Towards Streaming Referring Video Segmentation via Large Language Model

Supplementary Material

Table 1. Composition of datasets during training.

Type	Dataset
Image Segmenation	RefCOCO (17K), RefCOCO+ (17K) RefCOCOg (22K), Grand-f (214K)
Image QA	LLaVA 1.5 (665K)
Video Segmentation	Ref-YTVOS (3.5K), MeVIS (0.6K) ReVOS (1.7K), Ref-SAV (37K)

Table 2. Ablation study of training sample of sequence length on Ref-DAVIS, MeViS(val^u) and REVOS(Referring) datasets.

Sample per sequence	Ref-DAVIS	MeViS(val^u)	REVOS
1 frame	74.6	56.1	58.1
3 frames	75.8	58.4	58.6
5 frames	76.4	59.4	59.9
8 frames	76.1	59.2	60.1
12 frames	74.7	56.9	58.9

1. Additional Methods

1.1. Extra instruction

As previously discussed, in both the training and inference pipelines, when the OMCP strategy triggers a re-invocation of the MLLM, additional instructional prompts are introduced to guide the model toward attending to contextual information. Specifically, the model input is formulated as:

$$\text{Input} = I_i + R + E + \texttt{[INFO]},$$

where E represents the embedding of the additional instructional prompt. Accordingly, during training, the corresponding input token IDs are adjusted to incorporate these augmented prompts.

1.2. Training dataset

Table 1 summarizes the datasets employed during our training phase. Overall, our data configuration aligns with that of Sa2VA, except that we exclude data from the VideoQA task. Notably, for ablation studies against the baseline, we retrain Ss2VA using the identical dataset configuration to ensure a fair and controlled comparison.

2. More analysis

2.1. Ablation study

2.1.1. Training sequence sample.

The length of the training sequence encompasses temporal information of varying richness, which is critical for video-based tasks. It can be observed from Table 2 that, RVOS performance gradually improves as sequence length increases. However, beyond a certain threshold ($i, e, 8$), further extension of the sequence yields diminishing returns in performance gain while significantly increasing training time. We hypothesize that when frame sampling becomes overly dense, the inter-frame visual variations diminish, rendering the re-invocation of the MLLM to generate updated segmentation tokens less effective for foreground segmentation refinement.

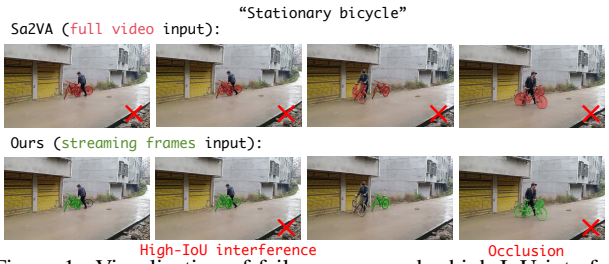

Figure 1. Visualization of failure cases under high-IoU interference and occlusion.

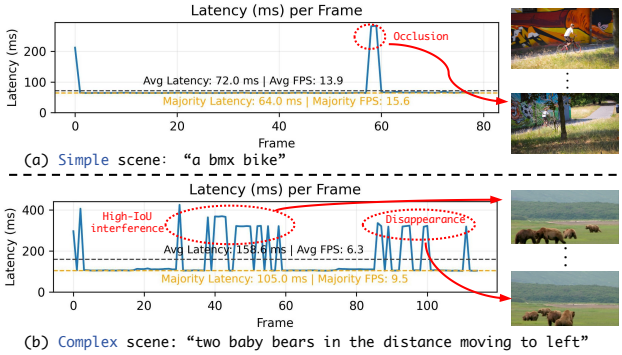

Figure 2. Distribution of inference latency across diverse scenarios on consumer-grade hardware.

2.2. FPS fluctuation analysis

We illustrate the FPS fluctuations in Fig. 2. In simple cases where [SEG] updates are not required, our model achieves around 15 FPS. Under complex cases (*e.g.*, intricate expressions and high-IoU distractors), our framework invokes the OMCP strategy. While this enhances accuracy, it involves extra MLLM calls that incrementally raise inference time. Under offline inference, our method achieves an average speed comparable to the baseline at approximately 14 FPS. Notably, while the baseline supports only offline pipelines, our approach remains effective for both offline and streaming settings.

2.3. Failure case analysis

Furthermore, we provide an analysis of failure cases, as illustrated in Fig. 1 and Fig. 3. Fig. 1 analyzes failure cases. High-IoU distractors delay OMCP triggering, while complete occlusion leads to intermittent SER failures. Despite these challenges, our streaming pipeline re-initiates perception and recovers, whereas offline methods cannot correct early errors, causing failures throughout the sequence. Since our input is processed in a streaming fashion, the model struggles during the early stages of the sequence on reasoning-intensive segmentation tasks that inherently require a global contextual understanding. As demonstrated in the Fig.3, the task requires segmenting the rider currently in the leading position. In the initial phase of the sequence, our method promptly identifies and segments the rider who is leading at that moment. As the sequence progresses and a new rider assumes the lead, our method dynamically adapts and switches its segmentation target accordingly. In contrast, the Sa2VA method, benefiting from its offline processing paradigm, can leverage global spatiotemporal context across the entire video to perform holistic reasoning, thereby correctly identifying and segmenting the rider who is initially trailing but eventually takes the lead. Owing to the streaming nature of the input, our method is inherently constrained to perceive only the current and past frames, limiting its access to future contextual information. Consequently, its performance on reasoning-intensive segmentation tasks, originally designed for offline video processing, remains suboptimal and warrants further improvement.

2.4. More visualization

We provide additional visualizations in Fig. 4 to further illustrate the advantages of our approach. In the upper part of Fig. 4, the language instruction specifies the segmentation of *all* zebras. While the Sa2VA method produces incomplete and semantically fragmented masks beyond the initial video segment, our method consistently maintains semantically complete and coherent segmentation throughout the entire sequence. The Fig. 4 (b) depicts scenarios involving occlusion and identity switches. Owing to our SER mechanism, our approach preserves semantic consistency even under challenging conditions where target entities become visually entangled or ambiguous.

Figure 3. Failure case analysis.

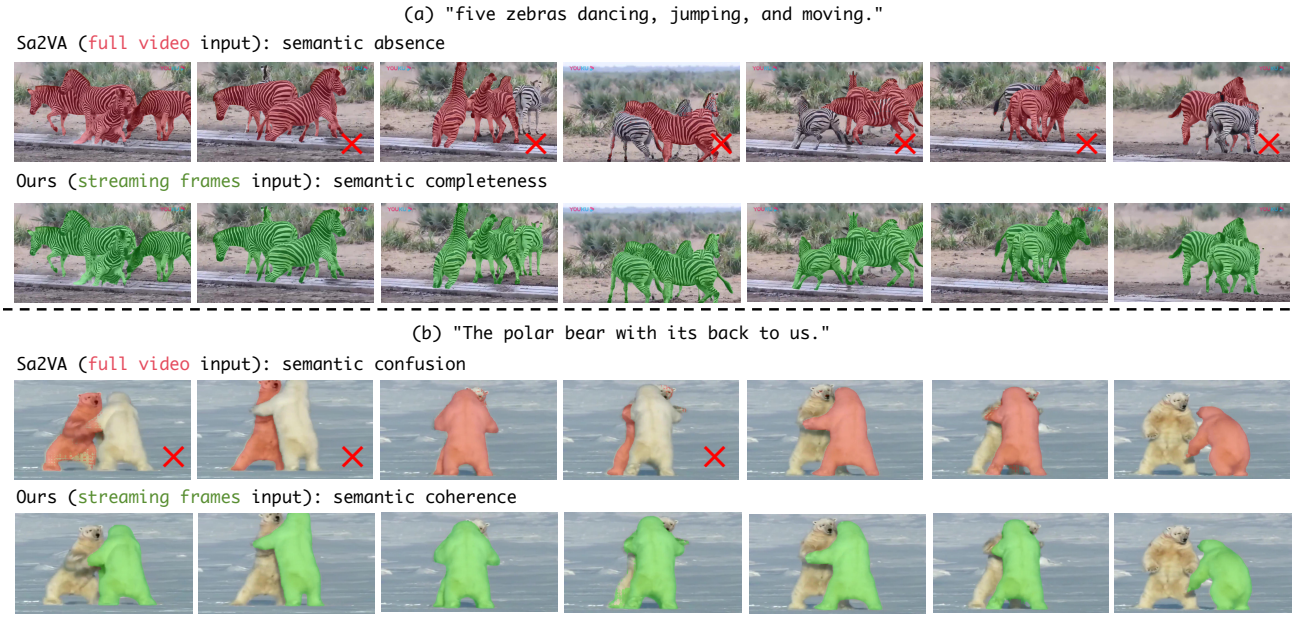

Figure 4. Segmentation map comparison of our StreamRVOS and Sa2VA. We employ a streaming inference pipeline, whereas Sa2VA adopts an offline processing paradigm.