Attack-Inference Aligned Universal Adversarial Attacks on Video Object Segmentation

Ruiqi Shen[1] **Chang Liu**[2 *] **Henghui Ding**[1*]
[1]Institute of Big Data, College of Computer Science and Artificial Intelligence, Fudan University, China
[2]Shanghai University of Finance and Economics
henghui.ding@gmail.com

Abstract

Modern video object segmentation (VOS) relies on a two-stage inference paradigm: target initialization and memory-based propagation. However, existing adversarial attacks on VOS often overlook this structure. To bridge this gap, we propose **AUV**, an adversarial framework that **A**ligns with the **U**niversal two-stage paradigm of **V**OS models. Specifically, AUV optimizes a universal adversarial perturbation (UAP) through decoupled training strategies tailored for both initialization and propagation stages. By disrupting the shared memory space, it achieves prompt-agnostic attacks across different modalities, including masks, points, boxes, and language. Extensive experiments show that AUV establishes state-of-the-art VOS attack performance, such as degrading mask-based SAM3 to only 8.4 $\mathcal{J}\&\mathcal{F}$ on DAVIS17-val (compared to 55.4 for the previous SOTA under the same training setup). It also exhibits superior temporal dynamics with the fastest onset, the most enduring persistence, and the support of any-time mid-video attacks. The AUV optimization framework is applicable to different VOS models, revealing vulnerabilities in real-world streaming VOS. Code is available at https://github.com/FudanCVL/AUV.

1 Introduction

Modern video object segmentation (VOS) [37, 4, 40, 2] systems follow a two-stage inference paradigm. In the first stage, the target is specified through a prompt, such as a mask, point, box, or language. In the second stage, the model propagates the target across frames using memory, with little or no further prompting. This two-stage design is foundational to recent VOS models such as XMem [6], Cutie [4], SAM2 [40], SAM3 [2], and is central to their robust performance.

Meanwhile, despite the rapid development of adversarial attacks, including those targeting VOS models [24, 60, 28, 54], **existing methods suffer from a fundamental training-inference misalignment**: they typically optimize the attacks under a single-stage formulation. Specifically, they either focus on the prompt-conditioned initial frame while neglecting memory-based temporal propagation [28, 24], or apply the same optimization objective for all frames during training [60, 54]. This mismatch between **a uniform attack optimization strategy and the two-stage VOS inference pipeline** often leads to two critical limitations: **(1)** the attacks do not effectively corrupt latent memory representations stored in the memory bank, potentially allowing the model to recover the target during propagation; **(2)** they often target the predicted mask, a prompt-biased by-product in VOS. Consequently, attacks optimized with sparse points or boxes often struggle to transfer to other prompts encoded by different branches, such as dense masks or language.

*Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

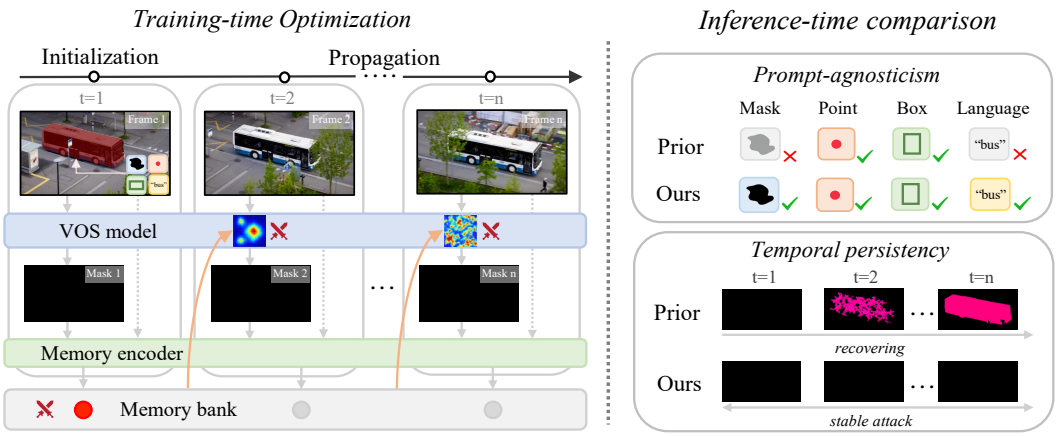


Figure 1: AUV aligns attack optimization with the two-stage VOS inference pipeline (initialization and propagation), yielding more persistent and prompt-agnostic attacks than prior methods.

Motivated by these observations, we propose **AUV**, a stage-**A**ligned **U**niversal **V**OS attack framework that optimizes a universal adversarial perturbation (UAP) to disrupt both the initialization and propagation stages of the target VOS model. As illustrated in Figure 1 (left), AUV decouples the attack optimization into two complementary stages that align with the VOS inference pipeline. **During prompt-based initialization**, it optimizes the UAP to corrupt the target's memory feature anchor before it is permanently stored in the memory bank, thereby erasing the target identity at its source. **During memory-based propagation**, it optimizes the **same** UAP to flatten the spatial saliency of memory-attended feature maps, pushing their activations toward spatial uniformity and thereby severing temporal target association across frames.

Our paradigm-aligned design not only delivers stronger attacks, with faster onset and more sustained temporal persistence, but also supports a broader range of prompt modalities and more flexible attack timing, as illustrated in Figure 1 (right). First, by targeting the shared latent memory space, AUV avoids prompt-specific optimization and transfers zero-shot across different prompt modalities, including masks, points, boxes, and language. Second, by targeting the propagation stage, AUV supports any-time mid-video attacks, where the perturbation can be injected from any intermediate frame and remains effective throughout the subsequent sequence.

Extensive experiments validate our design. Under the same training setup and perturbation budget, AUV consistently outperforms prior attacks across VOS benchmarks and prompt modalities on SAM3, with the performance gap even more pronounced during pure memory-based propagation. Moreover, the optimization framework of AUV applies naturally to different VOS models, suggesting it exploits shared properties of modern VOS systems, rather than a model- or prompt-specific weakness. The main contributions are summarized as follows:

- **Structural Alignment:** We identify a fundamental mismatch between existing uniform attack optimizations and the two-stage VOS inference pipeline. To address this, we propose AUV, a stage-aligned UAP-based attack framework that simultaneously corrupts memory anchor during initialization and flattens memory-attended features during propagation.

- **Prompt-Agnostic Attack:** Our attack targets the shared latent memory space rather than prompt-conditioned output masks, enabling zero-shot generalization across different prompt modalities, including mask, point, box, and language.

- **Metrics & Empirical Insights:** We introduce new metrics to evaluate the onset speed and persistence of video attacks. Extensive experiments show AUV's new SOTA attack performance, while further analyses provide insights into its underlying mechanisms.

2 Related works

2.1 Promptable Video Segmentation

Promptable Video Segmentation focuses on segmenting explicitly specified target(s) within videos, which includes Video Object Segmentation (VOS) [37, 53, 48, 55, 8, 27, 6, 5, 4], Promptable Concept Segmentation (PCS) [2, 14, 12, 11, 29, 17, 16, 15, 41, 26, 44], and interactive video segmentation [7, 34, 52, 40, 43]. Regardless of their differences, these tasks follow a **consistent two-stage inference paradigm**: prompt-based target initialization and memory-based temporal propagation. The former processes diverse prompts such as masks, points, boxes, and language to encode target identity, while the latter propagates memory-conditioned target information throughout the sequence for consistent video segmentation. This shared structure motivates an attack design tailored to the distinct roles of initialization and propagation.

2.2 Adversarial Attacks on Segmentation Models

Adversarial attacks on segmentation models add pixel-level perturbations to input images to disrupt segmentation predictions. Existing image-level methods [22] either disrupt image features to erode target identity [25, 58], or suppress output masks through tailored strategies [57, 61, 21, 45, 3, 42]. Recently, adversarial attacks have been extended to video segmentation models [24, 60, 28, 54], either targeting only the initial frame [28, 24] or using a shared optimization objective for all frames within the video sequence [54, 60].

Meanwhile, recent works have also explored "cross-prompt" attacks with promising results [60, 30, 23]. However, cross-prompt transferability demonstrated with points or boxes does not necessarily imply generalization to other prompt modalities, particularly dense masks or language, which are processed through distinct encoding pathways. In this work, we aim for a prompt-agnostic attack that remains effective regardless of prompt modalities by corrupting the shared latent memory space.

3 Methodology

3.1 Preliminaries and Problem Formulation

A video sequence is denoted as $\mathcal{V} = \{I_1, I_2, \ldots, I_T\}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$. Given a target-specifying prompt P (mask, point, box, or language) applied to the initial frame, a VOS model Φ predicts a binary mask sequence $\{M_i\}_{i=1}^{T}$ with $M_i \in \{0,1\}^{H\times W}$. To attack Φ, we optimize a universal adversarial perturbation (UAP) $\delta \in \mathbb{R}^{H\times W\times 3}$ that can be applied consistently to every frame and is strictly bounded by an ℓ_∞ budget ϵ to keep it invisible:

$$\|\delta\|_\infty \le \epsilon. \tag{1}$$

The adversarial video is $\tilde{\mathcal{V}} = \{I_i + \delta\}_{i=1}^{T}$, and the attack objective is to maximize the deviation between the adversarial prediction and the clean prediction:

$$\max_{\delta} \sum_{i=1}^{T} \mathcal{D}(\tilde{M}_i, M_i) \quad \text{s.t.} \quad \|\delta\|_\infty \le \epsilon, \tag{2}$$

where $\tilde{M}_i$ is the adversarial mask predicted at frame i and $\mathcal{D}(\cdot,\cdot)$ quantifies spatial divergence.

Two-stage VOS inference pipeline. Modern VOS models implement Φ through two distinct stages: an *initialization stage* Φ_{init} that produces an initial memory feature E_1, which is permanently stored in the memory bank $\mathcal{B}$ as an anchor; and a *propagation stage* Φ_{prop} that, for each subsequent frame, performs memory readout from $\mathcal{B}$ for temporal target association.

3.2 Paradigm Misalignment and Method Overview

Existing VOS attacks optimize δ using a uniform or single-stage objective, typically maximizing $\sum_i \mathcal{D}(\tilde{M}_i, M_i)$ either only on the initial frame or on all frames with the same supervision strategy. This ignores the distinct roles of Φ_{init} and Φ_{prop} in VOS inference. **We refer to this mismatch**

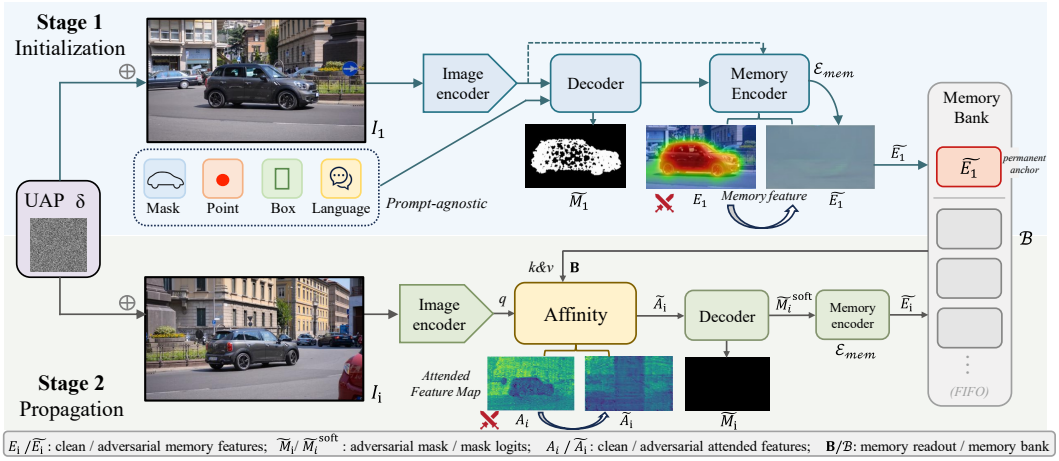


Figure 2: **Overview of AUV.** Our UAP-based AUV is optimized to corrupt the initial memory feature ($\tilde{E}_1$), attacking target identity at its source, and to flatten the spatial saliency of memory-attended feature maps ($\tilde{A}_i$), collapsing target association during propagation. It targets the shared latent memory space for prompt-agnostic attacks and supports attack optimization on different VOS models.

between uniform attack optimization and two-stage VOS inference as a *training-inference paradigm misalignment*, which could insufficiently disrupt both the initial memory anchor that establishes target identity and the memory readout that propagates it throughout the sequence.

Design principle. The overview of our method is shown in Figure 2. We propose **AUV**, whose optimization aligns with the two-stage VOS inference pipeline by decoupling the UAP training into two distinct stages that directly mirror Φ_{init} and Φ_{prop}:

- **Stage I - Initialization-aligned corruption.** We corrupt the initial memory feature $\tilde{E}_1$ *before* it is permanently stored in $\mathcal{B}$ as an anchor, erasing the target identity at its source. The target can be specified at initialization using a prompt P of any modality.
- **Stage II - Propagation-aligned flattening.** We flatten the spatial saliency of memory-attended feature maps $\tilde{A}_i$ in subsequent frames, disrupting the temporal chain of target propagation. The prompt encoder is disabled, consistent with VOS inference where no prompts are provided after initialization. The **same** UAP δ is shared across both stages.

3.3 Stage I: Corrupting the Permanent Memory Anchor

Why initialization must be attacked differently. In Φ_{init}, the prompt P first produces an initial mask M_1, after which a memory encoder $\mathcal{E}_{\text{mem}}$ fuses the image feature F_1 with the underlying soft values to obtain a target-specific memory feature $E_1 \in \mathbb{R}^{h\times w\times c}$. Modern VOS models store this E_1 in the memory bank as a ***permanent anchor*** for the entire sequence. Meanwhile, prior methods typically optimize δ to corrupt the *output* mask M_1 [24, 60, 28, 54], an intermediate product in the VOS pipeline. However, an empty binarized output mask $\tilde{M}_1$ does not imply that its underlying soft values $\tilde{M}_1^{\text{soft}}$ used for memory encoding contain no target information. As shown in Fig. 3, these residual spatial cues keep the target visible in $\tilde{E}_1$ despite the empty output mask, which allows potential recovery during subsequent propagation.

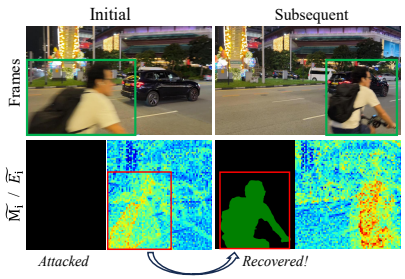


Figure 3: Previous video attacks [24, 28, 54, 60] suppress mask $\tilde{M}_1$, but $\tilde{E}_1$ remains subtly visible, enabling potential target recovery in later frames.

Direct memory-feature corruption. We therefore optimize the UAP δ to directly corrupt the memory feature E_1. Specifically, we define three representations: the adversarial feature $\tilde{E}_1 = \mathcal{E}_{\text{mem}}(I_1+\delta, \tilde{M}_1^{\text{soft}})$, the clean feature $E_1 = \mathcal{E}_{\text{mem}}(I_1, M_1^{\text{soft}})$, and a background-only reference $E_1^{\emptyset} = \mathcal{E}_{\text{mem}}(I_1, \mathbf{0})$, where $\mathbf{0}$ denotes an all-zero mask prior with no spatial target cues. The initialization-aligned loss pulls the adversarial feature toward the background state while pushing it away from the clean state:

$$\mathcal{L}_{\text{init}} = \frac{\|\tilde{E}_1 - E_1^{\emptyset}\|_2^2}{hwc} + \exp\left(-\frac{10\|\tilde{E}_1 - E_1\|_2^2}{hwc}\right), \tag{3}$$

where the exponential term pushes the adversarial feature away from the clean feature.

Unlike most prior video attacks that disrupt prompt-conditioned masks [24, 28, 54, 60], $\mathcal{L}_{\text{init}}$ is formulated in the shared memory feature space to corrupt the permanent memory anchor $\tilde{E}_1$ that preserves the target identity needed for subsequent propagation.

3.4 Stage II: Flattening Memory-Attended Features

Why propagation must be attacked differently. For each subsequent frame $i \in \{2, \ldots, T\}$, Φ_{prop} performs temporal association by cross-attending the image feature F_i to the memory readout $\mathbf{B} \in \mathbb{R}^{N\times h\times w\times c}$, producing a memory-attended feature map $A_i \in \mathbb{R}^{h\times w\times c}$:

$$A_i = \text{softmax}(F_i\mathbf{B}^\top)\mathbf{B} + F_i, \tag{4}$$

where N denotes the number of frames currently stored in the memory bank $\mathcal{B}$. In a benign VOS model, A_i exhibits strong spatial saliency that localizes the target [37, 20]. This saliency is the **carrier of temporal propagation**: the decoder uses A_i to predict M_i, whose soft values are then fused with image features to form a new memory feature stored in $\mathcal{B}$ for subsequent frames.

Propagation erosion via variance minimization. From this point, we optimize the **same** δ to flatten $\tilde{A}_i$ into a spatially uniform feature map, stripping away all target-localized activation peaks. Let $\tilde{\mu}_i \in \mathbb{R}^{1\times 1\times c}$ be the spatial mean of $\tilde{A}_i$: $\tilde{\mu}_i = \frac{1}{hw}\sum_{x=1}^{h}\sum_{y=1}^{w}\tilde{A}_i(x, y, :)$, the propagation-aligned loss penalizes spatial variance as follows:

$$\mathcal{L}_{\text{prop}}^{(i)} = \frac{1}{hwc}\sum_{x=1}^{h}\sum_{y=1}^{w}\left\|\tilde{A}_i(x, y, :) - \tilde{\mu}_i\right\|_2^2. \tag{5}$$

Flattening the saliency of $\tilde{A}_i$ prevents the decoder from recovering the target. The resulting collapsed state is then encoded into $\mathcal{B}$, triggering cascading failure throughout the remaining sequence.

3.5 Joint Two-Stage Optimization

Staged curriculum. Directly optimizing the combined loss from scratch is unstable: if the memory anchor E_1 is not yet corrupted, Stage II will have to disrupt propagation that still carries a strong clean target signal. We therefore adopt a two-phase curriculum that follows the causal dependency between initialization and propagation.

1. **Phase I.** Optimize δ using only $\mathcal{L}_{\text{init}}$ on the initial frame, without propagation, to establish a corrupted initial memory anchor $\tilde{E}_1$.
2. **Phase II.** Extend training to full video sequence. The initial frame continues to be supervised by $\mathcal{L}_{\text{init}}$, while all subsequent frames are jointly penalized by $\mathcal{L}_{\text{prop}}^{(i)}$.

The overall objective for optimizing the UAP δ is formulated as:

$$\mathcal{L}_{\text{total}} = \lambda\mathcal{L}_{\text{init}} + \sum_{i=2}^{T}\mathcal{L}_{\text{prop}}^{(i)}, \tag{6}$$

where $\lambda = 0.1$. In both phases, δ is optimized by targeting memory and memory-attended features, avoiding supervision on predicted masks or other prompt-conditioned decoder outputs. This enables robust prompt-agnostic attack. See Section 4.5 for in-depth analysis of the underlying mechanism.

Table 1: Attack performance on SAM3 [2] across VOS benchmarks and prompt modalities.
(All applicable baselines are retrained on SAM3 under the same settings.)

Methods	DAVIS17 ($\mathcal{J}\&\mathcal{F}\downarrow$)			YTVOS19 ($\mathcal{G}\downarrow$)			MOSEv2 ($\mathcal{J}\&\dot{\mathcal{F}}\downarrow$)			MOSEv1 ($\mathcal{J}\&\mathcal{F}\downarrow$)		
	Mask	Point	Box	Mask	Point	Box	Mask	Point	Box	Mask	Point	Box
Clean	92.2	72.1	89.1	89.7	80.0	86.8	60.3	55.8	58.9	78.4	70.1	77.0
UAPGD [10]	64.1	21.5	57.4	58.8	29.3	52.9	35.9	29.1	35.0	49.7	35.4	47.8
VOSPGD [24]	61.2	19.8	54.5	49.5	21.8	42.5	35.0	28.1	33.5	44.5	30.5	42.1
AttackSAM [57]	68.3	30.2	63.2	63.0	30.9	53.2	42.6	36.4	38.2	52.5	41.2	47.3
DarkSAM [61]	58.5	25.0	52.5	51.7	23.1	45.6	32.5	26.9	32.0	43.0	28.0	41.3
UAP-SAM2 [60]	55.4	23.9	47.4	46.3	22.0	35.8	30.4	25.5	29.1	36.4	22.5	31.3
AUV (Ours)	**8.4**	**7.1**	**7.6**	**14.0**	**10.3**	**12.7**	**24.6**	**19.9**	**23.5**	**21.1**	**18.3**	**20.2**

4 Experiments

4.1 Implementation Details

Following prior works [35, 10, 61, 60], we set the perturbation bound to $\epsilon = 10/255$ for all methods. We also provide results with smaller ϵ in the appendix. Following UAP-SAM2 [60], we train all methods on SAM3 using the YTVOS18 training set [50] for 10 epochs. For a fair comparison, we sample 8 frames per sequence and use the same learning-rate schedule across all methods. Training is conducted on a single NVIDIA RTX A6000 GPU with 48 GB of VRAM.

4.2 Evaluation Metrics

Following prior works [60, 24, 61], we use the standard VOS metrics [38] to evaluate adversarial performance. Furthermore, while these metrics reflect overall attack effectiveness, they fail to characterize the *temporal dynamics* of an attack, such as how quickly the attack takes effect or how long it could persist. To capture these aspects, we further introduce three temporal metrics: **Latency** (Lat), **Coverage** (Cov), and **Delayed-Coverage** (Cov_k).

Let $\mathcal{F}_v$ denote the set of *valid frames* successfully tracked by the benign VOS model. Specifically, a frame is considered valid if its benign score (e.g. $\mathcal{J}\&\mathcal{F}$) exceeds 0.3, following [60]. Among these valid frames, we define the set of *collapsed frames* $\mathcal{F}_c$ as those whose score under attack drops by at least 50% relative to the benign score.

Based on these two sets, we define: • **Latency** (Lat), which measures the normalized temporal position of the first collapsed frame in $\mathcal{F}_v$ and thus reflects how quickly the attack takes effect; • **Coverage** (Cov), which measures the proportion of collapsed frames in $\mathcal{F}_v$ and thus reflects the temporal persistence of the attack; and • **Delayed-Coverage** (Cov_k), which measures the attack coverage Cov for the mid-video attack starting from frame k. Formally, these metrics are defined as:

$$Lat = \begin{cases} \dfrac{\min_{t\in\mathcal{F}_c} \mathrm{rank}_{\mathcal{F}_v}(t)}{|\mathcal{F}_v|}, & \mathcal{F}_c \neq \emptyset, \\ 1, & \mathcal{F}_c = \emptyset, \end{cases} \qquad Cov = \frac{|\mathcal{F}_c|}{|\mathcal{F}_v|}, \qquad Cov_k = \frac{|\{\, t \in \mathcal{F}_c \mid t \geq k \,\}|}{|\{\, t \in \mathcal{F}_v \mid t \geq k \,\}|}, \tag{7}$$

where $\mathrm{rank}_{\mathcal{F}_v}(t)$ denotes the 1-based temporal position of frame t within $\mathcal{F}_v$, and k denotes the frame of mid-video attack injection. Lower Lat reflects faster onset, while higher Cov and Cov_k indicate better persistence. Low Cov implies potential target recovery during propagation.

4.3 Main Results

Overall Attack Performance. We evaluate AUV on the validation sets of seven benchmarks across four prompt modalities (masks, points, boxes, and language) on SAM3 [2]. As shown in Tables 1 and 2, AUV consistently outperforms prior attacks across benchmarks and prompt types, such as reducing mask-based $\mathcal{J}\&\mathcal{F}$ on DAVIS17 to only 8.4, while achieving the fastest attack onset (Lat) and the highest coverage (Cov). MOSE is excluded from temporal analysis because per-frame ground-truth masks are unavailable. AUV also generalizes to language prompts, achieving 0.1 cgF1 and 8.5 pHOTA on the SA-V split of SA-Co/VEval (Table 3) and enabling one-to-many attacks

Table 2: Temporal dynamics of the attacks on SAM3.

Method	DAVIS17						YTVOS19					
	Mask			Point			Mask			Box		
	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$
SegPGD [21]	53.7	36.5	45.7	10.0	86.4	87.5	53.9	38.6	46.4	46.6	46.6	55.3
DarkSAM [61]	58.3	33.5	41.6	11.0	84.6	85.0	56.1	36.6	43.8	46.6	44.8	52.6
UAP-SAM2 [60]	52.5	35.9	43.6	10.4	85.1	85.2	50.9	41.1	47.4	36.1	56.6	63.5
AUV (Ours)	**5.5**	**89.2**	**88.1**	**2.2**	**90.3**	**88.6**	**8.6**	**85.1**	**84.6**	**7.0**	**86.8**	**85.9**

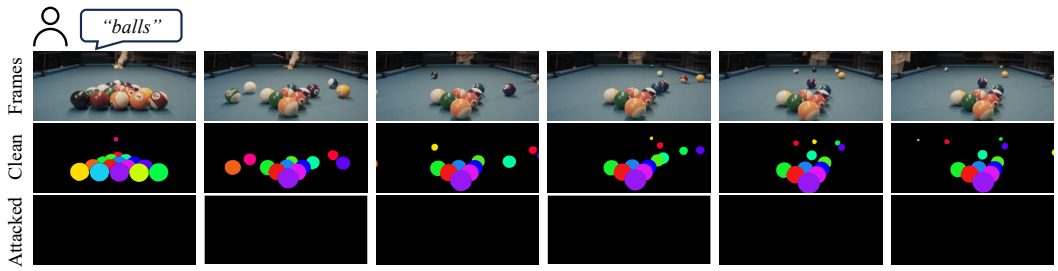


Figure 4: AUV simultaneously suppresses multiple targets specified by a single language prompt.

Table 3: Attack performance on SAM3 across language-based SA-Co/VEval benchmarks [2].

Methods	SA-V				YT-Temporal-1B				SmartGlasses			
	cgF1	pHOTA	pDetA	pAssA	cgF1	pHOTA	pDetA	pAssA	cgF1	pHOTA	pDetA	pAssA
Clean	29.3	60.7	44.7	83.2	50.2	70.5	60.5	82.7	33.5	60.2	46.2	79.3
UAD [31]	2.4	24.9	8.5	74.0	7.8	31.7	15.0	67.5	11.1	36.6	21.0	64.4
DarkSAM [61]	4.0	30.5	12.6	75.0	10.7	38.8	22.7	67.0	12.4	38.8	23.0	65.9
UAP-SAM2 [60]	1.7	23.9	7.8	74.6	3.5	26.2	11.1	62.4	9.1	36.3	21.4	62.4
AUV (Ours)	**0.1**	**8.5**	**1.2**	**64.3**	**0.3**	**9.1**	**1.7**	**49.9**	**1.9**	**18.4**	**6.4**	**53.4**

Table 4: Attack performance on different VOS models and benchmarks using mask prompts.

VOS models	Venue	State	DAVIS17			YTVOS19					MOSEv2			MOSEv1		
			$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	$\dot{\mathcal{J}}\&\dot{\mathcal{F}}$	$\dot{\mathcal{J}}$	$\dot{\mathcal{F}}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
XMem [6]	[ECCV'22]	Clean	86.2	82.9	89.5	85.5	84.3	88.6	80.3	88.6	36.3	34.7	37.9	57.6	53.3	62.0
		Adv	24.0	22.4	25.6	26.3	26.2	28.4	24.1	26.5	28.3	27.5	29.1	25.7	24.8	26.6
Cutie-base [4]	[CVPR'24]	Clean	87.9	84.6	91.1	87.0	86.0	90.5	82.0	89.6	42.8	41.1	44.4	69.9	65.9	74.1
		Adv	18.3	17.5	19.1	22.4	23.6	21.0	22.2	22.8	21.1	20.6	21.6	22.2	21.3	23.1
SAM2.1-T [40]	[ICLR'25]	Clean	88.8	85.4	92.2	87.4	85.5	90.0	83.0	91.3	45.9	44.1	47.8	71.6	67.4	75.9
		Adv	17.8	17.2	18.4	25.0	25.4	23.8	24.6	26.2	20.6	19.8	21.4	24.5	23.4	25.6
SAM2.1-B+ [40]	[ICLR'25]	Clean	89.0	86.0	92.0	88.0	86.0	90.4	83.6	91.8	46.0	44.2	47.8	73.6	69.6	77.6
		Adv	12.1	12.4	11.8	19.2	22.8	17.6	18.5	17.9	26.5	26.4	26.7	23.2	22.5	23.8
SAM2.1Long-B+ [19]	[ICCV'25]	Clean	89.4	86.4	92.5	88.1	86.0	90.5	84.0	91.8	47.4	45.5	49.2	74.7	70.6	78.8
		Adv	15.9	16.0	15.7	23.0	26.5	21.2	22.0	22.3	25.2	25.0	25.4	23.4	22.4	24.3

that simultaneously suppress multiple objects specified by a single language prompt (Figure 4). See appendix for more quantitative comparisons and visualizations.

Applicability of AUV to other VOS models. Table 4 demonstrates that the AUV framework can also be applied to other VOS models under the same optimization procedure, including SAM2 [40], Cutie [4] and XMem [6]. For example, it reduces the mask-based $\mathcal{J}\&\mathcal{F}$ of SAM2.1-B+ on DAVIS17 from 89.0 to 12.1, and the $\mathcal{G}$ of Cutie-base on YTVOS19 from 87.0 to 22.4, demonstrating that the proposed two-stage attack framework is applicable across different VOS models.

4.4 Two-Stage Joint Optimization

We validate our two-stage optimization on SAM3 by comparing it with two variants: training only on the initial frame (Stage I / Init. Only) or training only on subsequent frames (Stage II / Sub. Only).

Attack performance across inference settings. Table 5 compares their performance under three different *inference-time* attack settings: attacking only the initial frame (Init.); attacking only the subsequent frames (Sub.); or attacking the entire sequence (All). As shown, training on either stage

Table 5: Ablation on training stages and inference settings. Init.: Initial frame; Sub.: Subsequent frames.

Train \ Infer.	DAVIS17 $\mathcal{J}\&\mathcal{F}$			YTVOS19 $\mathcal{G}$			MOSEv1 $\mathcal{J}\&\mathcal{F}$		
	Init.	Sub.	All	Init.	Sub.	All	Init.	Sub.	All
Init. Only	47.2	23.9	26.2	46.0	24.4	29.4	43.2	20.9	21.4
Sub. Only	70.1	13.6	28.9	57.4	16.4	29.8	52.8	**18.4**	23.0
Init. + Sub.	**23.3**	**9.6**	**8.4**	**24.1**	**15.3**	**14.0**	**29.5**	20.3	**21.1**

Table 6: Ablation on temporal dynamics: Lat (Latency), Cov (Coverage), and Cov_k.

Train \ Infer.	DAVIS17			YTVOS19		
	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$
Init. Only	21.3	69.9	68.5	26.5	64.4	69.5
Sub. Only	21.4	64.9	87.8	29.1	62.2	**84.8**
Init. + Sub.	**5.5**	**89.2**	**88.1**	**8.6**	**85.1**	84.6

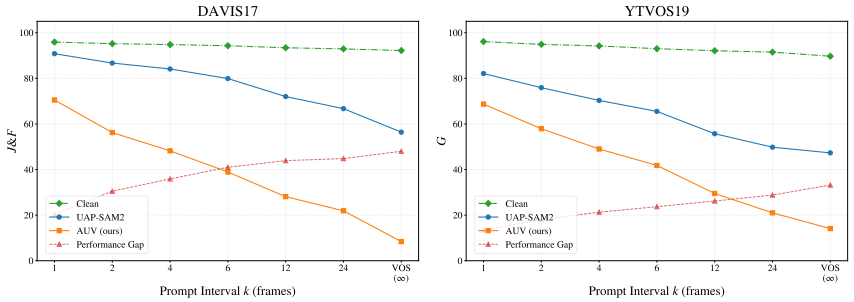


Figure 5: Comparison of attack performance under different prompting intervals k, where a larger k shifts the model's reliance from prompt-based initialization to memory-based propagation.

Table 7: Prompt-agnostic capability of AUV by training on one prompt type and evaluating on other unseen prompt types. †: Method's default setting.

Method	Training	DAVIS17 ($\mathcal{J}\&\mathcal{F}\downarrow$)			YTVOS19 ($\mathcal{G}\downarrow$)			SA-V	YT-1B	SG
		Mask	Point	Box	Mask	Point	Box	Language (pHOTA↓)		
UAP-SAM2 [60]	Scanning Points†	55.4	23.9	47.4	46.3	22.0	35.8	23.9	26.2	36.3
	Box	62.1	21.8	52.5	50.2	21.8	38.8	27.7	34.0	40.4
AUV (Ours)	Mask	**8.4**	7.1	7.6	**14.0**	10.3	12.7	**8.5**	**9.1**	**18.4**
	Point	17.7	**5.7**	11.2	19.1	**9.0**	13.0	12.2	13.8	21.5
	Box	15.2	7.8	**6.5**	17.5	11.0	**11.1**	11.7	12.4	20.2

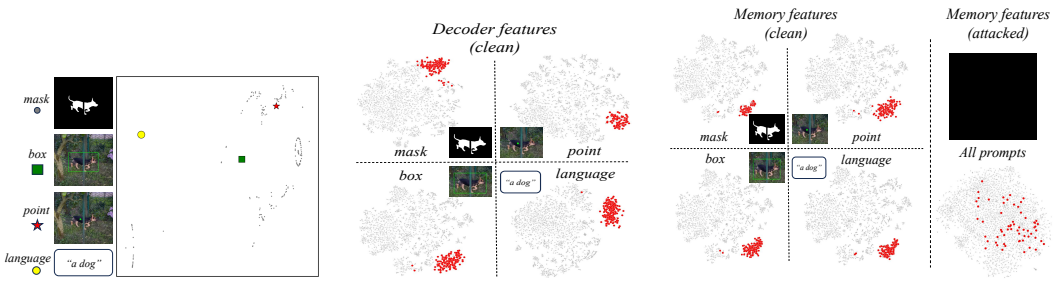
does not generalize well to the other. For instance, the UAP trained only on subsequent frames has limited effectiveness when applied to the initial frame, with SAM3 retaining a 70.1 $\mathcal{J}\&\mathcal{F}$ on DAVIS17. This reveals the intrinsic gap between the initialization and propagation stages of VOS models and supports our main motivation. With two-stage joint optimization, AUV achieves strong attack performance across all three settings, outperforming both single-stage-trained variants.

Temporal dynamics. Table 6 compares the temporal dynamics of three training configurations. The Cov_{20} results show that training only on the initial frame generalizes poorly to mid-video attacks, which is increasingly important with the rise of online VOS methods [1, 9, 49]. In contrast, our two-stage training achieves both faster onset (Lat) and higher coverage (Cov), along with strong mid-video attack performance (Cov_{20}), demonstrating superior temporal dynamics.

From prompt-based initialization to memory-driven propagation. We compare AUV with the previous SOTA, UAP-SAM2 [60], under periodic online correction, providing a ground-truth mask every $(k+1)$ frames for $k \in \{1, 2, 4, 6, 12, 24, \infty\}$. Here, $k = 1$ means the model is corrected every other frame, and $k = \infty$ corresponds to the standard VOS, with a mask provided only at initialization. Larger k increases reliance on memory-driven propagation and reduces human intervention. Figure 5 shows that AUV consistently outperforms UAP-SAM2 across all k values, from frequent ground-truth corrections to pure memory-driven propagation, with an even larger advantage in the latter, highlighting its sustained effectiveness throughout the VOS propagation stage.

4.5 Prompt-Agnostic Attack

Overall prompt-agnostic generalization. We investigate the prompt-agnostic capability of AUV on SAM3 by **training each variant exclusively on one prompt modality and evaluating it zero-shot on other unseen modalities**. As shown in Table 7, AUV remains highly effective regardless of the



(a) Prompt embeddings. (b) Decoder features. (c) Memory features.

Figure 6: t-SNE visualizations of latent features: (a) prompt embeddings; (b) final-block decoder features; and (c) memory features before (left) and after (right) the attack. Foreground and background features in ground-truth are marked in red and gray, respectively.

Table 8: Pairwise cross-prompt cosine similarities of decoder and memory features, averaged over all videos of each benchmark.

	Decoder Features			Memory Features E_1		
Benchmark	Mask–Point	Mask–Box	Point–Box	Mask–Point	Mask–Box	Point–Box
DAVIS17	0.780	0.682	0.860	0.983	0.997	0.985
YTVOS19	0.784	0.679	0.852	0.984	0.993	0.988
MOSEv1	0.772	0.626	0.852	0.991	0.998	0.992
MOSEv2	0.761	0.611	0.848	0.993	0.999	0.994

prompt type used to specify the target during training. For instance, the mask-trained UAP reduces point-based $\mathcal{J}\&\mathcal{F}$ to 7.1 on DAVIS17, while the box-trained UAP drops mask-based $\mathcal{G}$ to 17.5 on YTVOS19. This prompt-agnosticism extends to language prompts, with mask-trained UAP reaching 9.1 pHOTA on YT-1B of SA-Co/VEval. In contrast, existing method [60] suffers from a significant modality gap, showing degradation on dense masks due to the sparse-branch-specific optimization.

Investigating the root cause of prompt-agnosticism. Figure 6a visualizes the t-SNE [46] map of different prompt embeddings, where distinct prompt modalities form separated clusters. Accordingly, the decoder features used for mask prediction remain prompt-conditioned and exhibit different distributions, as shown in Figure 6b. Consequently, prior attacks targeting predicted masks can become tied to specific prompt-encoding branches, and therefore, struggle to generalize to other prompts processed by distinct branches (e.g., from sparse points to dense masks). In contrast, the distributions of memory features are highly aligned across prompt modalities (Figure 6c, left), suggesting that prompt-specific variations are largely filtered out in this shared space while common target cues are retained. By attacking this shared latent space (Figure 6c, right), AUV entangles foreground and background features, thus collapsing predictions across all prompt modalities.

To further reveal the mechanism behind this effect, we quantitatively analyze all videos across multiple VOS benchmarks. For each video, we compute pairwise cross-prompt cosine similarities separately for decoder and memory features using mask, point, and box prompts. As reported in Table 8, decoder-feature similarities can be as low as 0.6 to 0.7, revealing clear modality differences, particularly between dense (mask) and sparse (point & box) prompts. In contrast, memory-feature similarities consistently range from 0.98 to 1.00, confirming that the **memory space remains highly consistent regardless of whatever prompts**. This reveals the basis of AUV's prompt-agnosticism, which allows a UAP trained with one prompt modality to transfer zero-shot to others.

4.6 Defense Study

We evaluate the robustness of AUV against potential defenses. All experiments are conducted on the DAVIS17 validation set for a controlled comparison.

Spatial perturbation. We distort images using Gaussian blur with standard deviation $\sigma \in \{0.25, 0.5, 0.75, 1.0\}$. This smooths local image details, with larger σ producing stronger blur. As shown in Figure 7a, AUV remains robust across different spatial perturbation levels.

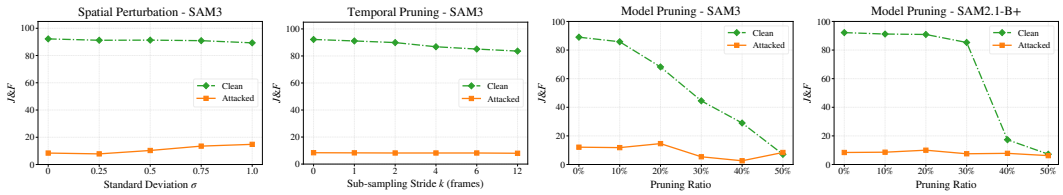


Figure 7: Defense study on spatial perturbation (a), temporal pruning (b), and model pruning (c,d).

Temporal pruning. We sub-sample every k-th frame within each video sequence, discarding the rest frames to reduce frame rate and disrupt temporal continuity of videos. As shown in Figure 7b, AUV's attack efficacy remains stable, demonstrating robustness to temporal discontinuity.

Model pruning. Following [62, 60], we apply magnitude-based unstructured pruning. At each pruning ratio, the smallest-magnitude weights in each targeted layer are set to zero. As in Figure 7c,d, AUV maintains attack effectiveness on both SAM3 and SAM2.1-B+ across different sparsity levels.

Periodic re-prompting. As shown in Figure 5, we provide a ground-truth mask every $(k+1)$ frames to simulate user corrections for target recovery. AUV remains effective across all tested intervals.

Memory-level defenses. We further evaluate random dropping of non-conditioning memory entries and more advanced memory-management strategies. AUV remains effective under both settings; detailed results are provided in the appendix.

5 Limitations and Broader Impact

AUV assumes white-box access to the target VOS model during offline UAP optimization, which is consistent with prior works that require gradient-based source-model access [22, 57, 60], and therefore does not currently support black-box optimization. Once optimized, the fixed UAP requires no further model access, prompt, or per-video optimization. Meanwhile, AUV is primarily designed for memory-based VOS and may be less effective on earlier architectures without explicit memory mechanisms. The attack could potentially be misused to disrupt deployed VOS systems; we therefore emphasize its intended use only for robustness evaluation and scientific research. It can also serve as a tool for assessing VOS robustness to adversarial perturbations and guiding the development of more resilient models.

6 Conclusion

We propose AUV, a universal adversarial attack framework for VOS models. AUV aligns its optimization with the two-stage VOS inference paradigm by decoupling UAP training into target initialization and memory propagation stages with tailored supervisions. By corrupting the shared memory space, it achieves strong prompt-agnostic attacks across masks, points, boxes, and language. Extensive experiments demonstrate AUV's state-of-the-art attack performance and the framework's broad applicability across VOS models. It delivers the fastest attack onset and superior persistence, while supporting flexible mid-video attacks, providing a strong baseline for future research.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62472104 and the Science and Technology Commission of Shanghai Municipality under Grant No. 25511103600, and Shanghai Pujiang Program 2025PJA201. Henghui Ding was supported by Xiaomi Young Scholars Program.

References

[1] Maksym Bekuzarov, Ariana Bermudez, Joon-Young Lee, and Hao Li. Xmem++: Production-level video segmentation from few annotated frames. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 635–644, 2023.

[2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[3] Zhenhua Chen, Chuhua Wang, and David Crandall. Semantically stealthy adversarial attacks against segmentation models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4080–4089, 2022.

[4] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024.

[5] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023.

[6] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European conference on computer vision*, pages 640–658. Springer, 2022.

[7] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5559–5568, 2021.

[8] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in neural information processing systems*, 34:11781–11794, 2021.

[9] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7415–7424, 2018.

[10] Yingpeng Deng and Lina J Karam. Universal adversarial attack via enhanced projected gradient descent. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1241–1245. IEEE, 2020.

[11] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Yu-Gang Jiang. GREx: Generalized referring expression segmentation, comprehension, and generation. *International Journal of Computer Vision*, 134(2):79, 2026.

[12] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. MeViS: A large-scale benchmark for video segmentation with motion expressions. In *IEEE/CVF International Conference on Computer Vision*, 2023.

[13] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 20224–20234, 2023.

[14] Henghui Ding, Chang Liu, Shuting He, Kaining Ying, Xudong Jiang, Chen Change Loy, and Yu-Gang Jiang. MeViS: a multi-modal dataset for referring motion expression video segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11400–11416, 2025.

[15] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16321–16330, 2021.

[16] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. VLT: Vision-language transformer and query generation for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7900–7916, 2023.

[17] Henghui Ding, Song Tang, Shuting He, Chang Liu, Zuxuan Wu, and Yu-Gang Jiang. Multimodal referring segmentation: A survey. *International Journal of Computer Vision*, 134(9):405, 2026.

[18] Henghui Ding, Kaining Ying, Chang Liu, Shuting He, Xudong Jiang, Yu-Gang Jiang, Philip HS Torr, and Song Bai. MOSEv2: A more challenging dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2508.05630*, 2025.

[19] Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. Sam2long: Enhancing sam 2 for long video segmentation with a training-free memory tree. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13614–13624, 2025.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[21] Jindong Gu, Hengshuang Zhao, Volker Tresp, and Philip HS Torr. Segpgd: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness. In *European Conference on Computer Vision*, pages 308–325. Springer, 2022.

[22] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 2755–2764, 2017.

[23] Shize Huang, Qianhui Fan, Zhaoxin Zhang, Xiaowen Liu, Guanqun Song, and Jinzhe Qin. Segment shards: Cross-prompt adversarial attacks against the segment anything model. *Applied Sciences*, 14(8):3312, 2024.

[24] Kaixun Jiang, Lingyi Hong, Zhaoyu Chen, Pinxue Guo, Zeng Tao, Yan Wang, and Wenqiang Zhang. Exploring the adversarial robustness of video object segmentation via one-shot adversarial attacks. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8598–8607, 2023.

[25] Yan Jiang, Guisheng Yin, Ye Yuan, Jingjing Chen, and Zhipeng Wei. Cross-point adversarial attack based on feature neighborhood disruption against segment anything model. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.

[26] Jingjing Li, Yue Feng, Yuchen Guo, Jincai Huang, Yongri Piao, Qi Bi, Miao Zhang, Xiaoqi Zhao, Qiang Chen, Shihao Zou, et al. Sam3-i: Segment anything with instructions. *arXiv preprint arXiv:2512.04585*, 2025.

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

[28] Ping Li, Yu Zhang, Li Yuan, Jian Zhao, Xianghua Xu, and Xiaoqin Zhang. Adversarial attacks on video object segmentation with hard region discovery. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6):5049–5062, 2023.

[29] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[30] Yuchen Liu and Pengxu Wei. Cross-prompt adversarial attack on segment anything model. In *Proceedings of the 2024 12th International Conference on Communications and Broadband Networking*, pages 34–39, 2024.

[31] Jiahao Lu, Xingyi Yang, and Xinchao Wang. Unsegment anything by simulating deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24294–24304, 2024.

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[33] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[34] Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10366–10375, 2020.

[35] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[36] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 262–271, 2020.

[37] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9226–9235, 2019.

[38] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

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

[40] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[41] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[42] Jérôme Rony, Jean-Christophe Pesquet, and Ismail Ben Ayed. Proximal splitting adversarial attack for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20524–20533, 2023.

[43] Ruiqi Shen, Chang Liu, and Henghui Ding. Sam-mt: Real-time interactive multi-target video segmentation. In *European Conference on Computer Vision*, pages 277–295. Springer, 2026.

[44] Ruiqi Shen, Chang Liu, and Henghui Ding. Sam3-dms: Decoupled memory selection for multi-target video segmentation of sam3. *arXiv preprint arXiv:2601.09699*, 2026.

[45] Yufei Song, Ziqi Zhou, Qi Lu, Hangtao Zhang, Yifan Hu, Lulu Xue, Shengshan Hu, Minghui Li, and Leo Yu Zhang. Segtrans: Transferable adversarial examples for segmentation models. *IEEE Transactions on Multimedia*, 2026.

[46] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[47] Jovana Videnovic, Alan Lukezic, and Matej Kristan. A distractor-aware memory for visual object tracking with sam2. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24255–24264, 2025.

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

[49] Qiangqiang Wu, Tianyu Yang, Wei Wu, and Antoni B Chan. Scalable video object segmentation with simplified framework. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13879–13889, 2023.

[50] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[51] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922*, 2024.

[52] 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.

[53] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *European Conference on Computer Vision*, pages 332–348. Springer, 2020.

[54] Rui Yao, Ying Chen, Yong Zhou, Fuyuan Hu, Jiaqi Zhao, Bing Liu, and Zhiwen Shao. Attention-guided adversarial attack for video object segmentation. *ACM Transactions on Intelligent Systems and Technology*, 14(6):1–22, 2023.

[55] Kaining Ying, Hengrui Hu, and Henghui Ding. MOVE: motion-guided few-shot video object segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 11632–11642. IEEE, 2025.

[56] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pages 69–85. Springer, 2016.

[57] Chenshuang Zhang, Chaoning Zhang, Taegoo Kang, Donghun Kim, Sung-Ho Bae, and In So Kweon. Attack-sam: Towards attacking segment anything model with adversarial examples. *arXiv preprint arXiv:2305.00866*, 2023.

[58] Sheng Zheng, Chaoning Zhang, and Xinhong Hao. Black-box targeted adversarial attack on segment anything (sam). *IEEE Transactions on Multimedia*, 27:1901–1913, 2024.

[59] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[60] Ziqi Zhou, Yifan Hu, Yufei Song, Zijing Li, Shengshan Hu, Leo Yu Zhang, Dezhong Yao, Long Zheng, and Hai Jin. Vanish into thin air: Cross-prompt universal adversarial attacks for sam2. *arXiv preprint arXiv:2510.24195*, 2025.

[61] Ziqi Zhou, Yufei Song, Minghui Li, Shengshan Hu, Xianlong Wang, Leo Yu Zhang, Dezhong Yao, and Hai Jin. Darksam: Fooling segment anything model to segment nothing. *Advances in Neural Information Processing Systems*, 37:49859–49880, 2024.

[62] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

Appendix

The appendix is structured as follows:

- **A.** Section A provides a detailed description of all **benchmarks** used in evaluation, including both standard VOS benchmarks and promptable concept segmentation benchmarks.
- **B.** Section B presents comprehensive **quantitative** comparisons between AUV and prior attack baselines across extensive benchmarks and prompt types, along with analysis of temporal dynamics.
- **C.** Section C details the optimization **algorithm** of AUV, including its two-phase curriculum for initialization-aligned memory corruption and propagation-aligned saliency flattening.
- **D.** Section D evaluates AUV's performance under **memory-level defenses**, including random memory dropping and advanced memory-management strategies.
- **E.** Section E further validates the robustness of AUV through two **stress tests** on SAM3: 1) input prompt variations, including spatial transformations and changes in prompt quantity; and 2) stricter perturbation budgets, where ϵ is gradually reduced from standard $10/255$ all the way to $4/255$ to assess AUV's efficacy.
- **F.** Section F highlights the **one-to-many** attack capability of AUV in language-based video segmentation, where a single language prompt corresponds to multiple targets and AUV suppresses them all in one shot.
- **G.** Section G further evaluates AUV on **image segmentation benchmarks**. Specifically, we test its attack performance on ADE20K [59] and RefCOCO [33, 56] under point, box, and language prompts.
- **H.** Section H provides **qualitative** comparisons with prior attack baselines, visually demonstrating the robust attack performance of AUV.

A Benchmarks

We provide additional details on the benchmarks used to evaluate AUV. Our evaluation covers both standard video object segmentation (VOS) benchmarks and promptable concept segmentation (PCS) benchmarks. The former focuses on tracking targets specified by masks, points, or bounding boxes. The latter evaluates language-based video segmentation, where targets are specified by natural language expressions and may correspond to one or multiple object instances. A detailed comparison of these benchmarks is given in Table 9.

A.1 Standard VOS Benchmarks

We evaluate AUV on four widely-adopted VOS benchmarks supporting different forms of target specification, including masks (semi-supervised VOS), points, and bounding boxes (interactive VOS). Details are as follows:

DAVIS17 [39] A high-quality VOS benchmark with precise pixel-level annotations and natural language labels (e.g., *pigs, dog, car-shadow*) for each video sequence.

YouTube-VOS 2019 (YTVOS19) [50] A large-scale VOS benchmark characterized by its diverse object categories and a significantly larger volume of video sequences compared to DAVIS17.

MOSEv1 [13] A challenging VOS benchmark focused on complex scenarios such as heavy occlusions, crowded environments, and frequent object disappearances and reappearances.

MOSEv2 [18] An extended version of MOSEv1 featuring more challenging cases, including adverse weather, low-light conditions, camouflaged objects, and scenarios requiring external knowledge.

Table 9: Summary of different benchmarks used in AUV's evaluation.

Benchmark	Year	Task	Prompt Type	Videos (Val. split)
DAVIS17 [39]	2017	VOS	Mask / Point / Box	30
YTVOS19 [50]	2019	VOS	Mask / Point / Box	507
MOSEv1 [13]	2023	VOS	Mask / Point / Box	311
MOSEv2 [18]	2025	VOS	Mask / Point / Box	433
SA-Co/VEval–SA-V [2]	2025	PCS	Language	2k NPs
SA-Co/VEval–YT-Temporal-1B [2]	2025	PCS	Language	1.7k NPs
SA-Co/VEval–SmartGlasses [2]	2025	PCS	Language	2.4k NPs

Table 10: Full comparison of AUV and existing attack baselines on SAM3 [2] across VOS benchmarks and prompt modalities. All baselines use the same training configurations on SAM3 as our AUV.

Methods	DAVIS17 ($\mathcal{J}\&\mathcal{F}\downarrow$)			YTVOS19 ($\mathcal{G}\downarrow$)			MOSEv2 ($\mathcal{J}\&\dot{\mathcal{F}}\downarrow$)			MOSEv1 ($\mathcal{J}\&\mathcal{F}\downarrow$)		
	Mask	Point	Box	Mask	Point	Box	Mask	Point	Box	Mask	Point	Box
Clean	92.2	72.1	89.1	89.7	80.0	86.8	60.3	55.8	58.9	78.4	70.1	77.0
PGD [32]	65.5	21.9	58.8	55.9	27.6	49.9	35.0	28.9	34.3	46.4	33.0	43.9
UAPGD [10]	64.1	21.5	57.4	58.8	29.3	52.9	35.9	29.1	35.0	49.7	35.4	47.8
SegPGD [21]	59.4	17.5	52.4	50.3	22.3	43.8	32.6	27.0	32.3	42.0	28.0	39.8
VOSPGD [24]	61.2	19.8	54.5	49.5	21.8	42.5	35.0	28.1	33.5	44.5	30.5	42.1
AttackSAM [57]	68.3	30.2	63.2	63.0	30.9	53.2	42.6	36.4	38.2	52.5	41.2	47.3
DarkSAM [61]	58.5	25.0	52.5	51.7	23.1	45.6	32.5	26.9	32.0	43.0	28.0	41.3
UAP-SAM2 [60]	55.4	23.9	47.4	46.3	22.0	35.8	30.4	25.5	29.1	36.4	22.5	31.3
AUV (Ours)	**8.4**	**7.1**	**7.6**	**14.0**	**10.3**	**12.7**	**24.6**	**19.9**	**23.5**	**21.1**	**18.3**	**20.2**

Table 11: Full comparison of attack temporal dynamics on SAM3. We omit MOSE because per-frame ground-truth masks are unavailable for its validation set.

Method	DAVIS17						YTVOS19					
	Mask			Point			Mask			Box		
	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$	$Lat\downarrow$	$Cov\uparrow$	$Cov_{20}\uparrow$
PGD [32]	61.3	28.3	34.0	12.3	82.5	82.8	61.2	31.7	38.0	52.7	39.4	47.2
UAPGD [10]	59.0	27.7	35.7	15.1	78.1	79.1	61.2	29.1	33.7	51.4	36.6	40.8
SegPGD [21]	53.7	36.5	45.7	10.0	86.4	87.5	53.9	38.6	46.4	46.6	46.6	55.3
AttackSAM [57]	60.4	29.2	36.1	18.3	75.5	76.7	62.6	29.6	35.7	47.3	44.0	51.1
DarkSAM [61]	58.3	33.5	41.6	11.0	84.6	85.0	56.1	36.6	43.8	46.6	44.8	52.6
UAP-SAM2 [60]	52.5	35.9	43.6	10.4	85.1	85.2	50.9	41.1	47.4	36.1	56.6	63.5
AUV (Ours)	**5.5**	**89.2**	**88.1**	**2.2**	**90.3**	**88.6**	**8.6**	**85.1**	**84.6**	**7.0**	**86.8**	**85.9**

A.2 PCS Benchmarks

SA-Co/VEval [2] A large-scale language-based video segmentation benchmark introduced in SAM3. Given a noun phrase (NP) (e.g., *"white car", "man in jeans"*), the model first detects and then tracks all corresponding targets throughout the entire sequence. This large-scale PCS benchmark comprises three subsets: SA-V (2k NPs), YT-Temporal-1B (1.7k NPs), and SmartGlasses (2.4k NPs).

B Full Results

Due to space constraints in the main paper, we provide full comparisons of different attack methods on SAM3 [2] across prompt modalities and benchmarks here. For a fair comparison, all baselines use **the same** training configurations as ours, as described in Section 4.1.

As shown in Table 10, AUV consistently surpasses existing attack baselines across all VOS benchmarks and prompt types, including mask, point, and box prompts. For example, on DAVIS17-val with point prompts, AUV achieves 7.1 $\mathcal{J}\&\mathcal{F}$, outperforming the previous SOTA UAP-SAM2 [60] of 23.9, even though the latter is specifically optimized for point-based attacks. On more challenging benchmarks such as MOSEv2 and MOSEv1, AUV also achieves the best performance across all prompt types. Notably, on MOSEv1 with mask prompts, AUV outperforms previous SOTA by 15.3 $\mathcal{J}\&\mathcal{F}$.

Also, as discussed in Section 4.2 of the main paper, standard VOS metrics do not capture the temporal dynamics of an attack. Therefore, we further compare AUV with the baselines in terms of attack onset (Lat), persistence (Cov), and mid-video persistence (Cov_k). As shown in Table 11, AUV demonstrates a clear advantage across all temporal metrics: it takes effect much faster, achieves long-lasting suppression, and exhibits robust online attack capability, driven by its propagation-aligned training on memory-based propagation.

For language-based segmentation, AUV achieves the strongest attack performance across all three subsets of SA-Co/VEval, as shown in Table 12. Notably, it reduces cgF1 and pHOTA on YT-Temporal-1B to 0.3 and 9.1, respectively, significantly surpassing the previous SOTA of UAP-SAM2 (3.5 and 26.2, respectively). We further visualize its one-to-many attack capability using language prompts in Section F.

Table 12: Full comparison of AUV and existing attack baselines on SAM3 across **language-based** SA-Co/VEval benchmarks [2]. Lower values are better for all metrics.

Methods	SA-V				YT-Temporal-1B				SmartGlasses			
	cgF1	pHOTA	pDetA	pAssA	cgF1	pHOTA	pDetA	pAssA	cgF1	pHOTA	pDetA	pAssA
Clean	29.3	60.7	44.7	83.2	50.2	70.5	60.5	82.7	33.5	60.2	46.2	79.3
PGD [32]	5.2	33.5	15.1	74.7	13.2	43.7	27.8	69.4	14.7	42.0	26.7	66.9
UAPGD [10]	8.1	38.1	20.3	72.5	17.4	45.1	30.8	66.9	16.5	44.2	30.0	65.9
SegPGD [21]	4.0	31.7	13.5	75.8	11.5	40.2	24.5	66.6	13.6	41.8	26.4	67.0
UAD [31]	2.4	24.9	8.5	74.0	7.8	31.7	15.0	67.5	11.1	36.6	21.0	64.4
DarkSAM [61]	4.0	30.5	12.6	75.0	10.7	38.8	22.7	67.0	12.4	38.8	23.0	65.9
UAP-SAM2 [60]	1.7	23.9	7.8	74.6	3.5	26.2	11.1	62.4	9.1	36.3	21.4	62.4
AUV (Ours)	**0.1**	**8.5**	**1.2**	**64.3**	**0.3**	**9.1**	**1.7**	**49.9**	**1.9**	**18.4**	**6.4**	**53.4**

C Algorithm

Algorithm 1 summarizes the optimization process of AUV, which aligns with the two-stage inference paradigm of modern VOS models through a two-phase curriculum.

In Phase I, we optimize the UAP δ to corrupt the initial memory anchor $\tilde{E}_1$ before it is permanently stored in the memory bank. In Phase II, we jointly optimize the same δ to corrupt the initial memory anchor and flatten the spatial saliency of memory-attended feature maps $\tilde{A}_i$, disrupting both initialization and propagation stages.

D Memory-Level Defenses

Extending Section 4.6, we evaluate two additional memory-level defenses on DAVIS17 using SAM2.1-B+ [40] and its extensions with more advanced memory management. We first apply random memory dropping, discarding 25%, 50%, or 75% of the non-conditioning memory entries at each frame during propagation. We then evaluate different SAM2 extensions, including SAM2Long [19], SAMURAI [51], and DAM4SAM [47], which incorporate their respective memory-management strategies. As shown in Table 13, AUV remains highly effective across different memory-level defenses.

Table 13: Attack performance ($\mathcal{J}\&\mathcal{F}$) under memory-level defenses on SAM2.1-B+.

Defense	**Memory Strategy**	**Mask↓**	**Point↓**	**Box↓**
Random dropping (25%)	Remove non-conditioning entries	15.8	7.6	11.6
Random dropping (50%)	Remove non-conditioning entries	16.4	7.8	11.9
Random dropping (75%)	Remove non-conditioning entries	18.2	8.4	12.7
SAM2Long (SAM2.1-B+)	Memory-tree search	15.9	8.6	11.5
SAMURAI (SAM2.1-B+)	Motion-aware memory selection	20.7	7.8	9.0
DAM4SAM (SAM2.1-B+)	Distractor-aware memory selection	15.8	7.3	14.1

E Further Robustness Analysis

E.1 Prompt Variations

We adopt the standard practice for evaluating point and box prompts in the main paper, following UAP-SAM2 [60] and DarkSAM [61], which uses the center point and tight bounding box derived from the target mask, respectively. To further evaluate the robustness of AUV, we test if prompt variations can successfully bypass the perturbation. Specifically, we evaluate (1) random shifts in point positions, (2) variations in point count, and (3) random translation and scaling of bounding boxes.

As shown in Table 14, AUV maintains consistent attack performance across prompt variations, demonstrating strong robustness to changes in **prompt position, translation, and count**. These results further indicate that AUV generalizes across diverse prompt patterns rather than relying on a specific prompt distribution.

E.2 Stricter Perturbation Bound

To evaluate the robustness of AUV under stricter perturbation bound, we reduce ϵ from the standard $10/255$ [35, 10, 36, 61, 60] to $8/255$, $6/255$, and $4/255$, respectively.

Algorithm 1 AUV: Attack-Inference Aligned Universal Adversarial Perturbation on Video Object Segmentation.

Require: A training video $\mathcal{V}$, VOS model $\Phi = (\Phi_{\text{init}}, \Phi_{\text{prop}})$, memory encoder $\mathcal{E}_{\text{mem}}$, perturbation bound ϵ, step size η, phase epochs $K_{\text{init}}, K_{\text{prop}}$, trade-off λ
Ensure: Universal adversarial perturbation δ
Mask outputs below denote soft values before binarization.
Initialize $\delta \in \mathbb{R}^{H\times W\times 3}$ and project $\delta \leftarrow \Pi_{\|\delta\|_\infty \le \epsilon}(\delta)$
Phase I: Initialization-only optimization
for $e = 1$ to K_{init} **do**
  for video $\mathcal{V} = \{I_1, \dots, I_T\}$ with prompt P **do**
    $M_1^{\text{soft}} \leftarrow \Phi_{\text{init}}(I_1, P)$
    $E_1 \leftarrow \mathcal{E}_{\text{mem}}(I_1, M_1^{\text{soft}}), \quad E_1^{\emptyset} \leftarrow \mathcal{E}_{\text{mem}}(I_1, \mathbf{0})$ ▷ $E_1 \in \mathbb{R}^{h\times w\times c}$
    $\widetilde{M}_1^{\text{soft}} \leftarrow \Phi_{\text{init}}(I_1 + \delta, P)$
    $\tilde{E}_1 \leftarrow \mathcal{E}_{\text{mem}}(I_1 + \delta, \widetilde{M}_1^{\text{soft}})$
    $\mathcal{L}_{\text{init}} \leftarrow \frac{\|\tilde{E}_1 - E_1^{\emptyset}\|_2^2}{hwc} + \exp\Big(-\frac{10\|\tilde{E}_1 - E_1\|_2^2}{hwc}\Big)$
    $\delta \leftarrow \Pi_{\|\delta\|_\infty \le \epsilon}\left(\delta - \eta \nabla_\delta \mathcal{L}_{\text{init}}\right)$
  end for
end for
Phase II: Joint initialization and propagation optimization
for $e = 1$ to K_{prop} **do**
  for video $\mathcal{V} = \{I_1, \dots, I_T\}$ with prompt P **do**
    $M_1^{\text{soft}} \leftarrow \Phi_{\text{init}}(I_1, P)$
    $E_1 \leftarrow \mathcal{E}_{\text{mem}}(I_1, M_1^{\text{soft}}), \quad E_1^{\emptyset} \leftarrow \mathcal{E}_{\text{mem}}(I_1, \mathbf{0})$
    $\widetilde{M}_1^{\text{soft}} \leftarrow \Phi_{\text{init}}(I_1 + \delta, P)$
    $\tilde{E}_1 \leftarrow \mathcal{E}_{\text{mem}}(I_1 + \delta, \widetilde{M}_1^{\text{soft}})$
    $\mathcal{L}_{\text{init}} \leftarrow \frac{\|\tilde{E}_1 - E_1^{\emptyset}\|_2^2}{hwc} + \exp\Big(-\frac{10\|\tilde{E}_1 - E_1\|_2^2}{hwc}\Big)$
    $\mathcal{B} \leftarrow \{\tilde{E}_1\}$
    $\mathcal{L}_{\text{prop}} \leftarrow 0$
    for $i = 2$ to T **do**
      $F_i \leftarrow \mathcal{E}_{\text{img}}(I_i + \delta)$
      $\mathbf{B}_i \leftarrow \text{Read}(\mathcal{B})$
      $\tilde{A}_i \leftarrow \text{softmax}\big(F_i \mathbf{B}_i^\top\big)\, \mathbf{B}_i + F_i$
      $\tilde{\mu}_i \leftarrow \frac{1}{hw}\sum_{x=1}^{h}\sum_{y=1}^{w} \tilde{A}_i(x, y, :)$
      $\mathcal{L}_{\text{prop}} \leftarrow \mathcal{L}_{\text{prop}} + \frac{1}{hwc}\sum_{x=1}^{h}\sum_{y=1}^{w} \Big\|\tilde{A}_i(x, y, :) - \tilde{\mu}_i\Big\|_2^2$
      $\widetilde{M}_i^{\text{soft}} \leftarrow \text{Decoder}(\tilde{A}_i)$ ▷ no prompt input during propagation
      $\tilde{E}_i \leftarrow \mathcal{E}_{\text{mem}}(I_i + \delta, \widetilde{M}_i^{\text{soft}})$
      $\mathcal{B} \leftarrow \text{FIFO}(\mathcal{B}, \tilde{E}_i)$ ▷ always retain the initial anchor $\tilde{E}_1$
    end for
    $\mathcal{L}_{\text{total}} \leftarrow \lambda \mathcal{L}_{\text{init}} + \mathcal{L}_{\text{prop}}$
    $\delta \leftarrow \Pi_{\|\delta\|_\infty \le \epsilon}\left(\delta - \eta \nabla_\delta \mathcal{L}_{\text{total}}\right)$
  end for
end for
return δ

As shown in Table 15, AUV remains effective under smaller perturbation budgets. Remarkably, even at the strictest bound of $4/255$, its performance remains **on par with** the previous SOTA of UAP-SAM2 [60], which uses a much larger perturbation budget of $10/255$. For example, on mask-based YTVOS19, AUV ($4/255$) achieves 41.0 $\mathcal{G}$ compared to 46.3 for UAP-SAM2 ($10/255$), while outperforming the latter on the challenging benchmarks of MOSEv2 and MOSEv1. Furthermore, at $\epsilon = 6/255$, AUV already establishes new SOTA results across different benchmarks compared to the previous SOTA.

F One-to-Many: Multi-Object Video Suppression

We further evaluate AUV in one-to-many scenarios, where a single language prompt specifies multiple objects.

Table 14: Stress test of prompt variations, evaluated on DAVIS17 across various point positions (left), number of points (middle), and bounding box transformations (right).

Position	Point Position		
	$\mathcal{J}\&\mathcal{F}\downarrow$	$\mathcal{J}\downarrow$	$\mathcal{F}\downarrow$
Center	7.1	7.1	7.0
Top-left	6.4	6.2	6.5
Top-right	6.3	6.2	6.3
Bottom-left	6.5	6.3	6.6
Bottom-right	5.3	5.1	5.4

Num.	Point Number		
	$\mathcal{J}\&\mathcal{F}\downarrow$	$\mathcal{J}\downarrow$	$\mathcal{F}\downarrow$
1	7.1	7.1	7.0
2	7.5	7.5	7.5
4	7.4	7.3	7.4
8	7.2	7.2	7.2
16	6.9	6.8	6.9

Position	Box Position		
	$\mathcal{J}\&\mathcal{F}\downarrow$	$\mathcal{J}\downarrow$	$\mathcal{F}\downarrow$
Center	7.6	7.5	7.7
Shift-left	7.7	7.4	7.9
Shift-right	7.6	7.4	7.8
Scale-up	7.7	7.5	7.8
Scale-down	7.3	7.0	7.5

Table 15: Stress test of lowering the perturbation bound ϵ. We evaluate AUV by gradually reducing ϵ from the standard $10/255$ all the way to $4/255$.

Method / Budget ϵ	DAVIS17 ($\mathcal{J}\&\mathcal{F}\downarrow$)			YTVOS19 ($\mathcal{G}\downarrow$)			MOSEv2 ($\mathcal{J}\&\dot{\mathcal{F}}\downarrow$)			MOSEv1 ($\mathcal{J}\&\mathcal{F}\downarrow$)		
	Mask	Point	Box	Mask	Point	Box	Mask	Point	Box	Mask	Point	Box
Clean	92.2	72.1	89.1	89.7	80.0	86.8	60.3	55.8	58.9	78.4	70.1	77.0
UAP-SAM2 ($10/255$) [60]	55.4	23.9	47.4	46.3	22.0	35.8	30.4	25.5	29.1	36.4	22.5	31.3
AUV ($10/255$)	**8.4**	**7.1**	**7.6**	**14.0**	**10.3**	**12.7**	**24.6**	**19.9**	**23.5**	**21.1**	**18.3**	**20.2**
AUV ($8/255$)	12.3	9.1	11.4	17.6	13.2	16.8	23.1	18.4	22.7	24.8	21.3	24.1
AUV ($6/255$)	18.1	14.5	16.4	23.7	16.9	21.0	29.7	25.1	28.1	32.4	26.1	31.0
AUV ($4/255$)	55.6	41.8	51.7	41.0	29.9	37.4	21.0	17.5	20.0	30.3	22.4	30.3



Figure 8: Multi-target video suppression by AUV. It simultaneously eliminates multiple targets specified by a single language prompt, including people in a crowd (top) and eggs (bottom).

As shown in Figure 8, AUV disrupts the segmentation of all specified targets, including people in a crowd (top) and eggs (bottom), causing their predicted masks to disappear immediately. This shows that AUV naturally extends from individual objects to target groups.

G Attack on images

Following [61], we further evaluate the attack performance of AUV on image segmentation. Specifically, we conduct experiments on SAM3 [2] using the validation splits of two image segmentation benchmarks: ADE20K

Table 16: Comparison of AUV and baselines on image segmentation benchmarks.

Settings	ADE20K (mIoU)			RefCOCO			
	Box	3-Points	1-Point	mIoU	Pr@0.5	Pr@0.7	Pr@0.9
Clean	81.4	77.3	46.7	60.9	67.3	61.5	30.4
PGD [32]	76.4	61.3	25.5	45.2	49.2	40.8	15.9
UAPGD [10]	73.7	65.5	23.3	50.4	55.7	46.9	18.2
SEGPGD [21]	74.7	56.9	16.0	44.0	47.6	39.5	15.3
DarkSAM [61]	75.1	59.3	18.7	44.0	47.9	39.1	14.7
UAP-SAM2 [60]	61.9	37.2	8.5	32.1	35.0	27.3	8.3
AUV (Ours)	**15.8**	**11.9**	**7.3**	**12.2**	**12.1**	**9.4**	**2.7**

[59] and RefCOCO [33, 56]. For ADE20K, a standard image segmentation benchmark, we select the first annotated target in each image and use either point or box prompts for interactive segmentation. For RefCOCO, a referring image segmentation benchmark, we directly use the provided language expression as the language prompt.

As shown in Table 16, AUV consistently achieves SOTA attack performance on image segmentation benchmarks, outperforming prior baselines across all settings, including point, box, and language. For instance, on box-based ADE20K, AUV reduces mIoU to 15.8, outperforming the previous SOTA of 61.9. Similarly, on language-based RefCOCO, it achieves an mIoU of 12.2, compared with the previous SOTA of 32.1. These results demonstrate that AUV generalizes effectively to both image and video segmentation attacks.

H Qualitative Comparisons

Finally, we provide qualitative comparisons between AUV and previous baselines, including SegPGD [21], UAD [31], DarkSAM [61], and UAP-SAM2 [60]. Results on clean videos are also provided for reference. We compare the attack performance across different prompt modalities: mask, point, and box, which are used to specify the target on the initial frame. The visual comparisons are shown in Figure 9, 10, 11, respectively.

As shown, AUV consistently outperforms all baselines across different prompt modalities. Specifically, it achieves the fastest onset and the most sustained attack coverage, maintaining strong effect throughout the entire video sequence regardless of the prompts used.

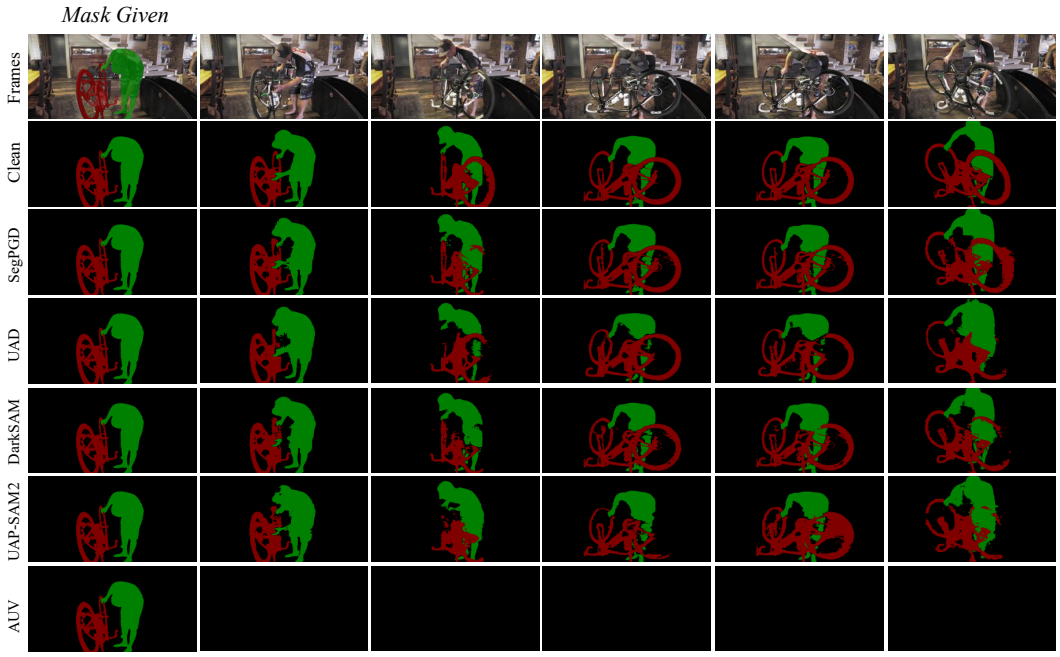


Figure 9: Qualitative comparison under mask prompts.

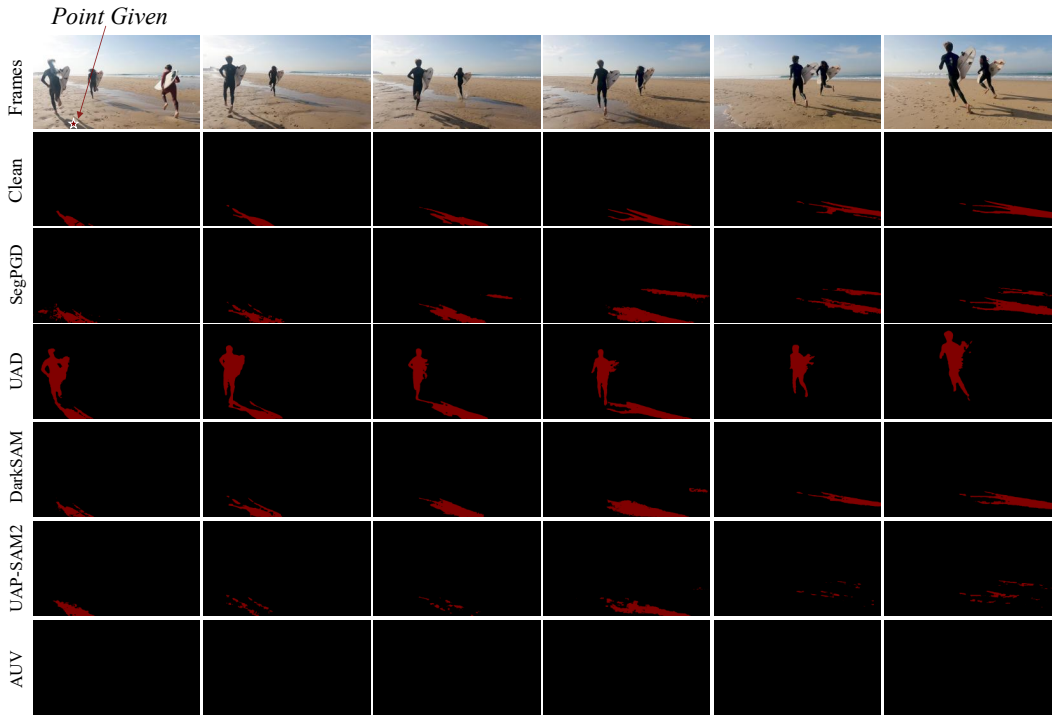


Figure 10: Qualitative comparison under point prompts.

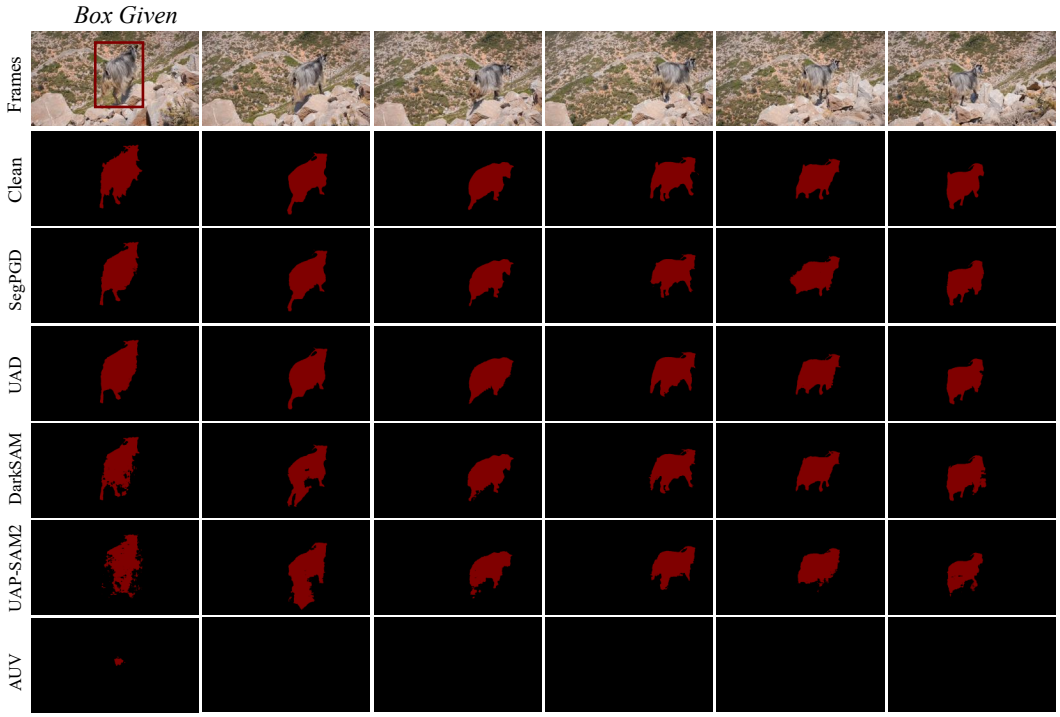


Figure 11: Qualitative comparison under box prompts.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction clearly define AUV's structural alignment with the two-stage VOS paradigm, detailing decoupled strategies for initialization and propagation. These claims are empirically verified through its SOTA and prompt-agnostic attack results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Section E.2 (Appendix) analyzes performance degradation under stricter perturbation bounds as ϵ drops from $10/255$ to $4/255$. Though still matching previous SOTA (which is at $10/255$), its strength is naturally constrained by the stricter budget.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: Section 4.5 (main paper) validates the root of prompt-agnosticism through latent space analysis, using t-SNE visualizations to prove that memory features remain invariant across diverse prompt modalities, even though these prompts have distinct distributions and are processed by different encoding branches.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Section 4.1 (main paper) and Algorithm 1 (Appendix) provide all necessary hyper-parameters, staged training logic, and the complete optimization procedure to ensure experimental reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Code is currently withheld to comply with double-blind review policies but will be open-sourced upon acceptance. Besides, detailed implementations have been given in the Method section (main paper) and Algorithm 1 (Appendix) to ensure reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 4.1 details all experimental settings, including training datasets (strictly align with the previous SOTA), VOS and PCS benchmarks used for evaluation, perturbation bound ϵ, etc.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: We provide extensive statistical data using both standard VOS metrics and newly introduced temporal dynamics (Lat, Cov, Cov_k) to evaluate attack speed and persistence. These results are rigorously presented through tables and linegraphs to ensure a comprehensive analysis to support our claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes].

Justification: Section 4.1 provides details on the GPUs used for training and their corresponding VRAM.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes].

Justification: The research complies with the NeurIPS Code of Ethics. It reveals vulnerabilities in modern VOS models through extensive and rigorous experiments.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes].

Justification: The paper discusses social impacts in the abstract and conclusion by emphasizing its role in promoting a robust benchmark for future research, while highlighting potential security risks to existing streaming applications powered by VOS models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The paper evaluates the vulnerabilities of VOS models and does not release high-risk data or models (e.g. LLMs) with misuse potential.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: All public datasets, including training datasets and evaluation benchmarks, as well as all the models used in the experiments, are clearly credited and cited in the references.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A].

Justification: The paper does not release new assets during double-blind submission. The code and the checkpoint will be released once accepted.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: LLMs were only used for helping grammar and spelling in this paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.