VIRST: Video-Instructed Reasoning Assistant for SpatioTemporal Segmentation

Supplementary Material

A. Implementation Details

A.1. Experimental Setup

Hardware specifications. All experiments were run on 8× NVIDIA H100 80GB GPUs, an Intel Xeon Platinum 8480+ CPU, and 2 TB RAM.

Dataset compositions. The overall dataset composition follows VISA [57] and LISA [24]. The datasets used for training are listed in Table 7. We reuse the official VISA dataloader implementations with a few minor adjustments.

Table 7. Datasets used for training VIRST.

Category	Dataset
RVOS	Ref-DAVIS17 [22] Ref-YouTube-VOS [45] MeViS [9] ReVOS [57]
Video Instance Segmentation	LV-VIS [46]
Referring Image Segmentation	RefCOCO [21] RefCOCO+ [21] RefCOCOg [21]
Semantic Segmentation	ADE20k [62] COCO-Stuff [4] PACO [39] PASCAL-Part [5]
Image Reasoning	ReasonSeg [24]
Video Instruction Tuning	VideoLLaVA-Instruct [32]

For PACO and PASCAL-Part, we modify the loader to produce complete part-level masks when a textual reference corresponds to multiple object parts, ensuring consistent supervision across part-segmentation datasets. In addition, a small number of ReVOS training samples contained misaligned annotations where the ground-truth mask did not match the associated frame; these corrupted instances were excluded. All validation and test sets were used exactly as provided, and no dataset-specific modifications were applied during inference.

Additionally, a subset of the VideoLLaVA-Instruct dataset was incorporated to preserve the model's video understanding capabilities. This additional supervision preserves robust semantic video understanding in VIRST, enabling it to generate natural-language responses in an autoregressive manner. Relevant results and analysis are provided in Appendix D.3.

A.2. Training Procedure

Initialization. We adopt VideoChat-Flash [29] as our vision–language backbone. Specifically, we use the publicly released `VideoChat-Flash-Qwen2-7B_res448` checkpoint from HuggingFace, chosen for its strong video understanding capability and reliable reproducibility. For the segmentation module, we employ SAM 2.1 initialized from the `sam2.1_hiera_large` checkpoint. All other components not explicitly mentioned are initialized from scratch.

Training objective. The overall training objective is expressed as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} + \lambda_{\text{token}}\mathcal{L}_{\text{token}} + \lambda_{\text{occ}}\mathcal{L}_{\text{occ}} + \lambda_{\text{iou}}\mathcal{L}_{\text{iou}}. \tag{16}$$

In all experiments, we set $\lambda_{\text{bce}} = 1.0$, $\lambda_{\text{dice}} = 1.0$, $\lambda_{\text{token}} = 1.0$, $\lambda_{\text{occ}} = 0.05$, and $\lambda_{\text{iou}} = 0.05$, to balance segmentation fidelity, reasoning alignment, and temporal smoothness.

Training details. We train the model using the AdamW optimizer. Training is performed in `bfloat16` with a linear warmup of 100 steps followed by a decay over the full schedule. We adopt ZeRO stage 2, a per-GPU micro-batch size of 1 (due to memory constraints from high-resolution video inputs), and 16-step gradient accumulation. All segmentation supervision is applied at a resolution of 1024×1024.

Dataset ratio. We group the training data into five categories: semantic segmentation, referring image segmentation, reasoning-based image segmentation, RVOS and video VQA. During training, samples are drawn with category-wise sampling weights of $[4, 3, 1, 12, 1]$.

A.3. Training Stages

Alignment Stage. We freeze all modules except the STF, the LM head, and the LoRA adapters. A constant learning rate of 2×10^{-4} is used.

Few-Image Prediction Stage. We continue with the same learning rate of 2×10^{-4} and unfreeze the mask decoder, memory attention module, memory encoder, and the multimodal projector. This stage enables full segmentation capability while keeping the VLM backbone mostly stable.

Propagation Stage. Same freezing configuration as in Stage 2, but the learning rate is reduced to 1×10^{-5}. This stage additionally activates propagation-based supervision to refine temporal consistency.

Results after each training stage. Table 8 summarizes performance after Stages 1–3 on the MeViS valid_u split. Each stage provides consistent improvements, with the final stage achieving the highest $\mathcal{J}$, $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$ scores.

Table 8. Performance after each training stage on MeViS (valid_u).

Stage	MeViS (valid_u)		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
Stage 1	57.6	63.6	60.6
Stage 2	61.1	67.8	64.4
Stage 3	69.6	75.7	72.6

Table 9. Ablation on training stage combinations on MeViS (valid_u).

Training Stages	MeViS (valid_u)		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
Full Pipeline (Stage 1+2+3)	**69.6**	**75.7**	**72.6**
Without Stage 1 (Stage 2+3)	62.8	68.8	65.8
Stage 3 Only	62.4	69.1	65.8

Training process ablation study. Table 9 presents the ablation study on different training-stage configurations. For the first variant, we removed Alignment Stage (Stage 1) by replacing it with the same configuration as Few-Image Prediction Stage (Stage 2), unfreezing all modules from the beginning. For the Propagation Stage (Stage 3) only setting, both Alignment Stage and Few-Image Prediction Stage were replaced with the Propagation Stage configuration. Across all configurations, the total number of epochs and all hyperparameters were kept identical to ensure fair comparison. As shown in the table, incorporating all three stages described in Section 3.5 is crucial for achieving strong performance.

B. Architectural Details

B.1. Anchor Frame Selection

B.1.1. Training

Alg. 1 outlines the training-time anchor-frame selection strategy. We randomly sample up to α anchor frames $\mathcal{A}_{\text{train}}$ from a video and collect their local temporal neighbors as propagation frames, enabling the model to learn propagation cues while keeping memory usage feasible for high-resolution mask prediction.

Algorithm 1: Anchor-Frame and Propagation-Frame Sampling during Training

Input : Video length T_{seg}, maximum propagation window n_{prop}
Output: Anchor-frame index set $\mathcal{A}_{\text{train}}$, propagation-frame index set $\mathcal{I}_{\text{prop}}$
Select $\mathcal{A}_{\text{train}} \leftarrow$ randomly sample $\min(\alpha, T_{\text{seg}})$ distinct frame indices from $\{0, \ldots, T_{\text{seg}} - 1\}$;
Sort $\mathcal{A}_{\text{train}}$ in ascending order;
Initialize $\mathcal{I}_{\text{prop}} \leftarrow \emptyset$;
foreach $k \in \mathcal{A}_{train}$ **do**
  Add preceding frame indices $\{k-2, k-1\}$ that lie within range to $\mathcal{I}_{\text{prop}}$;
  Add succeeding frame indices $\{k+1, k+2, \ldots, k+n_{\text{prop}}\}$ that lie within range to $\mathcal{I}_{\text{prop}}$;
Remove any elements of $\mathcal{A}_{\text{train}}$ from $\mathcal{I}_{\text{prop}}$;
Sort $\mathcal{I}_{\text{prop}}$ in ascending order;
return $\mathcal{A}_{\text{train}}, \mathcal{I}_{\text{prop}}$

Specifically, for each anchor frame, we include two preceding frames and up to n_{prop} subsequent frames as propagation targets. In our setting, $\alpha = 3$ throughout training and $n_{\text{prop}} = 5$.

B.1.2. Inference

Algorithm 2: Anchor-Frame Selection and Update during Inference

Input : Video length T_{seg}, number of selected anchors α
Output: Anchor-frame index set $\mathcal{A}$, per-frame anchor subset $\mathcal{I}^{(t)}_{\text{Anchor}}$
Set $K \leftarrow \max(1, \lfloor T_{\text{seg}}/4 \rfloor)$;
Uniformly sample K anchor indices from $\{0, \ldots, T_{\text{seg}} - 1\}$ to form $\mathcal{A}$;
Sort $\mathcal{A}$ in ascending order;
for $t = 0$ **to** $T_{seg} - 1$ **do**
  Compute distances $d(k, t) = |k - t|$ for all $k \in \mathcal{A}$;
  Sort $\mathcal{A}$ by increasing $d(k, t)$;
  Select $\mathcal{I}^{(t)}_{\text{Anchor}} \leftarrow$ first $\min(\alpha, |\mathcal{A}|)$ elements;
return $\mathcal{A}, \{\mathcal{I}^{(t)}_{\text{Anchor}}\}_{t=0}^{T_{\text{seg}}-1}$

At inference, we uniformly sample an anchor frame set $\mathcal{A}$ from the T_{seg} frames. We set $|\mathcal{A}| = \max(1, \lfloor T_{\text{seg}}/4 \rfloor)$. Since T_{seg} is capped at 32, this yields at most 8 anchor frames for longer videos, corresponding to a stride of $\Delta T_{\text{seg}} = 4$.

For mask prediction at time t, we select the α anchor frames closest to t from $\mathcal{A}$ to form the set $\mathcal{I}^{(t)}_{\text{Anchor}}$, as detailed in Alg. 2.

B.2. Anchor-Frame Memory Attention

For each frame t, we construct a unified memory token sequence from the anchor frame index set $\mathcal{I}^{(t)}_{\text{Anchor}}$ and the FIFO index set $\mathcal{I}^{(t)}_{\text{FIFO}}$, which contains the indices of the most recent P frames. The anchor-frame memory captures long-range context, while the FIFO memory focuses on recent frames.

For anchor-frame memory attention, we assign a temporal index

$$\tau(k) = \begin{cases} 0, & k \in \mathcal{I}^{(t)}_{\text{Anchor}}, \\ 1, 2, \ldots, P, & k \in \mathcal{I}^{(t)}_{\text{FIFO}}, \end{cases} \tag{17}$$

which modulates a learned temporal positional encoding $\text{PE}(\tau(k))$.

The memory tokens are constructed as

$$\mathbf{H}_t = \left[\mathbf{h}_k + \text{PE}(\tau(k)) \right]_{k \in \mathcal{I}^{(t)}_{\text{Anchor}} \cup \mathcal{I}^{(t)}_{\text{FIFO}}}, \tag{18}$$

where $[\cdot]$ denotes concatenation along the token dimension.

Given the current-frame features $\mathbf{S}^{(t)}_{\text{seg}}$, anchor memory attention produces memory-conditioned features

$$\tilde{\mathbf{S}}^{(t)}_{\text{seg}} = \text{CrossAttn}\Big(\mathbf{S}^{(t)}_{\text{seg}},\ \mathbf{H}_t\Big), \tag{19}$$

where $\tau(k) = 0$ encodes invariant anchor context and $\tau(k) > 0$ captures recency-aware FIFO cues.

B.3. Frame-Aware Video Tokenizer

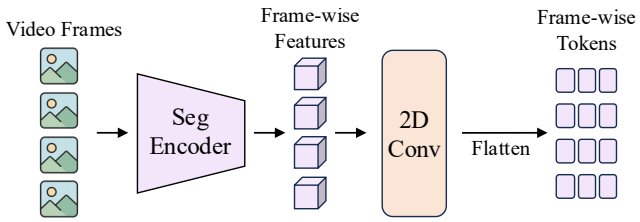


Figure 6. Frame-Aware Video Tokenizer architecture.

As shown in Fig. 6, we extract frame-wise segmentation features using the vision encoder of the segmentation model:

$$\mathbf{S}_{\text{seg}} \in \mathbb{R}^{H' \times W' \times T_{\text{seg}} \times C}. \tag{20}$$

Each feature map is downsampled through three 3×3 stride-2 convolutions. In our setting, $H' = 64$, $W' = 64$ and $C = 256$, so applying three successive $\frac{1}{2}$-scale convolutions reduces the spatial size to $H'/8 = 8$:

$$\mathbf{S}_{\text{down}} \in \mathbb{R}^{8 \times 8 \times T_{\text{seg}} \times C}. \tag{21}$$

The 8×8 grid is flattened into 64 spatial tokens per frame and projected into D dimensions:

$$\mathbf{S}_{\text{patch}} = \text{Linear}(\text{reshape}(\mathbf{S}_{\text{down}})) \in \mathbb{R}^{T_{\text{seg}} \times 64 \times D}. \tag{22}$$

Table 10. Efficiency of VIRST with different α.

α	Efficiency	
	FPS↑	Memory (GB)↓
2	5.14	37.52
4	5.04	37.52
6	4.98	37.52

Table 11. Inference speed comparison across methods.

Method	FPS↑
VISA-7B [14] w/o postproc.	1.47
VRS-HQ-7B [12]	3.81
HyperSeg-3B [13]	1.54
VIRST-7B ($\alpha = 3$, Ours)	**5.10**

Finally, the temporal and spatial axes are merged to construct the video-token sequence used in the cross-attention mechanism with $\mathbf{E}_{\text{ST}} \in \mathbb{R}^{N \times D}$:

$$\mathbf{S}_{\text{vid}} \in \mathbb{R}^{(T_{\text{seg}} \times 64) \times D}. \tag{23}$$

C. Efficiency Analysis

C.1. Effect of α on Performance and Efficiency

We analyze the effect of α on performance and efficiency. As shown in Tab. 10 and Tab. 6, increasing α improves performance with only minor impact on efficiency. Peak memory remains nearly constant (within < 0.01 GB), while FPS decreases slightly (from 5.14 to 4.98 as α increases from 2 to 6).

C.2. Inference Efficiency Comparison

We compare the inference efficiency of VIRST with existing methods in Tab. 11. We report FPS on the MeViS dataset using a single A100 GPU. VIRST achieves 5.10 FPS, compared to 3.81 FPS for VRS-HQ-7B and 1.47 FPS for VISA-7B. These results indicate that VIRST maintains competitive efficiency while performing joint reasoning and segmentation.

D. Qualitative Results

D.1. Video Segmentation Qualitative Results

Fig. 7 provides additional qualitative results for video segmentation, demonstrating strong performance across challenging cases such as multi-object scenes, heavy distractors, and small-object targets.

D.2. Image Segmentation Qualitative Results

Fig. 8 presents additional image reasoning-segmentation results, showing that the model can accurately localize objects even under complex, fine-grained textual descriptions.

D.3. Video Understanding Qualitative Results

Fig. 9 demonstrates that VIRST retains strong video understanding capability, with responses generated autoregressively.

D.4. Failure Cases

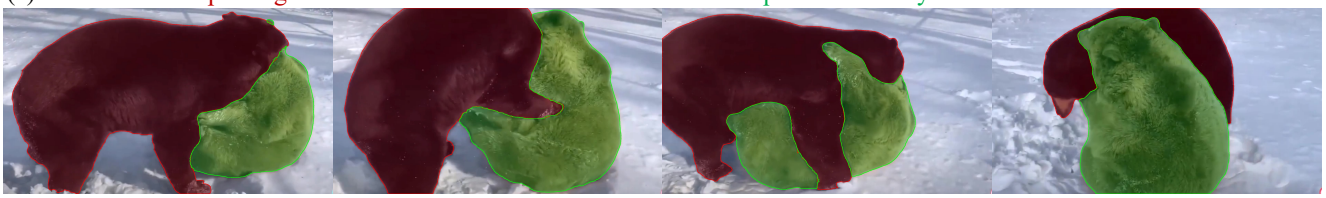
Fig. 10 shows failure cases of VIRST. In Fig. 10 (a), the scene contains many visually similar distractors, making the scenario inherently difficult. Although the queried object and its defining motion appear in the first frame, the target moves rapidly and undergoes heavy occlusions. VIRST initially tracks it, but the mask gradually drifts toward a similar distractor and eventually switches to it.

Fig. 10 (b) requires multi-step semantic reasoning: the task is to segment only the dice showing 3 and 5 (prime numbers). VIRST struggles to maintain this constraint over time, intermittently masking the die showing 6 and failing to consistently retain the mask for 5 in the final frame.

E. Limitations and Future Directions

While VIRST demonstrates strong performance in complex scenes and reasoning-intensive queries, several limitations remain. As shown in Appendix D.4, the model still struggles in highly cluttered environments with many distractor objects, and it can fail when the query requires multi-step semantic reasoning. Future work should explore training strategies that more explicitly ground step-by-step reasoning in video inputs, enabling tighter integration between pixel-level visual understanding and compositional language reasoning, and allowing VLMs to extend more effectively to long-video segmentation and complex video scenarios.

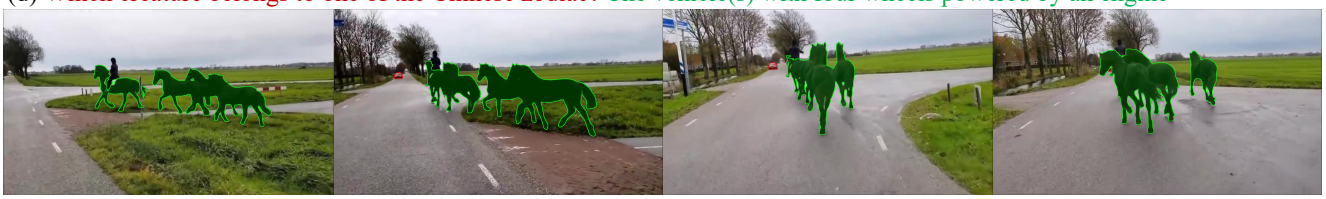
(a) The bear that is pinning down the other bear The bear that has been pinned down by the other bear



(b) A flock of moving poultry

(c) The person with the lizard in their hand The first lizard to be taken and gripped by hand

(d) Which creature belongs to one of the Chinese zodiac? The vehicle(s) with four wheels powered by an engine



(e) The dog(s) with the fluffiest fur The dog that sniffed the truck bed and then crawled under its companion

(f) Which animal(s) is/are herbivorous? Obstruction(s) that play the role of separation and guidance
Which mammal(s) of the three is/are primate(s)?

Figure 7. **Qualitative results of VIRST on videos.** Results are best viewed when zoomed in.

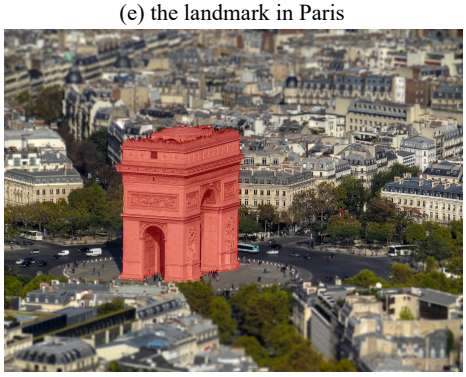
(a) If we were preparing for a road trip and wanted a vehicle that could transport our belongings while also offering a place to sleep, what type of vehicle in the picture would be best suited for this purpose?

(b) Helmets can effectively protect the safety of the head during cycling, but strong winds during high-speed cycling can affect the eyes. What part in the picture can protect the safety of the eyes?

(c) In order to safely transport personal belongings while riding a bicycle, what object should be used to secure and carry the items?

(d) When working at a desk, maintaining a comfortable seating position is critical. What item in the picture could be utilized to provide a raised seating area for added comfort?

(e) the landmark in Paris



(f) something that falls from the gun after shooting

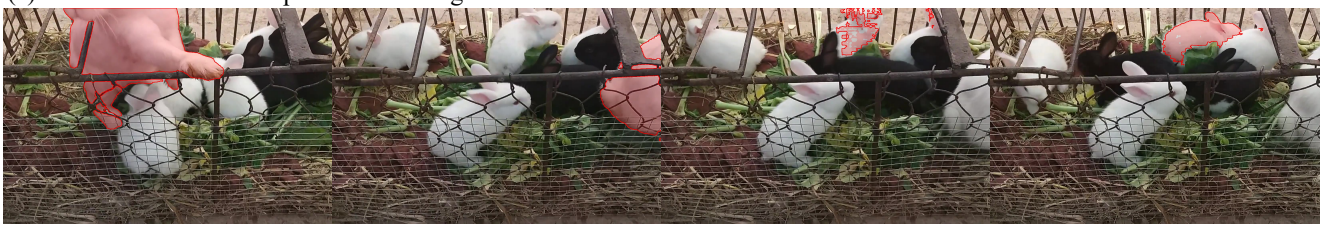
Figure 8. **Qualitative results of VIRST on images.** Results are best viewed when zoomed in.

(a) Q: Offer a detailed interpretation of the video's message and imagery.
A: The video continues with a close-up of the diced white onions sizzling in the black non-stick frying pan on the gas stove. The red circular heat indicator is still visible at the center of the pan. The onions are now more translucent, indicating they are further along in the cooking process. A hand holding a red spatula appears, stirring the onions to ensure even cooking. The spatula moves the onions around, and some pieces start to turn golden brown. The background remains consistent, showing parts of the gas stove and kitchen countertop. The video concludes with the onions continuing to cook, becoming more golden and caramelized.

(b) Q: Share a detailed breakdown of the video's storyline and visuals.
A: The video continues with the person sitting on the couch, still wearing a gray hoodie and shorts, and interacting with the plastic bag. The laptop with a bright screen remains visible in the background, along with some cables and various items scattered around. The person is seen placing a yellow package on the couch and then picks up a red package from the floor, examining it closely. The can that was previously added to the collection of items is also visible on the couch. The person continues to methodically organize the contents of the bag, adding more items to the pile on the couch and the floor around them. The scene maintains its focus on the person's actions and the homey atmosphere throughout the video.

Figure 9. **Qualitative video understanding results of VIRST.**

(a) The rabbit with its feet placed on the cage.



(b) Which die/dice points are prime numbers?

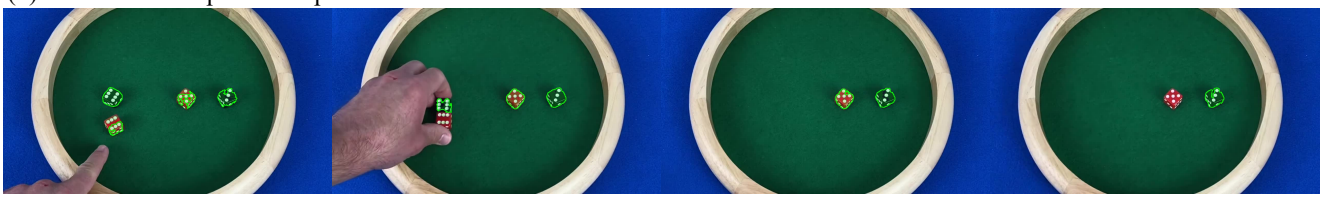


Figure 10. **Failure cases of VIRST.**