BOOSTING MLLM-SAM2 REFERRING VIDEO OBJECT SEGMENTATION VIA TEMPORAL MEMORY READING

Anonymous authors
Paper under double-blind review

ABSTRACT

Referring Video Object Segmentation (RVOS) aims to segment target objects in a video according to natural language expressions. Recent MLLM-SAM2 methods project MLLM tokens into sparse prompts for SAM2 mask decoding. However, they suffer from *object identification ambiguity* when distinguishing visually similar instances via temporal cues. Through probing analysis, we trace this failure to the prompt interface: frame-order decodability degrades significantly during token-to-prompt conversion, yielding collapsed prompt representations that confuse same-category instances. We propose **Temporal Memory Reading (TMR)**, which grounds expression-conditioned temporal queries in multi-frame SAM2 features and injects spatially adaptive query biases into memory attention throughout propagation. Experiments on multiple benchmarks show that TMR improves MLLM-SAM2 baselines on temporally ambiguous cases with similar distractors.

1 INTRODUCTION

Referring Video Object Segmentation (RVOS) aims to segment target objects in a video sequence according to a natural language expression. This task extends Referring Image Segmentation(RIS) from static images to videos, and requires the model to handle temporal variations across frames.

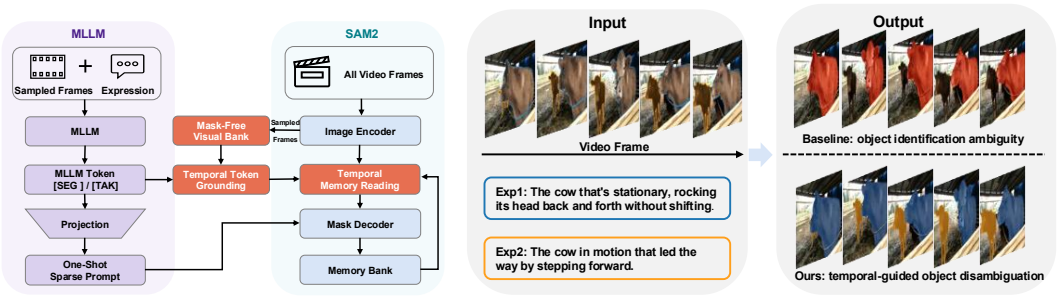
Recent RVOS methods(Yuan et al., 2025; Gong et al., 2025; Yan et al., 2024; Bai et al., 2024) increasingly build on Multimodal Large Language Model (MLLM)(Chen et al., 2023; Wang et al., 2024; Jin et al., 2024), forming a unified pipeline as shown in Fig.1(a). In this paradigm, MLLM first jointly encodes the sampled frames and expression, after which the hidden state of a special segmentation token (e.g., `[SEG]`, `[TAK]`) is projected into a sparse prompt to guide a SAM-style(Kirillov et al., 2023; Ravi et al., 2024; Carion et al., 2026) mask decoder. This simple yet effective design has delivered promising results.

However, the current paradigm tends to fail when the referred target must be distinguished from visually similar instances through temporal cues. In Fig. 1(b), two expressions identify different cows, yet the baseline predicts nearly the same mask for both. We call this failure *object identification ambiguity*. It is not limited to a qualitative example: among 327 within-video expression pairs referring to different same-category targets, 168 (51.4%) yield nearly identical masks under VRS-HQ-7B(Gong et al., 2025).

A natural explanation for these failures is that the MLLM itself lacks sufficient temporal reasoning. However, our probing analysis identifies that this temporal bottleneck primarily emerges across prompt construction. While the high-dimensional MLLM state reliably encodes frame-order information, decodability drops precipitously in the final 256-d SAM2 prompt. Strikingly, an unlearned random projection to the same dimensionality preserves significantly higher decodability than the baseline interface, and the resulting prompt-space similarity strongly correlates with same-category mask collapse. While these findings do not preclude upstream reasoning limitations, they demonstrate that the prompt interface actively discards critical temporal cues. Moreover, the one-shot prompt conditions initialization, whereas later predictions rely on SAM2 memory propagation and can inherit an incorrect initial selection.

To address this bottleneck, we propose **Temporal Memory Reading (TMR)**. While TMR retains the standard SAM2 prompt for mask initialization, it moves beyond relying solely on this one-shot

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) MLLM-SAM2 pipeline with TMR **(b) Object identification ambiguity**

Figure 1: (a) Standard MLLM-SAM2 pipeline (■ MLLM, ■ SAM2) relies on a one-shot sparse prompt used only at the anchor frame. TMR adds three components highlighted in ■ orange: a *Mask-Free Visual Bank*, *Temporal Token Grounding*, and *Temporal Memory Reading*. Together they continuously supply temporal cues to SAM2 memory attention. (b) On a temporally ambiguous example, the baseline collapses both expressions onto the same salient cow, while TMR disambiguates the referred target using temporal cues.

prompt as the exclusive carrier of referring information. The grounded tokens are then mapped into the SAM2 memory space to guide memory reading. At the anchor frame, TMR uses the same Mask-Free Visual Bank as the source of memory entries, thereby mitigating reliance on the one-shot prompt. For subsequent frames, it performs per-position retrieval on the grounded temporal tokens and injects them as a spatially-adaptive query bias prior to SAM2 memory attention. Consequently, temporal cues crucial for object disambiguation remain continuously accessible at every frame and at every spatial position, effectively preventing the amplification of early prompt biases.

Experiments connect this design to the diagnosed failure. On MeViS val^u, matched fine-tuning, multi learned sparse prompts, and repeating the original prompt at every frame score 56.8, 57.1, and 55.6 $\mathcal{J}\&\mathcal{F}$, versus 62.1 for TMR. TMR reduces collapse to 75/327 pairs; 87 of the original 168 collapse pairs both separate and improve pair-averaged target-specific segmentation. On the reasoning-oriented ReVOS benchmark, TMR achieves consistent gains across model scales, with the 14B model obtaining the best overall performance of 66.9% $\mathcal{J}\&\mathcal{F}$.

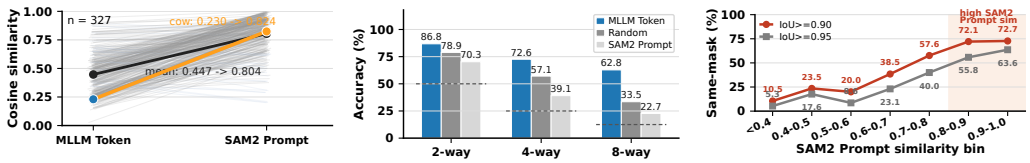
Our contributions are:

- We diagnose reduced frame-order decodability across the MLLM-SAM2 prompt interface and quantify the association between prompt similarity and same-category mask collapse.
- We introduce **Temporal Memory Reading (TMR)** to ensure object disambiguation. TMR grounds MLLM temporal tokens with frame features and utilizes them as query biases, providing continuous temporal guidance throughout propagation.
- We provide extensive experiments showing that TMR alleviates object identification ambiguity and improves RVOS performance under ambiguity and similar distractors.

2 Related Work

MLLM-based referring segmentation. LISA (Lai & others, 2023) projects a `[SEG]` token into SAM's prompt space, while LLM-Seg (Wang & Ke, 2024) selects among SAM proposals. VISA (Yan et al., 2024) and VideoLISA (Bai et al., 2024) extend language-conditioned mask prediction to videos. With SAM2 (Ravi et al., 2024), Sa2VA (Yuan et al., 2025) unifies image/video understanding and segmentation; VRS-HQ (Gong et al., 2025) aggregates frame-level `[SEG]` tokens into a video-level `[TAK]` prompt; and ViLLa (Zheng et al., 2025) improves video reasoning through hierarchical temporal modeling. TMR investigates how information already encoded by such hosts is delivered to the segmentation pathway.

Temporal conditioning and related interventions. SAMWISE (Cuttano et al., 2025) addresses the closely related behavior of SAM2 continuing to track an incorrect object through temporal feature adaptation and conditional track correction. GLUS (Lin et al., 2025) unifies global reasoning and local



(a) Object-similarity probe **(b) Temporal-order probe** **(c) Prompt-to-mask probe**

Figure 2: Probing the MLLM-SAM2 interface in VRS-HQ-7B. *MLLM Token* denotes the high-dimensional video-level representation before prompt construction, and *SAM2 Prompt* denotes the final fused 256-d sparse prompt fed to SAM2. (a) Expressions referring to different same-category instances become more similar in the SAM2 prompt space. (b) Temporal-order cues are more separable in the MLLM token space than in the SAM2 prompt space. (c) Higher SAM2 prompt similarity is associated with overlapping predictions for different targets.

tracking. SVAC (Zhang et al., 2025) scales input frames and clip-specific segmentation tokens. TMR grounds the pre-projection host state in SAM2 features and conditions memory-attention queries at each spatial position. Its contribution is the diagnosis and the resulting grounding-and-memory-reading design. Our eight-prompt control evaluates a specific anchor-prompt expansion; it does not reproduce SVAC.

Reasoning and inference protocols. Veason-R1 (Gong et al., 2026), VideoSeg-R1 (Xu et al., 2026), and ReVSeg (Li et al., 2025) improve reasoning and localization with reinforcement learning. Their strong results motivate evaluating TMR as a host augmentation. Sa2VA-i (Nekrasov et al., 2025) identifies training–inference inconsistencies in Sa2VA. Such protocol effects can coexist with the reduced probe accuracy observed here. We study this interface-level weakness and evaluate TMR with a matched fine-tuning control under the same training mixture.

3 Preliminary Analysis

We probe temporal separability in VRS-HQ-7B (Gong et al., 2025). Because preserving temporal ordering serves as the fundamental structural backbone for dynamic temporal cues, we operationalize this evaluation via frame-order decodability. Specifically, its MLLM emits per-frame ⟨[SEG]⟩ tokens and a video-level [TAK] token at width 4096. Their 256-d projections are fused by Temporal Dynamic Aggregation (TDA). We compare the pre-projection [TAK] state (*MLLM Token*) with the final fused prompt (*SAM2 Prompt*) to examine where such temporal cues degrade. This comparison covers prompt construction, including projection and TDA, without isolating their individual effects.

Same-category target prompts become more similar. We collect 327 expression pairs from the same video, where the two expressions refer to different same-category objects. As shown in Fig. 2(a), the mean cosine similarity within each pair increases from 0.447 in the MLLM Token space to 0.804 in the SAM2 Prompt space. In total, 91.1% of pairs have higher cosine similarity in the final prompt space. The cow case in our teaser shows the same trend, with similarity increasing from 0.230 to 0.824. This is consistent with reduced target separability, although cross-space cosine similarity alone does not measure information loss.

Temporal cues remain before prompt construction. We use a logistic-regression probe to predict which of K frame permutations is given to the MLLM (Details are in Appendix A). Fig. 2(b) shows that the MLLM Token reaches 86.8%/72.6%/62.8% for K=2/4/8, while the SAM2 Prompt drops to 70.3%/39.1%/22.7%. As a dimensionality control, we project the same MLLM Token to 256 dimensions with a random Gaussian matrix. This random control reaches 78.9%/57.1%/33.5%, exceeding the trained SAM2 Prompt by 8.6, 18.0, and 10.8 points. The larger accuracy drop for the trained prompt cannot be attributed to output dimensionality alone. This probe measures frame-order decodability, not whether the host correctly reasons about the referred target.

Prompt similarity is associated with mask collapse. We bin the 327 pairs by SAM2 Prompt similarity and measure the per-frame IoU between their predicted masks. Fig. 2(c) shows that the same-mask rate generally increases across the seven bins, with full results in Appendix Table 9. When prompt similarity is below 0.4, only 10.5% of pairs reach mask IoU $\geq$ 0.9, and 5.3% reach $\geq$ 0.95.

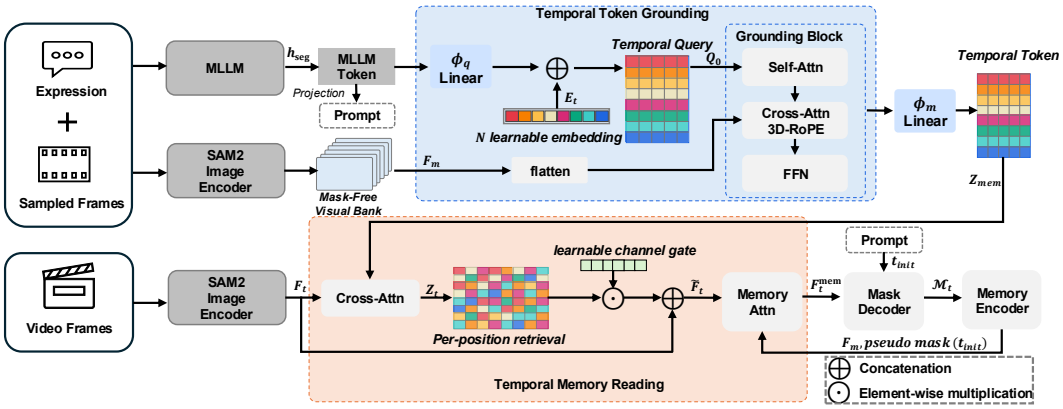


Figure 3: Overview of our method. **Temporal Token Grounding** expands h_{seg} into N temporal queries and grounds them against the Mask-Free Visual Bank to produce temporal tokens Z_{mem}. **Temporal Memory Reading** retrieves a per-position Z_t from Z_{mem} and injects it as a query bias into F_t before SAM2 memory attention.

When prompt similarity is between 0.9 and 1.0, these rates rise to 72.7% and 63.6%. This associates prompt similarity with *object identification ambiguity*, motivating a complementary conditioning path.

Implication and scope. Across the 327 pairs, 168 (51.4%) have an average prediction-to-prediction mask IoU of at least 0.9 over eight sampled frames. This rate describes the selected same-category pair set, not the full benchmark error rate. Together, the probes identify reduced temporal separability and an association with mask collapse; they do not establish a causal mechanism. Our working hypothesis is that grounding the pre-projection state in visual evidence and supplying it during memory reading can alleviate this ambiguity. Section 5.3 tests this intervention through controls and case-level analysis.

4 Method

4.1 Overview

Given a video $\mathcal{V} = \{I_t\}_{t=1}^{T}$ and a referring expression e, the host MLLM takes M sampled frames $\mathcal{S} = \{I_{\tau_m}\}_{m=1}^{M}$ as input and outputs a segmentation-token hidden state $h_{\text{seg}} \in \mathbb{R}^{d_m}$. The baseline projector maps h_{seg} to a 256-d sparse prompt, which is used only at the anchor frame for SAM2 mask initialization. TMR keeps this path unchanged. In parallel, it builds a Mask-Free Visual Bank from the sampled frames (Section 4.2), grounds temporal query tokens derived from h_{seg} with this bank (Section 4.3), and uses them to guide SAM2 memory reading at each propagation frame (Section 4.4). The overall framework is shown in Fig. 3.

4.2 Mask-Free Visual Bank

The Mask-Free Visual Bank (MFVB) is a shared component used by both stages of TMR. We pass each of the M MLLM-sampled frames through the (frozen) SAM2 image encoder and keep the top-level feature map at full spatial resolution:

$$F_m = \text{Enc}^{\text{img}}_{\text{SAM2}}(I_{\tau_m}) \in \mathbb{R}^{d_{\text{mem}} \times H \times W}, \quad m = 1, \ldots, M, \tag{1}$$

with $d_{\text{mem}} = 256$ the SAM2 memory channel width. We reuse SAM2 image features computed by the segmentation pipeline. The bank stores only $M{=}8$ sampled frames, not all video frames; its storage is fixed with respect to video length at a fixed feature resolution. We call the bank *mask-free* because no SAM prompt, predicted mask, GT mask, or propagated memory is used to construct it. The bank plays two roles: a grounding target (Section 4.3) and a source of anchor-frame memory entries (Section 4.4).

4.3 TEMPORAL TOKEN GROUNDING

Rather than compress h_{seg} into a single 256-d prompt, we generate N temporal queries at an intermediate width d_t=512, ground them against the MFVB using transformer decoder blocks, and compress to the SAM2 width only afterwards.

Temporal queries. N learnable embeddings $E_t \in \mathbb{R}^{N\times d_t}$ are added to a linear projection of h_{seg}:

$$Q_0[i] = \phi_q(h_{\text{seg}}) + E_t[i] \in \mathbb{R}^{d_t}, \quad i = 1, \ldots, N, \tag{2}$$

where $\phi_q : \mathbb{R}^{d_m} \to \mathbb{R}^{d_t}$. Each query is conditioned on the current video and expression through h_{seg}; $E_t[i]$ distinguishes query slots rather than assigning a fixed temporal concept to each slot.

Grounding block. Each transformer decoder block grounds the temporal queries against the MFVB and consists of self-attention, cross-attention, and a feed-forward network, all equipped with pre-LayerNorm and residual connections at a hidden dimension of d_t:

$$Q_1 = Q_0 + \text{SelfAttn}(\text{LN}(Q_0)), \tag{3}$$

$$Q_2 = Q_1 + \text{CrossAttn}_{\text{3D-RoPE}}(\text{LN}(Q_1),\ \{F_m\}_{m=1}^{M}), \tag{4}$$

$$Z_g = Q_2 + \text{FFN}(\text{LN}(Q_2)) \in \mathbb{R}^{N\times d_t}. \tag{5}$$

The cross-attention treats the MFVB as a sequence of $M \cdot H \cdot W$ key-value tokens at width d_{mem}, projecting them via W_K and W_V to the head dim; 3D-RoPE is applied to the keys after W_K using each entry's coordinates (m, h, w), while the temporal queries remain unrotated since they carry no intrinsic spatial-temporal coordinates. This encoding retains frame order and spatial location after the bank is flattened.

Finally, a learnable linear map $\phi_m : \mathbb{R}^{d_t} \to \mathbb{R}^{d_{\text{mem}}}$ projects Z_g to the SAM2 memory width:

$$Z_{\text{mem}} = \phi_m(Z_g) \in \mathbb{R}^{N\times d_{\text{mem}}}. \tag{6}$$

Z_{mem} is computed once per video-expression pair and shared across all propagation frames. Keeping ϕ_m external to the block makes the design choice explicit: grounding operates entirely at d_t, and compression happens only at the SAM2 interface.

4.4 TEMPORAL MEMORY READING

Per-position retrieval. At every propagation frame t, we use the current SAM2 feature map $F_t \in \mathbb{R}^{d_{\text{mem}}\times H\times W}$ at full spatial resolution as the query for a single cross-attention into Z_{mem}:

$$Z_t = \text{CrossAttn}(F_t,\ Z_{\text{mem}},\ Z_{\text{mem}}) \in \mathbb{R}^{(HW)\times d_{\text{mem}}}. \tag{7}$$

Each spatial position retrieves its own mixture of the N temporal tokens. The readout varies across frames and across positions within a frame.

Query bias injection. Z_t is mixed into the SAM2 query feature through a learnable per-channel gate:

$$\widetilde{F}_t = F_t + \mathbf{g} \odot Z_t, \quad \mathbf{g} = \sigma(\boldsymbol{\alpha}), \qquad F_t^{\text{mem}} = \text{MemAttn}_{\text{SAM2}}(\widetilde{F}_t,\ \mathcal{M}_t), \tag{8}$$

where σ is the logistic sigmoid, $\odot$ denotes element-wise multiplication, $\boldsymbol{\alpha} \in \mathbb{R}^{d_{\text{mem}}}$ is a learnable per-channel parameter, and $\mathcal{M}_t$ is the SAM2 memory bank consumed at frame t. The gate $\mathbf{g} \in (0, 1)^{d_{\text{mem}}}$ is broadcast over the spatial dimension and scales each channel of the grounded readout before addition. SAM2 memory-attention internals remain unchanged, with anchor-frame memory construction described below.

Anchor frame memory bank. At the anchor frame t_{init}, no real predicted-mask memory has yet been written, so standard SAM2 reads from memory entries seeded by the one-shot prompt. We replace these entries with MFVB-derived ones via the SAM2 memory encoder and a fixed, expression-agnostic pseudo mask P:

$$\mathcal{M}_{t_{\text{init}}} = \left\{\text{Enc}_{\text{SAM2}}^{\text{mem}}(F_m,\ P)\right\}_{m:\ \tau_m \neq t_{\text{init}}}. \tag{9}$$

We use an all-foreground constant for P by default. At all other propagation frames, $\mathcal{M}_t$ is the predicted-mask memory bank built in the standard SAM2 way. Only at the anchor frame does

the MFVB-derived bank provide a second evidence source that does not pass through the one-shot prompt.

Mask decoding and write-back. The SAM2 mask decoder turns F_t^{mem} into the mask logits $\hat{y}_t$. At t_{init} the decoder also consumes the baseline SAM prompt. The predicted mask is then encoded by the SAM2 memory encoder and added to the memory bank for later frames. Both steps remain unchanged from SAM2.

4.5 Training

We use the publicly released MLLM weights of the host MLLM-SAM2 RVOS baseline and keep them frozen. The SAM2 image encoder is also frozen. The remaining SAM2 modules (memory encoder, memory attention, mask decoder) are fine-tuned together with the TMR modules. The objective is a per-frame BCE-plus-Dice segmentation loss with a token-diversity regularizer:

$$\mathcal{L} = \mathcal{L}_{\mathrm{seg}} + \lambda_{\mathrm{div}}\,\mathcal{L}_{\mathrm{div}}, \quad \mathcal{L}_{\mathrm{div}} = \frac{1}{N(N-1)} \sum_{i \neq j} \left(\frac{Z_g^{(i)} \cdot Z_g^{(j)}}{\|Z_g^{(i)}\|\,\|Z_g^{(j)}\|} \right)^2. \tag{10}$$

$\mathcal{L}_{\mathrm{div}}$ penalizes squared cosine similarity between temporal queries to encourage diverse directions.

5 Experiments

5.1 Experimental Setup

Datasets and metrics. We evaluate on MeViS (Ding et al., 2023), Ref-YouTube-VOS (Seo et al., 2020), Ref-DAVIS17 (Khoreva et al., 2019), and ReVOS (Yan et al., 2024). We distinguish MeViS full val from val^u, which is used for diagnostic analyses and ablations. Metrics are region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and their average $\mathcal{J}\&\mathcal{F}$; gains below are absolute points.

Hosts and training controls. TMR augments Sa2VA-2B/8B/14B (Yuan et al., 2025) and VRS-HQ-7B (Gong et al., 2025). We freeze the host MLLM, baseline prompt projector, and SAM2 image encoder, and train TMR with the SAM2 memory encoder, memory attention, and mask decoder. Defaults are $M{=}8$, $N{=}8$, and $d_t{=}512$. For VRS-HQ-7B, a matched fine-tuned baseline uses the same five-source training mixture, sample counts, optimizer, schedule, and trainable SAM2 modules, but removes TMR. Prompt dropout is disabled for prompt-only controls to avoid weakening their only language-conditioning path. Appendix C gives the full recipe.

5.2 Quantitative Results

Host augmentation and reported comparisons. Tables 1 and 2 show positive $\mathcal{J}\&\mathcal{F}$ gains over each corresponding original host across the four benchmarks. The 14B configuration reaches 61.2 on MeViS val and 66.9 on ReVOS. Gains are uneven: the 8B Ref-DAVIS17 improvement is only 0.2, whereas the 2B Ref-YT-VOS improvement is 9.1. The extended comparison (Table 13) includes VideoSeg-R1 and ReVSeg, which are stronger on Ref-YT-VOS and/or Ref-DAVIS17, so we do not claim universal state-of-the-art performance. With the same InternVL2.5-8B backbone and MeViS val^u split, TMR scores 64.0 versus SVAC's 62.8; this is a reported comparison, not a controlled architectural comparison, because their frame budgets, MLLM tuning, and training data differ.

Separating TMR from additional fine-tuning. Matched fine-tuning (Table 13) changes the VRS-HQ-7B scores by at most 0.3 points. Relative to that control, TMR gains 5.3/5.7 on MeViS val^u/val, 2.9 on Ref-YT-VOS, 1.0 on Ref-DAVIS17, and 5.8 on ReVOS. Additional fine-tuning under this mixture therefore does not explain the gains. This control does not establish robustness to changes in the training mixture.

Paired-seed stability. We train the matched VRS-HQ-7B baseline and TMR with seeds 42, 43, and 44, pairing the data order and training settings. Mean paired gains (sample standard deviation) are $5.33{\pm}0.15$ on MeViS val^u and $1.00{\pm}0.10$ on Ref-DAVIS17. Two-sided 95% Student-t confidence intervals over the three paired differences are $[4.95, 5.71]$ and $[0.75, 1.25]$. Thus, the smaller Ref-DAVIS17 gain is reproducible across these runs. These intervals quantify training-seed variability on fixed evaluation sets; per-seed results are in Appendix D.3.

Table 1: Comparison on MeViS val, Ref-YouTube-VOS, and Ref-DAVIS17. †: reproduced host checkpoints. Parentheses show gains over the corresponding host. Bold/underline mark best/second-best values among listed methods. Extended comparisons and matched fine-tuning are in Table 13.

Method	Backbone	MeViS val			Ref-YouTube-VOS			Ref-DAVIS17		
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
Segmentation Experts (SAM2)										
SAMWISE(Cuttano et al., 2025)	CLIP-B	45.4	51.2	48.3	65.2	69.3	67.2	65.6	71.5	68.5
MPG-SAM2(Rong et al., 2025)	BEiT-3-L	50.7	56.7	53.7	71.7	76.1	73.9	68.8	76.0	72.4
Grounded-SAM2(IDEA-Research, 2025)	BERT	34.9	40.5	37.7	55.6	59.5	57.5	62.8	69.9	66.4
MLLM-SAM Segmentation Models										
LISA-7B(Lai & others, 2023)	LLaVA-7B	35.1	39.4	37.2	53.4	54.3	53.9	62.2	67.3	64.8
VISA-7B(Yan et al., 2024)	Chat-UniVi-7B	40.7	46.3	43.5	59.8	63.2	61.5	66.3	72.5	69.4
VISA-13B(Yan et al., 2024)	Chat-UniVi-13B	41.8	47.1	44.5	61.4	64.7	63.0	67.0	73.8	70.4
VideoLISA(Bai et al., 2024)	LLaVA-Phi-3-V	41.3	47.6	44.4	61.7	65.7	63.7	64.9	72.7	68.8
MLLM-SAM2 Segmentation Models										
ViLLa(Zheng et al., 2025)	InternVideo2-6B	46.5	52.3	49.4	64.6	70.4	67.5	70.6	78.0	74.3
SA2VA-2B † (Yuan et al., 2025)	Qwen3-VL-2B	41.3	48.0	44.7	61.3	64.6	62.9	67.6	76.9	72.2
Ours-2B	Qwen3-VL-2B	45.0 (+3.7)	52.2 (+4.2)	48.6 (+3.9)	70.3 (+9.0)	73.6 (+9.0)	72.0 (+9.1)	71.9 (+4.3)	79.5 (+2.6)	75.7 (+3.5)
VRS-HQ-7B †(Gong et al., 2025)	Chat-UniVi-7B	47.7	53.7	50.7	68.5	72.4	70.4	72.3	79.1	76.0
Ours-7B	Chat-UniVi-7B	53.3 (+5.6)	59.5 (+5.8)	56.4 (+5.7)	68.7 (+0.2)	77.5 (+5.1)	73.1 (+2.7)	72.8 (+0.5)	81.0 (+1.9)	76.9 (+0.9)
SA2VA-8B † (Yuan et al., 2025)	InternVL2.5-8B	51.8	58.5	55.1	70.1	74.6	72.4	72.1	81.1	76.6
Ours-8B	InternVL2.5-8B	55.1 (+3.3)	62.3 (+3.8)	58.7 (+3.6)	71.0 (+0.9)	78.1 (+3.5)	74.6 (+2.2)	**73.7** (+1.6)	79.9 (-1.2)	76.8 (+0.2)
SA2VA-14B † (Yuan et al., 2025)	InternVL3-14B	51.5	58.2	54.9	68.4	72.2	70.5	71.4	80.9	76.1
Ours-14B	InternVL3-14B	**57.7** (+6.2)	**64.6** (+6.4)	**61.2** (+6.3)	**71.9** (+3.5)	**78.3** (+6.1)	**75.1** (+4.6)	72.2 (+0.8)	**82.3** (+1.4)	**77.3** (+1.2)

Table 2: Comparison on ReVOS. †: reproduced host checkpoints. Parentheses show gains over the corresponding host. Bold/underline mark best/second-best values among listed methods. Extended comparisons and matched fine-tuning are in Table 13.

Method	Backbone	Referring			Reasoning			Overall		
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
MLLM-SAM Segmentation Models										
LISA-7B(Lai & others, 2023)	LLaVA-7B	44.3	47.1	45.7	33.8	38.4	36.1	39.1	42.7	40.9
VISA-7B(Yan et al., 2024)	Chat-UniVi-7B	49.2	52.6	50.9	40.6	45.4	43.0	44.9	49.0	46.9
VISA-13B(Yan et al., 2024)	Chat-UniVi-13B	55.6	59.1	57.4	42.0	46.7	44.3	48.8	52.9	50.9
MLLM-SAM2 Segmentation Models										
RGA-3B(Wang et al., 2025)	Qwen2.5-VL-3B	57.6	61.0	59.3	50.6	55.0	52.8	54.1	58.0	56.1
RGA-7B(Wang et al., 2025)	Qwen2.5-VL-7B	58.7	62.3	60.5	53.1	57.7	55.4	55.9	60.0	58.0
SA2VA-2B † (Yuan et al., 2025)	Qwen3-VL-2B	55.4	60.5	57.9	44.3	50.6	47.5	49.9	55.5	52.7
Ours-2B	Qwen3-VL-2B	55.8 (+0.4)	62.4 (+1.9)	59.1 (+1.2)	50.9 (+6.6)	57.1 (+6.5)	54.0 (+6.5)	53.4 (+3.5)	59.8 (+4.3)	56.6 (+3.9)
VRS-HQ-7B † (Gong et al., 2025)	Chat-UniVi-7B	59.8	64.5	62.1	53.5	58.7	56.1	56.6	61.6	59.1
Ours-7B	Chat-UniVi-7B	64.8 (+5.0)	69.4 (+4.9)	67.1 (+5.0)	**60.0** (+6.5)	**64.3** (+5.6)	**62.1** (+6.0)	62.4 (+5.8)	66.9 (+5.3)	64.6 (+5.5)
SA2VA-8B † (Yuan et al., 2025)	InternVL2.5-8B	62.4	68.0	65.2	55.9	62.4	59.1	59.2	65.2	62.2
Ours-8B	InternVL2.5-8B	**66.1** (+3.7)	**71.1** (+3.1)	**68.6** (+3.4)	58.8 (+2.9)	62.9 (+0.5)	60.8 (+1.7)	**62.5** (+3.3)	**67.0** (+1.8)	**64.8** (+2.6)
SA2VA-14B† (Yuan et al., 2025)	InternVL3-14B	62.5	68.1	65.3	56.4	63.0	59.7	59.5	65.5	62.5
Ours-14B	InternVL3-14B	**66.7** (+4.2)	**72.8** (+4.7)	**69.8** (+4.5)	59.5 (+3.1)	**68.4** (+5.4)	**64.0** (+4.3)	**63.1** (+3.6)	**70.6** (+5.1)	**66.9** (+4.4)

5.3 Ablations and Diagnostic Validation

All ablations use VRS-HQ-7B on MeViS val^u.

Component contributions. Table 3 shows that ungrounded injection reduces $\mathcal{J}\&\mathcal{F}$ from 56.6 to 47.8. Grounding reaches 57.6; adding per-position memory reading raises it to 60.8, and MFVB-derived anchor memory further improves it to 62.1. This supports grounding the host state in visual evidence and supplying it during propagation.

Capacity and repeated-prompt alternatives. Table 4 tests simpler explanations. Expanding h_{seg} into eight jointly trained 256-d anchor-frame prompts yields 57.1 overall and 43.8 on temporal expressions, versus 62.1 and 52.2 for TMR, despite a larger added parameter count (25.2M versus 15.49M). Re-applying the same baseline prompt at every propagation frame yields 55.6. These controls do not reproduce TMR's gains; they test this prompt expansion and repetition, not every possible multi-token or per-frame method. Wider/deeper single-prompt generators also give at most 0.5 points (Table 8).

Where the gains occur. Table 5 reports gains of 8.6 on Temporal expressions, 2.7 on Appearance expressions, 8.6 on same-category different-target pairs, and 9.4 on baseline-collapse pairs. Beyond

Table 3: Main component ablation on MeViS val^u. Init. MFVB denotes the use of the Mask-Free Visual Bank as the anchor-frame memory bank, in place of prompt-induced entries; it is only active at the anchor frame.

Variant	Temporal Token	Temporal Token Grounding	Temporal Memory Reading	Init. MFVB	J	F	$J\&F$
Baseline	–	–	–	–	55.2	58.0	56.6
+ Ungrounded Temporal Token	✓	–	–	–	44.5	51.1	47.8
+ Temporal Token Grounding	✓	✓	–	–	55.8	59.4	57.6
+ Temporal Memory Reading	✓	✓	✓	–	59.3	62.3	60.8
Ours	✓	✓	✓	✓	**60.4**	**63.8**	**62.1**

Table 4: Simple controls on MeViS val^u. Eight-prompt tokens are trained with the same mixture; repeated prompting reuses the baseline prompt at each frame. "–" denotes an unreported value.

Variant	Extra params	Overall	Temporal
Original baseline	–	56.6	43.6
Matched fine-tuning	–	56.8	–
Eight sparse prompts	25.2M	57.1	43.8
Repeated prompt	–	55.6	–
TMR	15.49M	**62.1**	**52.2**

Table 5: Overall and diagnostic-subset $\mathcal{J}\&\mathcal{F}$ on MeViS val^u. App./Temp.: Appearance/Temporal expressions; Same-cat: different same-category targets; Collapse: the fixed 168 baseline-collapse pairs.

Method	Overall	App.	Temp.	Same-cat	Collapse
Baseline	56.6	68.1	43.6	44.8	38.2
+ Grounding	57.6	68.2	45.6	46.0	39.3
+ Memory reading	60.8	69.7	50.7	51.8	45.8
TMR	**62.1**	**70.8**	**52.2**	**53.4**	**47.6**

Table 6: Case-level analysis of the 327 same-category pairs. Separation requires prediction-to-prediction mask IoU to fall below 0.9; improvement additionally requires higher target-specific pair-averaged $\mathcal{J}\&\mathcal{F}$.

Metric	Result
Collapse pairs, baseline	168/327 (51.4%)
Collapse pairs, TMR	75/327 (22.9%)
Original collapses separated	103/168 (61.3%)
Separated and improved	87/168 (51.8%)
New collapses	10/159 (6.3%)
Original-collapse $\mathcal{J}\&\mathcal{F}$	$38.2 \rightarrow 47.6$

Table 7: Design ablations on MeViS val^u. Each variant changes one component or hyperparameter of TMR with VRS-HQ-7B.

Variant	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
(i) Token diversity loss $\mathcal{L}_{\text{div}}$			
Ours (with $\mathcal{L}_{\text{div}}$)	**60.4**	**63.8**	**62.1**
w/o $\mathcal{L}_{\text{div}}$	58.5	61.9	60.2
(ii) MFVB positional encoding			
3D-RoPE (Ours)	**60.4**	**63.8**	**62.1**
w/o positional encoding	55.7	59.7	57.7
1D temporal RoPE	58.0	61.5	59.8
(iii) Per-position vs. global retrieval			
Per-position retrieval (Ours)	**60.4**	**63.8**	**62.1**
Single global query (avg-pool F_t)	57.2	60.6	58.9
(iv) Number of temporal tokens N			
$N = 4$	59.1	62.3	60.7
$N = 8$ (Ours)	**60.4**	**63.8**	**62.1**
$N = 16$	60.2	63.0	61.6
(v) Anchor-frame pseudo mask			
all-foreground (Ours)	**60.4**	**63.8**	**62.1**
centered Gaussian	57.4	61.1	59.3

Table 8: Prompt-generator capacity and TMR width d_t on MeViS val^u. Same-cat and Collapse follow Table 5.

Variant	Extra Params	Overall J&F	Same-cat	Collapse
Baseline	–	56.6	44.8	38.2
Wider Prompt Gen.	+17.83M	56.9	45.4	38.8
Deeper Prompt Gen.	+33.56M	57.1	45.7	39.2
d_t=256	+6.24M	59.4	49.5	43.1
d_t=512(Ours)	+15.49M	**62.1**	**53.4**	**47.6**
d_t=1024	+43.83M	62.0	53.2	47.4

grounding alone, memory reading adds 5.1 on Temporal versus 1.5 on Appearance expressions. With 372 Temporal and 421 Appearance expressions, approximately 74% of the aggregate gain comes from the Temporal subset, consistent with the diagnosis.

Case-level collapse analysis. Table 6 re-analyzes the same 327 pairs. TMR reduces collapse from 168 to 75 pairs. Of the original 168, 103 separate, and 87 also improve pair-averaged $\mathcal{J}\&\mathcal{F}$ against their respective targets (51.8% of all original collapses; 84.5% of the separated pairs). This criterion



Figure 4: Qualitative comparison. For each video, two referring expressions (A and B) point to different same-category objects. **Baseline**: VRS-HQ-7B exhibits *object identification ambiguity*, yielding nearly identical predictions for both expressions. **Ours**: TMR identifies the target of each expression and predicts distinct masks for A and B in the same frame.

requires better target-specific segmentation in addition to different masks. Ten of 159 previously non-collapsed pairs newly collapse.

Design choices. Table 7 compares five design choices. Removing diversity loss costs 1.9 points; replacing per-position retrieval with a global query costs 3.2. Spatial-temporal RoPE improves over temporal-only and no positional encoding by 2.3 and 4.4, respectively. Eight queries perform best among $N \in \{4, 8, 16\}$, and the all-foreground anchor pseudo mask outperforms the centered Gaussian by 2.8 points. The separate width sweep selects d_t=512 (Table 8).

5.4 Qualitative Results and Practical Scope

Figure 4 shows two expressions referring to different same-category targets. The baseline produces nearly identical masks, while TMR separates the targets. Retrieval and query-bias visualizations are in Appendix E.

Cost and coverage. The fixed eight-frame bank occupies approximately 64 MB in bf16. On seven DAVIS17 videos (431 frames, RTX 3090), with MLLM features pre-cached, SAM2-path throughput changes from 4.55 to 4.48 FPS and peak memory from 10.34 to 10.74 GB. This measures a 1.6% throughput decrease and 0.40 GB overhead, not full online MLLM latency. Training also requires an approximately six-hour feature-extraction pass and 57 hours per run on four RTX 3090 GPUs (Appendix C.5). Fixed bank storage does not grow with full video length, but longer videos can require denser sampling to cover decisive evidence; that accuracy–cost trade-off is not evaluated here.

6 Conclusion

We identify reduced frame-order decodability across the MLLM-SAM2 prompt-construction interface and an association between prompt similarity and same-category mask collapse. Motivated by these observations, TMR grounds expression-conditioned queries in sampled visual features and guides memory reading throughout propagation. Matched controls and case-level analysis show improved segmentation and fewer collapses on the diagnosed pair set, with larger gains on temporal than appearance expressions. Limitations and future work are in Appendix.F

AI USE STATEMENT

In this work, we used ChatGPT to assist with refining research hypotheses and the scope of conceptual claims, and drafting and editing manuscript text and LaTeX tables. These tools also assisted with literature search. An LLM assisted the appearance/temporal diagnostic-subset labeling described in Appendix C.8; the protocol combines rule-based labeling, LLM cross-checking, and human adjudication of borderline cases. The authors take responsibility for the accuracy, originality, and final content of this work, including all AI-assisted text, claims, and artifacts.

ETHICS STATEMENT

This work studies referring video object segmentation using existing public benchmarks and pretrained models. The method may support video editing and annotation, but could also facilitate privacy-invasive tracking or surveillance. Incorrect target identification and biases inherited from data or pretrained models can cause harm in downstream applications. Our benchmark evaluation does not establish demographic fairness or deployment reliability; remaining failures and evaluation limitations are discussed in Appendix F. Downstream use should respect data and model licenses, privacy and consent requirements, and include evaluation and human oversight appropriate to the application. We acknowledge the ICLR Code of Ethics and the authors' responsibility for responsible research and reporting.

REPRODUCIBILITY STATEMENT

Section 4 specifies the TMR architecture and training objective. Section 5 describes the benchmarks, metrics, and controls, while Appendix C documents the training data mixture, sampling, trainable modules, hyperparameters, inference protocol, and compute costs. Appendix A details the probing procedure, and Appendix C.8 describes diagnostic-subset construction. Appendix D provides the matched-control and case-level analysis protocols, including the paired-seed results in Appendix D.3. Code and trained TMR weights will be released upon acceptance.

REFERENCES

Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Zheng Shou. One Token to Seg Them All: Language Instructed Reasoning Segmentation in Videos, September 2024. URL http://arxiv.org/abs/2409.19603. arXiv:2409.19603 [cs].

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment Anything with Concepts, March 2026. URL http://arxiv.org/abs/2511.16719. arXiv:2511.16719 [cs].

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. *arXiv preprint arXiv:2312.14238*, 2023.

Claudia Cuttano, Gabriele Trivigno, Gabriele Rosi, Carlo Masone, and Giuseppe Averta. SAM-WISE: Infusing wisdom in SAM2 for text-driven video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://claudiacuttano.github.io/SAMWISE/.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. MeViS: A large-scale benchmark for video segmentation with motion expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2694–2703, 2023.

Sitong Gong, Yunzhi Zhuge, Lu Zhang, Zongxin Yang, Pingping Zhang, and Huchuan Lu. The devil is in temporal token: High quality video reasoning segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29183–29192, 2025.

Sitong Gong, Yunzhi Zhuge, Lu Zhang, Jiazuo Yu, Pingping Zhang, Xu Jia, and Huchuan Lu. Reinforcing video object segmentation to think before it segments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3835–3844, 2026.

IDEA-Research. Grounded-SAM-2: Ground and Track Anything in Videos, 2025. URL https://github.com/IDEA-Research/Grounded-SAM-2.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-UniVi: Unified Visual Representation Empowers Large Language Models with Image and Video Understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13700–13710, Seattle, WA, USA, June 2024. IEEE. ISBN 979-8-3503-5300-6. doi: 10.1109/CVPR52733.2024.01300. URL https://ieeexplore.ieee.org/document/10655738/.

Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation. *International Journal of Computer Vision*, 127(9):1175–1197, 2019.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, and others. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Xinyu Lai and others. LISA: Reasoning Segmentation via Large Language Model. *arXiv preprint arXiv:2308.00692*, 2023.

Yifan Li, Yingda Yin, Lingting Zhu, Weikai Chen, Shengju Qian, Xin Wang, and Yanwei Fu. ReVSeg: Incentivizing the reasoning chain for video segmentation with reinforcement learning. *arXiv preprint arXiv:2512.02835*, 2025.

Lang Lin, Xueyang Yu, Ziqi Pang, and Yu-Xiong Wang. GLUS: Global-Local Reasoning Unified into A Single Large Language Model for Video Segmentation. In *CVPR*, 2025.

Alexey Nekrasov, Ali Athar, Daan de Geus, Alexander Hermans, and Bastian Leibe. Sa2VA-i: Improving Sa2VA results with consistent training and inference. *arXiv preprint arXiv:2509.19082*, 2025.

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

Fu Rong, Meng Lan, Qian Zhang, and Lefei Zhang. MPG-SAM 2: Adapting SAM 2 with Mask Priors and Global Context for Referring Video Object Segmentation. *arXiv preprint arXiv:2501.13667*, 2025.

Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pp. 208–223. Springer, 2020.

Haochen Wang, Cilin Yan, Shuai Wang, Xiaolong Jiang, Xu Tang, Yao Hu, Weidi Xie, and Efstratios Gavves. Towards open-vocabulary video instance segmentation. In *proceedings of the IEEE/CVF international conference on computer vision*, pp. 4057–4066, 2023.

Haochen Wang, Qirui Chen, Cilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, Weidi Xie, and Stratis Gavves. Object-centric video question answering with visual grounding and referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22274–22284, 2025.

Junchi Wang and Lei Ke. LLM-Seg: Bridging Image Segmentation and Large Language Model Reasoning, April 2024. URL http://arxiv.org/abs/2404.08767. arXiv:2404.08767 [cs].

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution, October 2024. URL http://arxiv.org/abs/2409.12191. arXiv:2409.12191 [cs].

Zishan Xu, Yifu Guo, Yuquan Lu, Fengyu Yang, Junxin Li, and Lihua Cai. VideoSeg-R1: Reasoning video object segmentation via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11496–11504, 2026. doi: 10.1609/aaai.v40i14.38132.

Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. VISA: Reasoning Video Object Segmentation via Large Language Models. In *ECCV*, 2024.

Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2VA: Marrying SAM2 with LLaVA for Dense Grounded Understanding of Images and Videos. *arXiv preprint*, 2025.

Li Zhang, Haoxiang Gao, Zhihao Zhang, Luoxiao Huang, and Tao Zhang. SVAC: Scaling is all you need for referring video object segmentation. In *British Machine Vision Conference*, 2025. URL https://bmvc2025.bmva.org/proceedings/292/.

Rongkun Zheng, Lu Qi, Xi Chen, Yi Wang, Kun Wang, and Hengshuang Zhao. ViLLa: Video reasoning segmentation with large language model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23667–23677, 2025.

A Detailed Probing Protocols

This appendix provides full protocols for the three probes summarized in Section 3. All experiments are conducted on the publicly released VRS-HQ-7B checkpoint without any additional fine-tuning. The probing scripts are named `probe_grounding.py` (Probes A and C) and `probe_comprehensive.py` (Probe B).

A.1 Common Setup

Model and tokens. The host model is VRS-HQ-7B, in which the MLLM jointly encodes T sampled video frames and the referring expression and emits two kinds of tokens: a per-frame $\langle$SEG$\rangle$ token (one per sampled frame) and a single video-level $\langle$TAK$\rangle$ token. Both kinds are at width d_m=4096. They are mapped to 256-d through the same projection layer `text_hidden_fcs`; the 256-d $\langle$TAK$\rangle$ projection is then fused with a softmax-cosine-similarity-weighted average of the 256-d $\langle$SEG$\rangle$ projections (the Temporal Dynamic Aggregation, TDA) to produce the final 256-d sparse prompt that is consumed by SAM2. We refer to:

- the 4096-d $\langle$TAK$\rangle$ hidden state *before* projection as the ***MLLM Token***;
- the final post-TDA 256-d sparse prompt as the ***SAM2 Prompt***.

Dataset. We use the MeViS `val_u` split, which provides multiple expressions per video and same-category distractors required by Probes A and C. We sample T=8 frames uniformly per video.

Feature extraction. For every (video, expression) pair we record the MLLM Token and the SAM2 Prompt by a single forward pass of VRS-HQ-7B in evaluation mode. Both features are stored in `float32` for the linear classifiers and similarity computations below.

Predicted masks. For Probe C we additionally cache the binary predicted mask per frame produced by VRS-HQ-7B.

Reproducibility. Random seed is fixed to 42 throughout. The linear classifier in Probe B uses 5-fold `StratifiedKFold` cross-validation with `StandardScaler` feature standardization and `LogisticRegression` (`C=1.0`, `max_iter=2000`, `class_weight=balanced`).

A.2 Probe A: Cross-Interface Pair Similarity

Goal. Test whether different same-category targets within the same video become more similar in the SAM2 Prompt space than in the MLLM Token space.

Pair construction. A pair of expressions (e_i, e_j) from the same video is included if and only if all of the following hold:

1. e_i and e_j refer to *different* `anno_id`s in the MeViS metadata;
2. the inferred object category of e_i equals that of e_j, where the category is obtained by a noun-phrase parser applied to each expression and validated by manual inspection of the top categories;
3. the initial frame index used by VRS-HQ-7B is identical for both expressions, so that any prompt-space difference is not attributable to a different initial frame.

This filtering yields 327 same-category different-annotation pairs across the MeViS `val_u` split.

Similarity metric. For each pair we compute cosine similarity in both feature spaces:

$$\cos_{\text{raw}}(i,j) = \frac{\langle h_i^{\text{TAK}}, h_j^{\text{TAK}} \rangle}{\|h_i^{\text{TAK}}\| \, \|h_j^{\text{TAK}}\|}, \quad \cos_{\text{prompt}}(i,j) = \frac{\langle p_i, p_j \rangle}{\|p_i\| \, \|p_j\|}, \tag{11}$$

where h^{TAK} is the MLLM Token and p is the 256-d SAM2 Prompt.

Statistics. We report (i) mean and median of $\cos_{\text{raw}}$ and $\cos_{\text{prompt}}$ over the 327 pairs, and (ii) the fraction of pairs with $\cos_{\text{prompt}} > \cos_{\text{raw}}$. Section 3 reports the mean similarities $0.447 \rightarrow 0.804$ (mean delta $+0.357$, median delta $+0.407$). Higher prompt-space similarity occurs in 91.1% $(298/327)$ of the pairs, with an average increase of $+0.401$ among these increasing pairs.

A.3 Probe B: Temporal-Order Probing

Goal. Test whether frame-order labels are more accurately decoded from the MLLM Token than from the SAM2 Prompt.

Frame orderings. For each video we generate three classification tasks of increasing difficulty by feeding $T=8$ frames in different orders while keeping the expression identical:

- **2-way** (chance 50%): {FORWARD, REVERSE}.
- **4-way** (chance 25%): {FORWARD, REVERSE, INTERLEAVE-EVEN-ODD, SHIFT-HALF}.
- **8-way** (chance 12.5%): the four orderings above plus {SHIFT-QUARTER, REVERSE-INTERLEAVE, EVEN-FIRST-REVERSED, SEEDED-RANDOM} (seed $=42$).

Classifier. For each $K \in \{2, 4, 8\}$ and each feature stream (MLLM Token, SAM2 Prompt), we train a logistic-regression classifier with the protocol in Appendix A.1. We report 5-fold cross-validated mean accuracy.

Sample sizes. 738 valid expressions $\times$ K frame orderings per expression yields $K \cdot 738$ feature samples per task. The classifier predicts the ordering label.

Reported numbers. Section 3 cites $86.8\%/72.6\%/62.8\%$ on the MLLM Token and $70.3\%/39.1\%/22.7\%$ on the SAM2 Prompt for $K=2/4/8$. The chance-corrected retention $(\text{acc}_{\text{prompt}} - 1/K)/(\text{acc}_{\text{raw}} - 1/K)$ is 55.2%, 29.6%, and 20.3% respectively, showing reduced chance-corrected probe accuracy: approximately 55% is retained for the 2-way task, and 30%/20% for the 4-way/8-way tasks. These ratios quantify probe accuracy rather than an information-theoretic fraction of retained information.

A.4 Probe C: Prompt-to-Mask Collapse Probe

Goal. Test whether high SAM2 Prompt similarity is associated with nearly identical predicted masks.

Pair set. We use the same 327 pairs as Probe A, with the additional requirement that VRS-HQ-7B produces a non-empty predicted mask for both expressions on every sampled frame.

Mask similarity metric. For each pair we compute the per-video mask Intersection-over-Union, averaged over the $T=8$ sampled frames:

$$\text{maskIoU}(i, j) = \frac{1}{T} \sum_{t=1}^{T} \frac{|\hat{y}_i^{(t)} \cap \hat{y}_j^{(t)}|}{|\hat{y}_i^{(t)} \cup \hat{y}_j^{(t)}|}, \tag{12}$$

where $\hat{y}_i^{(t)}$ is the binarized predicted mask of expression i on frame t. A pair with $\text{maskIoU} \geq 0.9$ is treated as a *same-mask collapse*; the same-mask rate at threshold τ is the fraction of pairs with $\text{maskIoU} \geq \tau$.

Prompt-similarity binning. We bin pairs into seven ranges of $\cos_{\text{prompt}}$: $[0.0, 0.4)$, $[0.4, 0.5)$, $[0.5, 0.6)$, $[0.6, 0.7)$, $[0.7, 0.8)$, $[0.8, 0.9)$, $[0.9, 1.0]$. The 0.4 boundary corresponds to typical within-category similarity in the MLLM Token space ($\bar{\cos}_{\text{raw}}=0.447$); the 0.9 boundary isolates the high-collapse tail. Bin sizes are 19, 17, 35, 52, 85, 86, 33 pairs respectively (sum $= 327$).

Reported statistics. Across the seven bins, the same-mask rate at τ=0.9 generally rises from 10.5% to 72.7%, and at τ=0.95 from 5.3% to 63.6%. The full breakdown is shown in Table 9. Globally, 168/327 pairs satisfy maskIoU$\geq$0.9 and 122/327 satisfy maskIoU$\geq$0.95. Pearson correlation between $\cos_{\text{prompt}}$ and maskIoU is r=0.41 (Spearman ρ=0.46), both with $p<10^{-15}$.

Table 9: Per-bin statistics for Probe C on MeViS `val_u`, 327 same-category different-annotation pairs.

$\cos_{\text{prompt}}$ bin	n	Mean maskIoU	Median maskIoU	Rate ≥ 0.9	Rate ≥ 0.95
$[0.0, 0.4)$	19	0.122	0.021	10.5%	5.3%
$[0.4, 0.5)$	17	0.570	0.865	23.5%	17.6%
$[0.5, 0.6)$	35	0.580	0.706	20.0%	8.6%
$[0.6, 0.7)$	52	0.579	0.787	38.5%	23.1%
$[0.7, 0.8)$	85	0.687	0.921	57.6%	40.0%
$[0.8, 0.9)$	86	0.817	0.961	72.1%	55.8%
$[0.9, 1.0]$	33	0.852	0.985	72.7%	63.6%

B Cross-Model Probing: Sa2VA-InternVL3-14B

These supplementary probes compare the 5120-d MLLM hidden representation and the 256-d SAM2 prompt in the Sa2VA-InternVL3-14B host. They describe representation changes across the interface; the primary diagnosis and case-level validation use VRS-HQ-7B.

B.1 Probe Results

Probe A – pair similarity across the interface. On 327 within-video same-category different-annotation expression pairs, the mean cosine similarity rises from 0.430 in MLLM Token space to 0.710 in SAM2 Prompt space, an increase of 0.280.

Probe B – temporal-order classification. Per-K breakdown of Sa2VA Probe B:

K	h_seg accuracy	prompt accuracy	Δ
2	0.92	0.55	0.37
4	0.85	0.32	0.53
8	0.78	0.18	0.60

The displayed accuracies give chance-corrected retention ratios $(\text{acc}_{\text{prompt}} - 1/K)/(\text{acc}_{\text{raw}} - 1/K)$ of 11.9%, 11.7%, and 8.4% for K=2/4/8, respectively. Prompt accuracies exceed chance by 5.0, 7.0, and 5.5 percentage points. These are ratios of probe accuracies, not direct measurements of information content.

Probe C – prompt-to-mask collapse correlation. Seven-bin breakdown:

$\cos_{\text{prompt}}$ bin	n	Mean mask IoU
$[0.0, 0.4)$	28	0.05
$[0.4, 0.5)$	29	0.18
$[0.5, 0.6)$	27	0.32
$[0.6, 0.7)$	35	0.55
$[0.7, 0.8)$	82	0.74
$[0.8, 0.9)$	91	0.86
$[0.9, 1.0]$	35	**0.92**

C Implementation Details

This appendix provides the training recipe and hyperparameter values for the experiments in Section 5. Design-choice ablations, prompt-generator capacity, and TMR width sweeps are reported in the main text in Tables 7 and 8. Reference scripts (`train_TMR.py` and `run_train_TMR_vrshq7b.sh`) will be released with the code.

C.1 Training Data and Sampling

Dataset mixture. We train on a five-source mixture of public referring-video and large-vocabulary video-segmentation datasets, with the per-epoch sampling ratio biased toward MeViS and ReVOS to match the temporal-reasoning emphasis of TMR:

Dataset	Per-epoch sample count	Share
MeViS train (Ding et al., 2023)	8,000	31.3%
Ref-YouTube-VOS train (Seo et al., 2020)	8,000	31.3%
ReVOS train (Yan et al., 2024)	6,000	23.4%
LV-VIS train (Wang et al., 2023)	3,200	12.5%
Ref-DAVIS17 train (Khoreva et al., 2019)	400	1.6%
Total	25,600	100%

The mixture is resampled at the start of each epoch, so the model sees $25{,}600 \times \text{epochs}$ training clips in total. Compared with a uniform mix that would treat every dataset equally, this schedule allocates more capacity to the two benchmarks (MeViS, ReVOS) where temporal disambiguation and reasoning are the dominant difficulty.

Frame sampling. For each training clip we draw a window of $\text{max_frames}{=}16$ contiguous frames from the source video and uniformly subsample $\text{num_train_frames}{=}9$ frames within the window. The conditioning anchor frame t_{init} inherits the host MLLM's key-frame selection (Section 4.4); during training we treat any of the 9 sampled frames as a valid anchor and randomize across mini-batches.

Prompt dropout. To prevent the model from over-relying on the baseline 256-d prompt and to encourage the temporal-token route to carry referring information, we linearly anneal a prompt-dropout rate from 0 at epoch 0 to 0.15 at the final epoch.

C.2 Optimization

Optimizer. AdamW with $\text{lr}{=}1{\times}10^{-4}$, weight decay 0.01, $\beta_1{=}0.9$, $\beta_2{=}0.999$.

Schedule. 10 epochs in bf16 mixed precision. Learning rate follows a cosine schedule with 500 linear warm-up steps. Gradient clipping at norm 1.0 and gradient accumulation of 4 micro-batches give an effective batch size of $4 \times 4 = 16$ clips on the 4-GPU setup below.

Trainable scope. The host MLLM and the SAM2 image encoder are kept frozen; the SAM2 memory encoder, memory attention, mask decoder, and all TMR modules (temporal queries E_t and the linear maps ϕ_q, ϕ_m, the grounding block, the per-position retrieval cross-attention, and the per-channel gate $\boldsymbol{\alpha}$) are jointly fine-tuned. The baseline prompt projector `text_hidden_fcs` is frozen.

C.3 TMR Hyperparameters

Table 10 lists every hyperparameter introduced in Section 4. Defaults are used in the main result tables; ablation values are given in Section 5 and noted in the table where relevant.

Gate initialization. The per-channel gate $\mathbf{g} = \sigma(\boldsymbol{\alpha})$ is initialized so that $\mathbf{g} \approx 0.1 \cdot \mathbf{1}$, i.e. each channel initially scales the grounded readout Z_t by 0.1. Concretely we initialize $\boldsymbol{\alpha} = \sigma^{-1}(0.1) \cdot \mathbf{1}$. Training can subsequently adjust each channel's scaling coefficient.

Table 10: TMR architecture and training hyperparameters. "Argname" refers to the corresponding flag in `train_TMR.py`.

Symbol	Value	Argname
Dimensions		
d_m (MLLM hidden)	4096	`d_mllm`
d_t (grounding intermediate)	512	`grounding_d_attn`
d_{mem} (SAM2 memory)	256	`d_mem_q`
Temporal queries (Sec. 4.3)		
N (number of temporal queries)	8	`n_tokens`
E_t initialization	$\mathcal{N}(0, 0.02)$	—
Grounding block (SA + CA + FFN)		
Layers (L)	2	`grounding_layers`
Heads	8	`grounding_heads`
FFN expansion	$2\times$	`grounding_ffn_mult`
3D-RoPE base frequency	10,000	—
LayerNorm placement	pre-LN	—
Mask-Free Visual Bank (Sec. 4.2)		
M (sampled bank frames)	8	`max_bank_tokens`
Per-position retrieval (Sec. 4.4)		
Heads	4	`retrieval_heads`
Query-bias injection		
Gate parameterization	$\mathbf{g} = \sigma(\boldsymbol{\alpha})$	—
$\boldsymbol{\alpha}$ initialization	$\sigma^{-1}(0.1)$ (i.e. $\mathbf{g} \approx 0.1$)	`init_lambda`
Anchor frame memory (Sec. 4.4)		
Pseudo mask P	all-foreground ($P \equiv 1$)	`pseudo_mask_mode=ones`
Loss		
λ_{div} (token diversity)	0.1	`lambda_div`
BCE weight	2.0	—
Dice weight	0.5	—
Optimization		
Optimizer	AdamW	—
Learning rate	1×10^{-4}	`lr`
Weight decay	0.01	`weight_decay`
Warm-up steps	500	`warmup_steps`
LR schedule	cosine	—
Epochs	10	`epochs`
Gradient accumulation	4	`grad_accum`
Gradient clip	1.0	`grad_clip`
Precision	bf16	`precision`
Prompt dropout (linear schedule)	$0 \rightarrow 0.15$ over epochs	`max_prompt_dropout`

C.4 Loss Function

Per training clip the segmentation loss aggregates BCE-with-logits and a smoothed Dice loss over all K valid (non-empty-GT) frames:

$$\mathcal{L}_{\text{seg}} = \frac{1}{K}\sum_{k=1}^{K}\big(2.0\,\mathcal{L}_{\text{BCE}}(\hat{y}_k, y_k) + 0.5\,\mathcal{L}_{\text{Dice}}(\hat{y}_k, y_k)\big), \tag{13}$$

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2|\hat{y} \cap y| + 1}{|\hat{y}| + |y| + 1}. \tag{14}$$

The total loss is $\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_{\text{div}}\,\mathcal{L}_{\text{div}}$ with λ_{div}=0.1 and $\mathcal{L}_{\text{div}}$ defined in Eq. 10. The diversity loss discourages similar directions among the N queries; removing it reduces $\mathcal{J}\&\mathcal{F}$ by 1.9 points in Table 7(i).

C.5 Hardware and Runtime

MLLM-feature pre-extraction. The host MLLM is the most expensive component of the pipeline, and its weights are kept frozen throughout (Section 4.5). We exploit this by running the host MLLM *once offline* on every (video, expression) pair in the training mixture, caching the segmentation-token hidden state h_{seg} to disk (one tensor per pair, 4096-d in fp16, $\sim$8 KB each). At training time TMR consumes the cached h_{seg} directly, so each forward/backward step only traverses the SAM2 modules and the 15.49M-parameter TMR head, not the 7B host MLLM. This cache covers $\sim$56,000 (video, expression) pairs from the five-source training mixture and occupies $\sim$0.5 GB on disk; pre-extraction itself takes $\sim$6 hours on the same 4-GPU setup and is amortized across all subsequent runs (e.g. ablations, hyperparameter sweeps).

Wall-clock. Training is performed on 4$\times$ NVIDIA RTX 3090 (24 GB) GPUs with PyTorch DDP (`torchrun --nproc_per_node=4`). bf16 mixed precision is used throughout. With the cached-feature setting above, one epoch takes approximately 5.75 hours wall-clock on the VRS-HQ-7B host, so a full 10-epoch run completes in roughly 57 hours ($\sim$2.4 days). Peak GPU memory is below 22 GB per device.

Without pre-extraction. If the host MLLM is forwarded online during training (i.e. no cache), per-step cost is dominated by the 7B-parameter MLLM forward pass. In that mode a single epoch under the same recipe would not fit on 4$\times$ RTX 3090 without aggressive offloading, and even with offloading the per-epoch wall-clock multiplies by roughly an order of magnitude. We therefore adopt cached features as the default and recommend it for any host MLLM whose weights are frozen.

Inference speed and memory. Table 11 compares the inference cost of TMR against the vanilla SAM2 baseline (i.e., the same SAM2 image encoder, memory attention, and mask decoder, but without per-position retrieval, gated query-bias injection, or MFVB-derived anchor memory). Both modes are run on the identical set of 7 DAVIS17 videos (431 propagation frames in total) using a single NVIDIA RTX 3090 with bf16 mixed precision, and with MLLM features pre-cached. TMR reduces SAM2-path throughput by 1.6% and 0.40 GB peak GPU memory ($< 4\%$) on top of the vanilla SAM2 pipeline; the SAM2 image encoder remains the dominant cost ($\sim$74% of wall-clock).

Table 11: Inference speed and peak GPU memory of TMR vs. the vanilla SAM2 baseline on 7 DAVIS17 validation videos (431 frames; RTX 3090, bf16; MLLM features pre-cached). Δ shows the overhead of TMR over baseline.

Metric	Baseline	TMR	Δ
FPS	4.55	4.48	-1.6%
Peak GPU memory (GB)	10.34	10.74	$+0.40$

These measurements quantify overhead in the cached-feature SAM2 path on this seven-video workload. They exclude the online host MLLM pass and do not establish scaling to longer videos or denser visual banks.

C.6 INFERENCE

Pipeline. At test time we follow each baseline's standard inference protocol; only the SAM2 path is replaced by TMR-augmented memory reading. For a given (video, expression) input we (1) run the host MLLM once to obtain h_{seg}, (2) run the SAM2 image encoder on the $M{=}8$ key frames selected by the host pipeline to build the MFVB and on every video frame to obtain $\{F_t\}_{t=1}^{T}$, (3) compute Z_g and Z_{mem} once via the grounding block and ϕ_m, and (4) propagate frame-by-frame: for each t we perform per-position cross-attention into Z_{mem} to obtain Z_t, inject the gated bias $\sigma(\boldsymbol{\alpha}) \odot Z_t$ into F_t, run SAM2 memory attention with the appropriate $\mathcal{M}_t$, decode the mask, and write the predicted mask back to the memory bank.

Anchor frame and propagation order. The anchor frame t_{init} is selected by the upstream key-frame strategy of the host MLLM and is not necessarily the first temporal frame; we adopt this selection unchanged. Propagation runs forward and backward from t_{init}, following the standard SAM2 video predictor; both directions share the same Z_{mem} and the same gated injection.

Memory bank composition. At $t{=}t_{\text{init}}$ the memory bank $\mathcal{M}_{t_{\text{init}}}$ is constructed from MFVB-derived entries via the SAM2 memory encoder applied to each F_m (with $m{:}\,\tau_m{\neq}t_{\text{init}}$) and the expression-agnostic pseudo mask $P{\equiv}1$. At every other propagation frame t, $\mathcal{M}_t$ is the standard SAM2 predicted-mask memory bank; recently written entries dominate by SAM2's default windowing.

Resolution and precision. Frames are resized so that the longer side equals 1024 (SAM2 default) and padded to a square. Inference is run in bfloat16 on a single NVIDIA RTX 3090; the SAM2 image encoder is the dominant cost, while the TMR modules add a near-negligible overhead, as quantified in Table 11.

No test-time tricks. We use no test-time augmentation (no flip / multi-scale / multi-crop), no model ensembling, no temperature or threshold tuning, and no post-processing (no CRF, no temporal smoothing). The SAM2 mask decoder's logit threshold is left at 0.

Outputs. For each frame the binary predicted mask $\hat{y}_t$ is upsampled to the original frame resolution by SAM2's standard upsampling head, written to disk as a PNG, and evaluated against the official benchmark scoring scripts ($\mathcal{J}$, $\mathcal{F}$, $\mathcal{J}\&\mathcal{F}$).

C.7 COMPUTATIONAL COST ANALYSIS

All numbers are measured on a single NVIDIA RTX 3090 in bfloat16 (multi-GPU training uses $4\times$ the same device); MLLM features are pre-cached.

Additional parameters. On VRS-HQ-7B, the default TMR configuration ($d_m{=}4096$, $d_t{=}512$, $N{=}8$, $L{=}2$) adds 15.49M trainable parameters, using the same counting scope as the width sweep and prompt-control comparison (Tables 8 and 4). This counts the added TMR modules, including the entry projection, temporal embeddings, grounding blocks, output projection, retrieval attention, and gate. The host MLLM, baseline prompt projector, and SAM2 image encoder are frozen and excluded. The jointly fine-tuned SAM2 memory encoder, memory attention, and mask decoder are shared with the matched baseline and are also excluded from this added-parameter count. At $d_t{=}256/1024$, the corresponding added counts are 6.24M/43.83M. Only the entry projection ϕ_q scales directly with the host hidden width, so 15.49M describes this host configuration rather than an identical count for every host.

Inference latency. Per-stage timings on 7 DAVIS17 videos (431 propagation frames) are in Table 12. The SAM2 image encoder dominates (160.9 ms/frame, $\sim$75% of wall-clock); TMR's grounding block runs once per (video, expression) at 3.9 ms; the per-frame TMR cost (retrieval 0.35 ms + injection 0.06 ms) is 0.41 ms, i.e. 0.2% of the per-frame budget.

GPU memory and FLOPs. TMR adds 0.40 GB to peak GPU memory at inference (Table 11); the cached MFVB tensor occupies approximately 64 MB in bf16, with other activations and working memory contributing to peak usage. The two TMR cross-attentions are $\mathcal{O}(N{\cdot}M{\cdot}HW{\cdot}d_h)$ once and

Table 12: Per-stage inference latency, measured with `torch.cuda.Event` timers (RTX 3090, bf16; 7 DAVIS17 videos, 431 propagation frames).

Stage	Frequency	Mean time
SAM2 image encoder	per frame	160.9 ms
TMR grounding block (L=2)	once / video-exp	3.9 ms
TMR per-position retrieval	per frame	0.35 ms
TMR gated injection	per frame	0.06 ms
SAM2 memory attention	per frame	46.7 ms
SAM2 mask decoder	per frame	5.7 ms
SAM2 memory write-back	per frame	2.2 ms
Per-frame total		216.0 ms ($\approx$ 4.6 FPS)

$\mathcal{O}(HW \cdot N \cdot d_h)$ per frame. These asymptotic terms describe attention interactions and do not include every projection or activation.

Training cost. On a 7B host with cached MLLM features, training takes $\sim$5.75 hours per epoch on 4$\times$ RTX 3090 ($\sim$57 hours for the full 10-epoch run); MLLM-feature extraction is a one-time $\sim$6-hour offline pass amortised across all subsequent runs.

C.8 Diagnostic-Subset Partitioning

Each of the 793 MeViS `val_u` expressions is labeled as *appearance-identifiable* (target distinguishable from a single frame, e.g. by colour, size, or static spatial position) or *temporal-identifiable* (target distinguishable only by observing multiple frames, e.g. by motion direction, action sequence, or state change). The labeling combines a rule-based pass that scores temporal versus appearance cues from regular-expression patterns, an LLM cross-validation pass that re-labels each expression with a brief rationale, and a small human-review pass that adjudicates the $\sim$150 borderline cases (low rule confidence or rule–LLM disagreement). The final partition is 421 appearance vs. 372 temporal expressions, and these labels define the Appearance and Temporal columns of Table 5.

D Additional Results and Rebuttal Controls

D.1 Extended Comparisons and Matched Fine-Tuning

Tables 1 and 2 in the main text report the region/contour breakdowns and ReVOS referring/reasoning scores. Table 13 adds recent-method comparisons, MeViS val^u results, and the matched fine-tuning baseline. We report their available $\mathcal{J}\&\mathcal{F}$ scores without inferring unavailable $\mathcal{J}/\mathcal{F}$ components from averages.

D.2 Matched Training and Prompt Controls

The matched VRS-HQ-7B baseline removes MFVB, grounding, and memory-reading components while retaining the same five-source mixture, per-source sample counts, optimizer, ten-epoch schedule, and trainable SAM2 components. Its baseline prompt projector remains frozen. We disable prompt dropout because the one-shot prompt is its only language-conditioning path. This isolates additional SAM2 fine-tuning under the chosen mixture, not robustness to alternative mixtures or unseen training sources.

The multi-token control maps h_{seg} into eight trainable 256-d tokens and jointly feeds them to SAM2 as anchor-frame sparse prompts. It uses the same training mixture, with no MFVB, grounding, or memory reading, and also disables prompt dropout. Its additional parameter count is approximately 25.2M, compared with 15.49M for the TMR configuration in the width sweep. It reaches 57.1 overall and 43.8 on Temporal expressions, versus 62.1 and 52.2 for TMR.

The repeated-prompt control applies the same 256-d baseline prompt at every propagation frame, keeping the other baseline components and evaluation settings unchanged. It reaches 55.6, versus

Table 13: RVOS comparison ($\mathcal{J}\&\mathcal{F}$). MeViS val and val^u are different evaluation splits. †: reproduced host checkpoints; FT: matched fine-tuning without TMR. Other methods use reported results and differ in training data, frame budgets, and tuning. Dashes indicate unavailable results. Component scores for the original comparison set are in Tables 1 and 2.

Method	Backbone	MeViS val^u	MeViS val	Ref-YT-VOS	Ref-DAVIS17	ReVOS
SAMWISE (Cuttano et al., 2025)	CLIP-B	–	48.3	67.2	68.5	–
MPG-SAM2 (Rong et al., 2025)	BEiT-3-L	–	53.7	73.9	72.4	–
Grounded-SAM2 (IDEA-Research, 2025)	BERT	–	37.7	57.5	66.4	–
LISA-7B (Lai & others, 2023)	LLaVA-7B	–	37.2	53.9	64.8	40.9
VISA-7B (Yan et al., 2024)	Chat-UniVi-7B	–	43.5	61.5	69.4	46.9
VideoLISA (Bai et al., 2024)	LLaVA-Phi-3-V	–	44.4	63.7	68.8	–
ViLLa (Zheng et al., 2025)	InternVideo2-6B	–	49.4	67.5	74.3	–
GLUS (Lin et al., 2025)	LISA-7B	–	51.3	67.3	–	–
SVAC-8B (Zhang et al., 2025)	InternVL2.5-8B	62.8	–	–	79.4	59.2
Veason-R1-7B (Gong et al., 2026)	Qwen2.5-VL-7B	–	52.2	–	–	61.3
VideoSeg-R1-7B (Xu et al., 2026)	Qwen2.5-VL-7B	–	55.3	81.3	79.8	61.1
ReVSeg-7B (Li et al., 2025)	Qwen2.5-VL-7B	–	59.8	73.1	80.8	62.1
Sa2VA-2B†	Qwen3-VL-2B	–	44.7	62.9	72.2	52.7
+ TMR	Qwen3-VL-2B	–	48.6	72.0	75.7	56.6
VRS-HQ-7B†	Chat-UniVi-7B	56.6	50.7	70.4	76.0	59.1
VRS-HQ-7B + FT	Chat-UniVi-7B	56.8	50.7	70.2	75.9	58.8
VRS-HQ-7B + TMR	Chat-UniVi-7B	62.1	56.4	73.1	76.9	64.6
Sa2VA-8B†	InternVL2.5-8B	–	55.1	72.4	76.6	62.2
+ TMR	InternVL2.5-8B	64.0	58.7	74.6	76.8	64.8
Sa2VA-14B†	InternVL3-14B	–	54.9	70.5	76.1	62.5
+ TMR	InternVL3-14B	–	61.2	75.1	77.3	66.9

56.6 for key-frame prompting. This tests repeated delivery of the existing prompt; it does not evaluate all possible retrained per-frame prompting architectures.

D.3 Paired-Seed Results

For each pre-specified seed in $\{42, 43, 44\}$, we train a matched VRS-HQ-7B baseline and TMR with paired data ordering and evaluate both checkpoints on MeViS val^u and Ref-DAVIS17. Seed 42 is the original reported run. Table 14 gives the individual scores; Table 15 gives sample means and standard deviations. The confidence interval is

$$\bar{\Delta} \pm t_{0.975,2}\, s_\Delta/\sqrt{3}, \qquad \Delta_s = (\mathcal{J}\&\mathcal{F})_{\mathrm{TMR},s} - (\mathcal{J}\&\mathcal{F})_{\mathrm{FT},s}. \tag{15}$$

Statistics and intervals use the full-precision scores before display rounding. The intervals capture training-seed variation conditional on the fixed data mixture and evaluation sets; they do not estimate uncertainty over new videos or datasets.

Table 14: Three paired training seeds. Scores are $\mathcal{J}\&\mathcal{F}$.

Benchmark	Seed	Matched FT	TMR	Paired gain
MeViS val^u	42	56.8	62.1	+5.3
	43	57.1	62.3	+5.2
	44	56.6	62.1	+5.5
Ref-DAVIS17	42	75.9	76.9	+1.0
	43	76.0	77.1	+1.1
	44	75.9	76.8	+0.9

D.4 Capacity-Control Notes

The prompt-generator capacity and TMR width sweeps are reported in the main text in Table 8, alongside the design ablations in Table 7 (Section 5.3). Added-parameter counts use the scope defined in Appendix C; they exclude the shared, jointly fine-tuned SAM2 modules.

Table 15: Mean $\pm$ sample standard deviation across seeds and 95% Student-t confidence intervals over paired gains.

Benchmark	Matched FT	TMR	Paired gain	95% CI
MeViS val^u	56.83 ± 0.25	62.17 ± 0.12	5.33 ± 0.15	$[4.95, 5.71]$
Ref-DAVIS17	75.93 ± 0.06	76.93 ± 0.15	1.00 ± 0.10	$[0.75, 1.25]$

D.5 Case-Level Collapse Protocol

We use the fixed 327 Probe-C pairs and predictions already generated for baseline and TMR, with no new training or inference. Let c_b and c_o be the mean mask IoUs between the two expressions under baseline and TMR. An original collapse has $c_b \geq 0.9$, a separated original collapse additionally has $c_o < 0.9$, and a new collapse has $c_b < 0.9$ and $c_o \geq 0.9$. For each expression we evaluate against its own target annotation, then average the two $\mathcal{J}\&\mathcal{F}$ scores within the pair. We report separation with improvement only if that pair average increases. Of 168 original collapses, 103 separate, including 87 with improved target-specific quality. The remaining 65 original collapses plus 10 new collapses account for the 75 TMR collapses. These are descriptive pair-level results; expression pairs can share a video, so they are not asserted to be independent benchmark samples.

D.6 Expression Composition and Uneven Gains

A fixed rule-based classifier identifies explicit temporal words or constructions in each full evaluation set (Table 16). This is distinct from the finer MeViS val^u diagnostic partition in Appendix C.8. The temporal-cue prevalence and 14B gains follow the same ordering, but three dataset-level observations do not prove that expression composition causes the differences. Host baseline quality also matters: Ref-YT-VOS baseline/gain pairs are 62.9/+9.1 (2B), 70.4/+2.7 (7B), 72.4/+2.2 (8B), and 70.5/+4.6 (14B). Because the backbones differ, this is not a controlled scaling experiment.

Table 16: Explicit temporal cues in full evaluation sets; counts are descriptive rule-based labels. Baseline and gain use the 14B host.

Dataset	Expressions	Temporal cues	Ratio	Baseline	Gain
MeViS val	2,236	1,377	61.6%	54.9	+6.3
Ref-YT-VOS	2,096	888	42.4%	70.5	+4.6
Ref-DAVIS17	244	52	21.3%	76.1	+1.2

E Per-Frame Visualizations of TMR

This appendix illustrates spatial retrieval and query-bias injection in two *stationary-state* cases. The expressions distinguish a non-moving target from moving same-category distractors with similar appearance. In the displayed frames, the baseline follows a distractor while TMR predicts the referred target. These examples complement the quantitative case-level analysis in Section 5.3.

Each figure has five rows for the same five sampled frames:

- **Input**: the raw frames.
- **Baseline**: VRS-HQ-7B prediction overlaid in red.
- **TMR**: our prediction overlaid in blue.
- **Retrieval**: per-position $\arg\max$ over the $N{=}8$ grounded temporal tokens (Eq. 7). Each color denotes one token.
- **Query Bias**: the injection magnitude $\|\mathbf{g} \odot Z_t\|$ from Eq. 8 in the inferno colormap; bright yellow denotes larger vector magnitude, without measuring its semantic content or correctness.

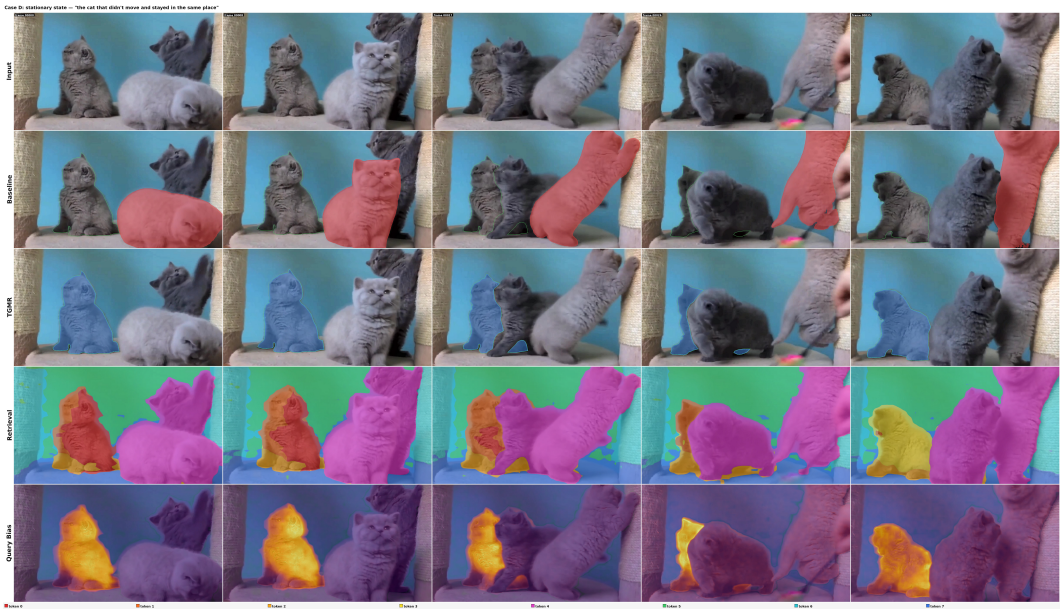


Figure 5: Per-frame method visualization on **Case 1**, expression: *the cat that didn't move and stayed in the same place*. Top to bottom: input, VRS-HQ-7B baseline prediction (red), TMR prediction (blue), per-position retrieval over $N{=}8$ grounded temporal tokens, and query-bias magnitude $\|\mathbf{g} \odot Z_t\|$. In the displayed frames, the target region selects warm-colored token indices and shows larger injection magnitude than the moving distractors. Colors denote token indices, without predefined target/distractor semantics.

Case 1: stationary kitten among moving kittens. Figure 5 shows similarly-colored kittens on a cat tower. The expression is *the cat that didn't move and stayed in the same place*, referring to the kitten on the left. The baseline predicts a central/right distractor at the anchor frame and continues to follow a moving cat in the displayed later frames. In the retrieval map, the target region selects red/orange token indices while moving distractors select magenta. The query-bias map has a larger magnitude near the stationary target, and TMR's predictions remain aligned with that target in the shown frames, including those with overlap and clutter. These visualizations show spatially differentiated readout alongside the improved prediction.

Case 2: parked bike vs. walking bike. Figure 6 shows two similar-looking bikes: a parked bike on the left (the target) and a bike being walked by a pedestrian. The expression is *the bike that hasn't moved*. The baseline overlaps the target only partially at frame 0 and follows the moving bike in the displayed frames from frame 11 onward. The retrieval map assigns different token indices to the parked and moving bikes, including when they are adjacent at frame 22. Query-bias magnitude is larger near the parked bike, which TMR continues to segment in the five displayed frames. The grounded token bank is computed once; its position-dependent readout changes at each propagation frame. This illustrates the intended spatial conditioning, without attributing the result to a unique internal cause.



Figure 6: Per-frame method visualization on **Case 2**, expression: *the bike that hasn't moved*. The parked bike and the bike being walked share nearly identical appearance. TMR routes them to different temporal tokens (warm vs. magenta) and concentrates the query-bias injection on the parked bike across all sampled frames. The baseline collapses onto the moving bike from frame 11 onward.

F Limitations and Future Work

Scope of the diagnosis. The probes compare frame-order decodability across prompt construction and measure the association between prompt similarity and mask collapse. They do not isolate projection from TDA, and predicting input permutations does not establish correct target-specific reasoning. Cross-space cosine similarity is not an information-theoretic measure. TMR accesses both the pre-projection host state and a visual bank; its gains do not isolate recovery of attenuated host information from new visual grounding or altered memory conditioning. The 327 same-category pairs form a selected diagnostic set, not an estimate of the full benchmark error distribution. Residual and newly introduced collapses remain after TMR.

Host and visual-evidence dependence. TMR keeps the host MLLM frozen. It cannot reliably compensate for evidence absent from the host state or missing from the sampled visual bank. Occlusion, small objects, incorrect upstream interpretation, and appearance-association errors can require additional mechanisms. The fixed expression-agnostic anchor pseudo mask also omits target-location priors, and alternative anchor selection has not been evaluated.

Uneven gains and training distribution. The benefit depends on the benchmark and original host quality. Matched fine-tuning isolates additional training on VRS-HQ-7B under the chosen five-source mixture; it does not establish robustness to alternative mixtures, excluded training sources, or all host configurations. The rule-based expression-composition comparison is descriptive. Three-seed confidence intervals quantify training variability on two fixed evaluation settings, not uncertainty over new datasets.

Bank coverage and computational cost. MFVB stores eight sampled feature maps at native resolution, adding memory relative to prompt-only conditioning. At fixed M, H, W, bank storage is independent of full video length, but a fixed bank can miss decisive events in longer videos. Increasing the frame budget or resolution has additional storage and grounding cost; we have not measured that accuracy–cost trade-off. Reported speed excludes the host MLLM forward pass, and training requires offline feature extraction plus SAM2/TMR optimization.

Future work and transfer. Joint host optimization, adaptive sampling, bank compression, and sparse or refreshed memory may address these limits. Current evidence supports augmentation across four host configurations within RVOS, not transfer across task families. Language-guided instance segmentation or open-vocabulary video segmentation would require suitable instance-conditioned queries and access to memory reading. Standard VOS or tracking would additionally require a non-language query source, such as first-frame mask-pooled features or motion queries; these extensions remain unevaluated.