal{B}_{scene}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
I	Linear	✗	✓	59.2	50.1
II	Convolution	✗	✓	58.9	50.2
III	Convolution	✓	✓	<u>59.9</u>	50.9
IV	Cross-attn	✗	✗	58.2	51.2
V	Cross-attn	✓	✗	59.6	51.4
VI	Cross-attn	✗	✓	59.8	<u>51.7</u>
VII	Cross-attn	✓	✓	**60.6**	**52.9**

Table 6: The ablation study on the design of TCH.

A.3 Licences

We will release our work under permissive open licences, including the Cut-VOS benchmark (CC by 4.0), the SAAS code(including pre-processing and evaluation codes), and main checkpoints(Apache 2.0).

1. *Do you have reason to believe the annotations in this dataset may change over time? Do you plan to update your dataset?* No.
2. *Are there any conditions or definitions that, if changed, could impact the utility of your dataset?* No.
3. *Will you attempt to track, impose limitations on, or otherwise influence how your dataset is used? If so, how?* The Cut-VOS benchmark would be released under a permissive CC by 4.0 licence.
4. *Were annotators informed about how the data is externalized? If changes to the dataset are made, will they be informed?* No.
5. *Is there a process by which annotators can later choose to withdraw their data from the dataset? If so, please detail.* No.

ID	N_{enc}	N_{dec}	#Parameters(M)	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
I	1	1	91.4	58.1	49.8
II	1	2	91.4	58.4	50.1
III	2	1	92.4	59.0	50.6
IV	2	2	92.5	**59.4**	**51.2**
V	3	3	93.6	59.1	50.6
VI	4	4	94.8	59.3	50.8

Table 7: The impact of the number of attention layers.

ID	$\mathcal{L}_{exis}$	$\mathcal{L}_{box}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
I	0	0	57.3	60.7	59.0	49.0
II	0.1	0.1	57.8	60.8	59.3	50.4
III	0.5	0	57.7	60.8	59.2	50.2
IV	0	0.5	58.1	60.5	59.3	49.8
V	0.5	0.5	**58.3**	**61.0**	**59.6**	**51.2**
VI	1	1	58.3	60.9	59.6	50.8
VII	4	4	57.9	60.9	59.4	50.5

Table 8: Comparison of model performance under different loss weights of $\mathcal{L}_{exis}$ and $\mathcal{L}_{box}$.

ID	Groups	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
I	1	58.0	60.7	59.3	50.7
II	2	58.3	61.1	59.7	51.2
III	4	**58.6**	**61.4**	**60.0**	**51.3**
IV	6	57.9	60.7	59.3	50.6
V	8	57.9	60.7	59.3	50.6
VI	10	57.8	60.6	59.2	50.6

Table 9: The experimental results on the same checkpoint with different numbers of partition groups in the LMM.

ID	Setting	τ_p(%)	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
I	No Limit	0.0	58.6	61.3	59.9	**51.5**
II	Loose	1.0	58.4	61.1	59.8	51.2
III	Moderate	2.5	**58.6**	**61.4**	**60.0**	51.3
IV	Strict	7.5	58.1	60.9	59.5	51.3
V	Extreme	15.0	58.0	60.8	59.4	51.0

Table 10: SAAS's performance under different settings of τ_p, tested on the same checkpoint.

B Code

In this section, we mainly discuss our proposed data augmentation strategy and multi-shot segmentation method in detail. The main content includes: further explanatory notes on selected algorithms, comprehensive comparative experiments on hyperparameter configurations, sensitivity analysis of relevant parameters, *etc*. Unless otherwise specified, all experiments in this section are conducted on Cut-VOS.

B.1 Transition Mimicking Data Augmentation

Firstly, we further describe the TMA algorithm in detail, including the introduction of all involved random variables used to control different transition patterns and how these patterns are actually generated. Overall, the workflow of the TMA algorithm is shown in Algorithm 1.

The probability options are marked in red in the algorithm, involving p_{trans}, p_{once}, p_{cut}, p_{sam}, p_{copy}, and p_{hflip}. They work together to control the augmented data distribution. To better set their values, we conduct an exhaustive comparison of experiments on the TMA with different probability options. We train SAAS with 6 different settings for 20 epochs. The final results are reported in Table 5. Among these settings, settings I and II represent more aggressive augmentation strategies, tending to perform more numerous and complex augmentations with a higher likelihood to generate combined transformations. In contrast, settings III and IV adopt a more conservative strategy with lower frequency and mild augmentations. Settings V and VI are moderate, generating different transitions in a more balanced manner.

The experimental result shows that the moderate TMA settings are most beneficial to train MVOS models, reaching higher $\mathcal{J}\&\mathcal{F}$ and $\mathcal{J}_t$ on Cut-VOS. The observation offers reliable guidance for our final decisions on hyperparameters.

B.2 Transition Comprehension Module

We first study the different designs of the aggregator involved in TCH and how $\mathcal{B}_{scene}$ and Q_i influence the model's performance. The results are shown in Table 6. A Comparison between methods I, II, and VI reveals the best performance of the cross-attention aggregator (59.8% vs. 59.2% and 58.9%). More experiments based on cross-attention aggregator (IV, V, VI, VII) further clarify the roles of Q_i and $\mathcal{B}_{scene}$ in TCH.

To explore the specific architecture of the encoder to extract the transition state and the decoder to utilize Q_i to refine previous memories, we conduct more experiments on it. Based on a common multi-head vision transformer layer (Dosovitskiy et al. 2020; Liu et al. 2021b) with a RoPE positional encoding (Su et al. 2024), as reported in the main paper, we further adjust the number of stacked transformer layers (N_{enc} and N_{dec}) to study the best design of them. For each experiment, we keep the other hyperparameters the same and retrain the model on the same hardware for 20 epochs. The final results are reported in Table 7.The result reveals that an insufficient number of layers limits the model's expressive power, while excessive layers may introduce difficulties in training and convergence. Experimental results demonstrate that setting both the N_{enc} and N_{dec} to 2 yields an optimal performance. This model architecture is adopted in the other experiments.

We also conduct experiments to validate the effectiveness of two auxiliary objectives, which complement the ablation study presented in the main paper. We adjust the weights of $\mathcal{L}_{exis}$ and $\mathcal{L}_{box}$ for different settings and train each model for 20 epochs. The experiments covered the process of adjusting weights from small to large and different relative proportions to explore the optimal values. The results are shown in Table 8. When the weights of $\mathcal{L}_{exis}$ and $\mathcal{L}_{box}$ are both set as 0.5, the model achieves a best performance of 59.6% $\mathcal{J}\&\mathcal{F}$ and 51.2% $\mathcal{J}_t$. Compared with not using auxiliary objectives (I), it achieved a $\mathcal{J}_t$ improvement of approximately 2.2%, reflecting the effectiveness of auxiliary objectives. As their weights increase (VI, VII), we observed a gradual decline in performance, possibly due to the impact on the primary mask objective during learning. Therefore, we keep them both as 0.5 in the main experiments.

B.3 Training-free Memory Refinements

The memory refinements introduced in the paper include a local memory bank to store local, fine-grained features and a scene memory bank used in TCH to build a basic understanding of the scene. These training-free refinements consistently enhance the performance, as shown in the ablation study in the main paper. In

Method	Venue	FPS	YouMVOS†				Cut-VOS			
			$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
SAM2 (Ravi et al. 2024)	ICLR'25	**22**	68.6	69.0	68.8	64.2	54.1	56.1	55.1	46.9
SAMURAI (Yang et al. 2024)	Preprint'24	18	68.2	68.8	68.5	62.5	55.0	58.2	56.6	47.7
SAM2LONG (Ding et al. 2024)	ICCV'25	11	70.0	70.7	70.4	65.7	55.0	57.9	56.5	48.5
DAM4SAM (Videnovic et al. 2025)	CVPR'25	17	70.5	71.6	71.1	65.9	56.2	58.9	57.6	48.6
SAAS (Ours)	AAAI'26	21	**73.4**	**73.7**	**73.5**	**68.9**	**59.4**	**61.9**	**60.7**	**53.1**

Table 11: Experiment results on more vision methods, including SAMURAI, SAM2LONGA, and DAM4SAM. They are all built upon the SAM2 model in a training-free manner.

Method	DAVIS2017-val			MOSE			LVOSv2					YoutubeVOS
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{G}$
SAM2-B+	86.8	93.1	90.0	69.4	77.4	73.4	80.1	87.0	78.3	85.1	82.6	88.2
SAAS-B+	87.3	92.8	90.0	69.0	76.9	73.0	79.3	86.1	80.3	87.4	83.3	88.6
SAM2-L	86.9	**93.4**	90.2	**70.3**	**78.2**	**74.2**	**80.8**	**87.4**	80.2	87.3	83.9	88.8
SAAS-L	**87.5**	93.1	**90.3**	70.1	77.2	73.6	80.5	87.3	**81.6**	**89.0**	**84.6**	**89.2**

Table 12: Experiment results on previous VOS datasets. We test the SAM2 and SAAS methods in a zero-shot setting.

Method	Oracle	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_t$
SAM2	✗	54.0	56.4	55.2	47.2
SAM2	✓	91.1	95.1	93.1	97.0
SAAS	✗	59.7	62.2	60.7	53.1
SAAS	✓	91.3	95.4	93.3	97.2

Table 13: Cross-shot oracle experiment on Cut-VOS.

this section, we mainly focus on the technical details in the local memory bank $\mathcal{B}_{local}$.

We first explore the impact of the number of partition groups in the process of local memory bank extraction. Intuitively, too few groups may fail to segment the object into independent sub-regions, while too many groups could compromise semantic properties. We change the number of groups gradually, from 1 to 10, and benchmark the model performance on **the same** retrained checkpoint for a fair comparison. The results are shown in Table 9.

As the number of groups increases, the model performance first improves, then declines, peaking at 4 groups (60.0% $\mathcal{J}\&\mathcal{F}$ and 51.3% $\mathcal{J}_t$), confirming the hypothesis that a moderate number of groups exerts a positive influence on model performance. Thus, the partition groups are set as 4 in other experiments. Another variable to be set is the proportion threshold τ_p, which we use to control the construction of LMM to prevent over-partitioning on small targets. Thereby, its value may influence the quality of captured fine-grained features. We conduct a comparison experiment to study this via testing the SAAS segmentation results on the Cut-VOS under 5 settings with different values of τ_p: no limitation (0%), loose (1%), moderate (2.5%), strict (7.5%), and extreme strict (15%). The results are illustrated in Table 10.

Overall, the model seems not really sensitive to τ_p of small values ($\leq$ 2.5%). However, when τ_p gets larger, our proposed local memory bank no longer fulfills its intended function, leading to a plainly visible degradation. The result indicates that further partitioning the small objects into pixel pieces won't lead to really downside. But we are still willing to set it as a moderate value, *e.g.* 2.5%, considering the robustness and reasonability.

B.4 SAAS Hyperparameters Selection

Drawing from the comprehensive results of the above comparison experiments, we establish a well-defined hyperparameter configuration for the SAAS method, supporting the main results presented in the paper. We set p_{trans}, p_{once}, p_{cut}, p_{same}, p_{copy}, p_{hflip} as 0.60, 0.60, 0.70, 0.40, 0.75, 0.55 in TMA respectively, adopting a relatively balanced strategy. For the transition comprehension module, we set the number of both encoder layers and decoder layers as 2. The structure of the aggregator is decided as cross-attention layers, following the result of ablation studies in the main paper. Two new auxiliary objectives are both enabled, with a weight of 0.05. The local memory bank $\mathcal{M}_{local}$ is constructed with local detail features from 4 sub-regions, only if the proportion of the ground truth mask exceeds 2.5% in the conditional frame. The entire configuration works well in exhaustive experiments, outperforming the baseline significantly and achieving 74.2% $\mathcal{J}\&\mathcal{F}$ on YouMVOS and 62.5% $\mathcal{J}\&\mathcal{F}$ on Cut-VOS. We believe this content will help reproduce our method on other devices.

C Experiments

C.1 Comparison to More Vision Methods

We benchmark some most recent approaches that are building upon SAM2 as well, *e.g.* SAMURAI (Yang et al. 2024), SAM2LONG (Ding et al. 2024), and DAM4SAM (Videnovic, Lukezic, and Kristan 2025). Since these methods are designed in a training-free manner, we train another SAM2-B+ checkpoint under the same experiment setting and benchmark these methods with the only checkpoint for a fair comparison. The results are shown in Table 11. The results reveal that despite marginal improvements brought by these methods, they still struggle with complex multi-shot videos. Also, they significantly lag behind our proposed SAAS, which is specifically introduced for MVOS, across both $\mathcal{J}\&\mathcal{F}$ and $\mathcal{J}_t$ metrics. The second-best method, DAM4SAM, improves $\mathcal{J}\&\mathcal{F}$ from 55.1% to 57.6% compared to the baseline, but shows a 3.1% $\mathcal{J}\&\mathcal{F}$ and 4.5% $\mathcal{J}_t$ gap to SAAS. The experiments highlight the SAAS's superiority on MVOS compared to existing vision methods.

Figure 10: More qualitative experiment results, including a video containing many shot transitions of different types, and a case to segment static objects. SAAS performs robust segmentation capacity on these complex videos.

C.2 Performance on Previous VOS Datasets

We also report the performance of SAAS on some previous single-shot VOS datasets, including DAVIS2017 (Pont-Tuset et al. 2017), MOSE (Ding et al. 2023b), LVOSv2 (Hong et al. 2024), and YoutubeVOS (Xu et al. 2018), compared to the baseline SAM2 model. The main results are shown in Table 12. For each dataset, we adopt their official repositories or websites for evaluation for a fair comparison. The experiment follows a zero-shot setting, where we directly generalize the official SAM2 model checkpoints and our checkpoints from the main experiments to these datasets.

The experimental result demonstrates that SAAS achieves a similar performance on single-shot videos compared to the baseline model. On DAVIS2017-val, SAM2 and SAAS exhibit nearly identical performance. SAAS achieves a 0.5% performance improvement on LVOSv2 and YoutubeVOS, while showing a marginal degradation on the complex VOS dataset MOSE.

We consider the comparable performance to the baseline to be expected, as the optimizations in our method rely on the shot transition detection, without incorporating additional designs for single-shot videos. The observation that the improvement on LVOSv2 is mainly brought by unseen objects may demonstrate that our method achieves more accurate detection of object disappearance and reappearance. For the decrease of $\mathcal{J}\&\mathcal{F}$ on MOSE, our SAAS primarily lags behind SAM2 in segmentation contour quality ($\mathcal{F}$), highly likely caused by serious occlusion phenomena

(a) Extreme close-up view

(b) Appearance change

Figure 11: Some visualized failure cases of the SAAS method on the Cut-VOS benchmark. Case (a) shows an extreme close-up view that suddenly zooms in on the dashboard of a motor. Case (b) requires the model to match the same person with different clothing and hairstyles. Our SAAS method still has difficulties correctly segmenting these cases, which require a stronger reasoning ability.

in MOSE. The TMA strategy lacks a specialized design for occlusion scenarios, directly placing objects on the top layer during replication. Additionally, occlusion instances are underrepresented in the training dataset (YTVOS). The bias in the data distribution may cause the drop in $\mathcal{F}$.

C.3 Oracle Experiments

In this section, we supplement an oracle experiment which assumes the models possess a perfect cross-shot segmentation module. This means that the models can always segment the target object correctly in each shot segment, i.e., the ground truth mask is provided in the first frame of each shot. The result is shown in Table 13.

The $\mathcal{J}_t$ metrics haven't reached 100% mainly due to the delayed cut in transition type, which requires not only the first ground truth mask of the shot. Also, some refinements made by the model bring minor disturbances. However, the $\mathcal{J}\&\mathcal{F}$ and $\mathcal{J}_t$ have reached a very high level overall. This reveals that the most challenging aspect of our proposed Cut-VOS benchmark lies in complex shot transitions, rather than coherent segmentation within the same shot. While existing methods can achieve a high-quality *intra-shot* segmentation, they struggle with *inter-shot* segmentation. Under the premise of disregarding shot transitions, the SAM2 model and SAAS method demonstrate comparable segmentation performance, which aligns with the experimental results in Section C.2.

C.4 Qualitative Results

We provide more qualitative experiment results on our Cut-VOS benchmark, as shown in Figure 10, to further show the superiority of the proposed SAAS method compared to the existing VOS models. In the upper case, the video contains many transitions of different types, including *close-up view*, *horizon transformation*, *scene change*, *etc*. The SAM2 model misses the target objects at a very early time, while the SAAS method correctly tracks and segments them, despite some minor artifacts. In the down case, we mainly show a sample to track static objects. Though these static objects don't perform complex motion patterns and significant appearance changes, the SAM2 model struggles to correctly track them coherently and distinguish them among similar distractors in complex multi-shot videos. In contrast, SAAS exhibits improved segmentation performance.

C.5 Limitations and Failure Cases

In this section, we further investigate the latent limitations of the SAAS method. Upon careful examination of the generated mask, we observe that, despite enhanced object matching across shots, SAAS still relies on visual feature matching, which lacks robust long-range reasoning. This deficiency contributes to segmentation errors, especially in cases where the target object's appearance completely changes during shot transitions, as depicted in Figure 11. Case (a) involves an extreme *close-up view* transition, rapidly zooming in on a local sub-region of the object that was entirely unseen in the previous frames, rendering some of our advancement strategy ineffective. In such cases, humans rely on commonsense knowledge or inferences about filming intent for correct decision-making, a process that poses significant challenges for current methods. In case (b), we highlight a girl which have different clothing and hairstyles across the shots. Unluckily, our method tends to misidentify her as a separate individual, treating her as a distractor. Our proposed TMA strategy can not effectively simulate similar scenarios, which may be a key contributing factor to the observed phenomenon.