
Omni-R1: Appendix

Anonymous Author(s)

Affiliation

Address

email

A Appendix Overview

This appendix provides additional details on the experimental setup, model architecture along with training pipeline, and supplementary results that support the findings presented in the main paper.

- **Implementation Details:** this section details additional aspects of our method: our two-system architecture (including user instructions for its MLLM components and prompts for the downstream SAM2 model), our reward design, and the differences between training and inference procedures.
- **Ablation Studies:** this section provides ablation studies of System 1 on the reward component and dataset selection.
- **Visualization Results:** this section provides more visualization results, including examples in comparison with other methods and failure case analysis.
- **More Analysis:** this section provides analysis on the hallucination issue and the influence of resolution on general video understanding tasks.
- **Limitations and Future Work:** this section discusses the limitations of our method and potential future work.

B Implementation Details

User Instructions on Two Systems. To enable MLLMs to perform keyframe selection and referred object captioning, we designed the prompt as shown in the figure 1. We formulate keyframes as time duration segments and assign spatial description text to each duration. Additionally, we observed that during training, the model could be influenced by the timestamp patterns seen in the prompt examples. Therefore to increase the diversity of keyframe distributions during training and prevent the model from overfitting to specific timestamps, we incorporated randomized timestamps into the prompt, encouraging the model to focus on learning keyframe selection and caption rather than simply copying training timestamps. For AVS tasks, we designed a similar prompt (see Figure 2) to guide the model in analyzing the audio content and identifying the corresponding visual grounding description. The prompt emphasizes the need to avoid temporal expressions and instead focus on visual cues.

The intermediate results are then interpreted as frames and paired descriptions before being fed into System 2. The prompt for System 2 follows the official grounding prompt of Qwen2.5-VL, where the output is a list of bounding boxes and their corresponding labels in JSON format.

Prompt Design for SAM2 as Downstream Segmenter. Once the keyframe grounding results are obtained from System 2, we assign a unique identifier to each detection result using a tuple format: (roll_out_idx, frame_idx, pred_obj_idx, bbox). This ensures that any detection within a single GRPO group can be uniquely referenced.

Since all detections within a group share the same input context, we optimize inference efficiency by processing all detection results in a single forward pass. Specifically, we feed the entire video segment into SAM2, and for each detection tuple, we assign a unique object ID. These object IDs are used as conditioning inputs to SAM2 to obtain their respective segmentation masks.

Specifically, we maintain a mapping dictionary between detection tuples and assigned object IDs $\mathcal{P} : \text{tuple} \rightarrow \text{obj_id}$, which enables us to reverse-map SAM2’s outputs $\{\hat{\mathcal{M}}_{\text{obj_id}} | \text{obj_id} \in \mathcal{P}(\text{tuple})\}$ back to the original detection structure $\hat{\mathcal{M}}_{\text{tuple}}$. The segmentation results are then matched to the corresponding grounded predictions and used for reward evaluation.

Reward Design of $R_{\mathcal{K}}$ To enable the model to learn from diverse keyframe selections, we design an evaluation reward function, $R_{\mathcal{K}}$, which assesses both the diversity and quality of the chosen keyframes. This function is formulated as:

$$R_{\mathcal{K}} = \lambda_1 R_{\text{diversity}} + \lambda_2 R_{\text{num}} + \lambda_3 R_{\text{saliency}}$$

where the last component, R_{saliency} , has been clarified in the main paper. Here, we detail the first two reward components.

Firstly, to discourage the model from selecting temporally adjacent keyframes, which can lead to redundant System 2 inference, we introduce a distribution reward function, $R_{\text{diversity}}$. This component evaluates the distributional diversity of keyframes by calculating the temporal intervals between them. Specifically, all keyframes are sorted in chronological order. We then compute the temporal interval $t_{i+1} - t_i$ between each pair of consecutive keyframes. The final $R_{\text{diversity}}$ value is subsequently determined based on the collection of all such inter-frame temporal intervals.

The diversity reward $\mathcal{R}_{\text{diversity}}(\mathcal{S})$ can be defined as the sum of an overlap punishment term and a distribution reward term:

$$\mathcal{R}_{\text{diversity}}(\mathcal{S}) = \text{overlap_punish} \cdot |\mathcal{I}| + \text{dist_reward} \cdot |\mathcal{D}|$$

where:

- \mathcal{S} is the set of selected items. Let $\mathcal{S}_{\text{sorted}} = (s_1, s_2, \dots, s_M)$ be the sequence of $M = |\mathcal{S}|$ items sorted according to the relevant criteria (e.g., timestamps).
- \mathcal{I} is the set of indices i for which an "overlap" condition is met between s_i and s_{i+1} . Specifically, assuming $\text{idx}(s_j)$ gives an identifier for item s_j :

$$\mathcal{I} = \{i \in \{1, \dots, M-1\} \mid \text{idx}(s_i) = \text{idx}(s_{i+1})\}$$

$|\mathcal{I}|$ is the number of such identified overlaps (e.g., pairs of consecutive items with identical identifiers).

- \mathcal{D} is the set of indices of items in $\mathcal{S}_{\text{sorted}}$ that are not considered the start of an overlap as defined by \mathcal{I} :

$$\mathcal{D} = \{j \in \{1, \dots, M\} \mid j \notin \mathcal{I}\}$$

Therefore, $|\mathcal{D}| = M - |\mathcal{I}|$.

- overlap_punish is the coefficient for the punishment. For this term to act as a punishment, overlap_punish should typically be a negative value (e.g., -0.2), or if it’s a positive value, it should be subtracted from the reward.
- dist_reward is the coefficient for the reward given to items not initiating an overlap.

The formula $\mathcal{R}_{\text{diversity}}(\mathcal{S})$ can also be written as:

$$\mathcal{R}_{\text{diversity}}(\mathcal{S}) = (\text{overlap_punish} - \text{dist_reward}) \cdot |\mathcal{I}| + \text{dist_reward} \cdot M$$

Reward Design of $R_{\mathcal{A}}$ The specific formulation of $R_{\mathcal{A}}$ is as follows:

$$R_{\mathcal{A}} = \frac{1}{K} \sum_{i=1}^K (1 - \mathcal{L}_{\text{Hungarian}}(\mathcal{B}_{s_i}, \mathcal{B}_{s_i}^*))$$

where \mathcal{B}_{s_i} denotes the set of predicted bounding boxes at the s_i -th frame, and $\mathcal{B}_{s_i}^*$ represents the corresponding set of ground truth bounding boxes. The function $\mathcal{L}_{\text{Hungarian}}$ refers to the Hungarian matching loss [1, 2], and K is the total number of selected keyframes.

The Hungarian matching loss $\mathcal{L}_{\text{Hungarian}}$ is computed based on the Intersection-over-Union (IoU) between predicted and ground truth bounding boxes. Specifically, a cost matrix \mathcal{M} is first constructed using the IoU values between each pair of predicted and ground truth boxes. Then, the Hungarian algorithm is applied to the negative matrix $-\mathcal{M}$ to obtain the optimal one-to-one matching that minimizes the total negative cost, which corresponds to maximizing the overall IoU-based matching accuracy.

Reward Design of $R_{\mathcal{G}}$ For $R_{\mathcal{G}}$, we adopt a simple aggregated IoU as the reward function. Specifically, for each detected object, we accumulate the predicted segmentation masks across all objects to construct a per-frame mask set $\hat{\mathcal{M}}_t$. Then, we compute the Intersection-over-Union (IoU) between the predicted masks and the corresponding ground truth masks \mathcal{M}_t^* on each frame. The final reward is obtained by averaging the IoU values across all frames.

$$R_{\mathcal{G}} = \frac{1}{T} \sum_{t=1}^T \text{IoU}(\hat{m}_t, m_t^*)$$

Training and Inference Strategy For video clips, we first feed them into System 1 at a relatively low resolution of a per-frame pixel $128 \times 28 \times 28$, which allows us to process longer video segments during training and inference. Then System 2 predicts detection results at a higher resolution of $900 \times 28 \times 28$. **For VOS tasks**, we adopt a random uniform sampling strategy during training, selecting between 8 and 24 frames per video to enhance temporal diversity and robustness. All SAM2-based segmentation and reward evaluations are then applied to these resampled clips at their original input resolution. **For RefAVS tasks**, we observed severe cross-modal hallucination issues during preliminary experiments, particularly when reasoning jointly over full-length audio and multi-frame video inputs. To mitigate this, we introduce a simplified variant, RefAID (Referring Audio-Image Detection), which reduces the AVS problem to object detection using only the first video frame and the corresponding full audio query. In this setting, no SAM2 segmentation is used; training is driven solely by detection-based rewards.

During inference, we resample a fixed maximum of 24 frames per video for VOS tasks. Unlike training, segmentation and evaluation are conducted over the full video sequence using SAM2 to align with standard benchmark protocols. For RefAVS, we adopt the same resampling and evaluation procedure as in VOS, ensuring consistency across task settings.

C Ablation Studies

We conduct an ablation study to investigate the effect of progressively designed reward components $R_{\mathcal{K}}$ (keyframe coverage), $R_{\mathcal{A}}$ (alignment via Hungarian matching), and $R_{\mathcal{G}}$ (global grounding IoU) on the overall performance. To ensure a fair comparison, all models are trained for one epoch on the ReVOS and MeVIS datasets and are constrained to select exactly four keyframes unless otherwise noted.

Table 1 reports the results across four evaluation subsets of ReVOS: referring, reasoning, single-object, and multi-object. We observe that the combination of $R_{\mathcal{K}} + R_{\mathcal{G}}$ achieves the highest overall score (39.9%), outperforming the full combination $R_{\mathcal{K}} + R_{\mathcal{A}} + R_{\mathcal{G}}$ (38.4%). This suggests that the inclusion of $R_{\mathcal{A}}$ may introduce instability rather than improvement.

We hypothesize that this is due to the nature of $R_{\mathcal{A}}$, which relies on Hungarian matching over temporal sequences. Given that the segmentation model SAM2 already incorporates strong temporal priors, the additional alignment-based reward may not effectively capture useful gradients and could introduce variance from imperfect IoU estimation. In contrast, $R_{\mathcal{G}}$, which aggregates IoUs across frames, directly reinforces temporal consistency and spatial correctness, leading to more stable learning dynamics.

Interestingly, $R_{\mathcal{K}}$ alone provides a surprisingly strong baseline (35.2%), demonstrating that ensuring keyframe coverage is already beneficial. However, only when combined with $R_{\mathcal{G}}$ do we observe consistent improvements across all task types, including reasoning (34.2%) and multi-object scenarios (42.6%).

Prompt for OMNI-R1 as System 1

- Given a `[frames]` seconds video and a reference instruction: `[ref_prompt]` that may involve temporal behavior, identify the exact object(s) `[ref_prompt]` in the video that matches the description.
- Select about 4 most relevant moments that contain the referred object(s) with the best view.
- Then, simplify the identified object into a short and clear visual grounding description that can be used for single-image reference at each moment.
- Avoid temporal phrases and comparison phrases like “walking”, “moving”, “approaching”, “bigger” or “smaller”, but instead describe visible visual cues like clothing, pose, position, or grouping.
- Try to select moments that are **temporally well-distributed across the video**, rather than clustered in the same part of the timeline. Avoid selecting multiple timestamps that are adjacent or overlapping; instead, prefer clearly distinct moments that each offer unique visual information. It is better to choose the most relevant and highly representative moments **spanning the entire video**, rather than picking all from the beginning.
- Explain your reasoning in `<think></think>` and output the final result in `<answer></answer>`. Your final answer should be a JSON object in the following format:

```
<think> your analysis about the video and reference instruction </think>
<answer>
{
  "start_time": "00:[start]",
  "end_time": "00:[end]",
  "description": "direct description of referred object(s) at this moment"
}
</answer>
```

Figure 1: Keyframe selection and recaptioning prompt for System 1.

Prompt for OMNI-R1 as System 1

- Given a `[audio_duration]` audio and a reference instruction: `[ref_prompt]`, which involves temporal and audio-related behavior, first analyze the objects in the image that are producing sound, including both human voices and instrument sounds.
- Based on the audio content, identify the exact object `[ref_prompt]` in the image that matches the audio.
- Then, simplify the identified object into a short and clear visual grounding description that can be unambiguously recognized in a single image without relying on audio.
- Avoid using temporal expressions such as “playing” or “moving”; instead, describe visible visual cues such as clothing, pose, position, or grouping.
- Explain your reasoning in `<think></think>` and output the final result in `<answer></answer>`.

Figure 2: Audio analyzing and recaptioning prompt for System 1.

119 Lastly, we include a model pre-trained with 2,000 samples from the RefCOCOg grounding dataset
 120 as a reference upper bound. This model achieves 44.7% overall, demonstrating the potential of our
 121 design in VOS tasks lies in the grounding capabilities of the model.

122 These findings validate the effectiveness of R_G as a grounding-aware reward and highlight the
 123 limitations of alignment-based matching in the presence of strong perceptual priors.

124 D Visualization Results

125 **Mask Quality** Since our method utilizes the SAM2 model for segmentation, without fine-tuning
 126 mask decoder, the final mask output is more stable than those methods that rely on additional training
 127 on segmentation mask decoder. As can be seen in Figure 3a, in this simple example, our method is
 128 able to segment the target object with a mask consistent with the ground truth, while Sa2VA predicts
 129 the right target but generates a mask with holes and noise.

Table 1: Ablation study on the reward function R_K , R_A and R_G for System 1. The first model is trained with additional 2,000 samples from grounding dataset refcocog.

Method	ReVOS				
	<i>Referring</i>	<i>Reasoning</i>	<i>Single</i>	<i>Multi</i>	<i>Overall</i>
Omni-R1 + <i>refcocog</i>	52.5	36.9	45.0	46.6	44.7
$R_K + R_A + R_G$	44.2	32.5	38.2	41.9	38.4
$R_K + R_G$	45.5	34.2	39.6	42.6	39.9
$R_K + R_A$	43.1	29.5	36.8	37.5	36.3
R_K	44.1	26.2	36.6	34.1	35.2

Temporal Reasoning Our System 1 leverages temporal reasoning to improve segmentation accuracy. As can be seen in Figure 3b, in this example, one has to **watch the whole video and analyze the video context to make a correct prediction** about the next bottle to be picked up. Our method is able to select the bottle that is about to be picked up, while Sa2VA fails to do so and segments the bottle that is already picked up. A similar case is shown in Figure 4a, where one has to identify the object moving fastest in the view and our method selects the right object while Sa2VA fails to do so. Both case show that our method is able to leverage temporal reasoning to improve the segmentation accuracy.

Detail Understanding Our System 1 leverages detail reasoning to improve segmentation accuracy. Figure 3c shows a scenario where detail reasoning is needed to figure which wineglass is likely will be finished first. Our System 1 is **already able to select the right wineglass but still makes a loose description for System 2 to analyze detail information**, while Sa2VA fails to understand the instruction and segments all the wineglasses. This shows that our System 1 is able to delay detail reasoning for System 2 for detail understanding to improve the segmentation accuracy.

E More Analysis

Hallucination Analysis During the training of our RefAVS task, we identified a significant hallucination problem, which we attribute to the complexity of multi-modal video and audio inputs. To systematically evaluate this issue, we conducted targeted assessments on audio-related hallucinations using the JUDGE subset of AVHBench [3], the first comprehensive benchmark designed to evaluate the perception and comprehension abilities of audio-visual large language models (LLMs).

As shown in Table 2, our base model (Qwen2.5-Omni-7B) achieves an accuracy of 58.47% on the JUDGE subset. Training on 1600 AVS samples leads to a modest improvement (60.77%), which is further enhanced to 61.51% by applying the GRPO KL loss with a reduced coefficient ($\beta = 0.001$). Notably, increasing the AVS training samples to 10400 does not yield better results, suggesting potential overfitting or task imbalance.

On the other hand, training with VOS tasks alone significantly boosts accuracy to 66.01%, and the best performance (71.88%) is obtained by jointly training on both AVS and VOS tasks. This represents a substantial improvement of 13.41% over the base model, demonstrating that multi-task training not only enhances audio-visual grounding but also mitigates hallucination issues more effectively.

These results confirm the effectiveness of leveraging task diversity and balanced reward optimization in improving the robustness of multimodal reasoning.

Video Resolution Influence on General Video Understanding Tasks To evaluate the influence of input resolution and prompting strategy on general video understanding, we compare model performance across different configurations on the VideoMME and MVBench benchmarks, as summarized in Table 3. All models are evaluated under two resolution settings: the default resolution of $128 \times 28 \times 28$ and a higher resolution of $256 \times 28 \times 28$ (denoted with *), with and without the proposed *thinking* prompting strategy.

We observe that increasing the input resolution consistently leads to performance gains across all models. For instance, Qwen2.5-Omni improves from 66.1% to 67.0% on MVBench when evaluated at higher resolution. Similarly, our Omni-R1-1600 model benefits from the resolution increase,

Table 2: Performance on AVHBench (JUDGE subset, total 5302 samples). In the table, AVS tasks are trained on RefAVS dataset and VOS tasks are trained on ReVOS and MeViS datasets. The default GRPO KL loss weight $\beta = 0.04$.

Method	AVHBench JUDGE	
	Correct Answers	Accuracy
Base Model	3100	58.47%
AVS 1600 samples	3222	60.77%
AVS 1600 samples with $\beta = 0.001$	3261	61.51%
AVS 10400 samples	3120	58.85%
VOS	3500	66.01%
AVS and VOS	3811	71.88%

Table 3: Performance comparison with different resolution and thinking prompt on VideoMME and MVBench. Resolution refers to either $128 \times 28 \times 28$ (default) or $256 \times 28 \times 28$ (high). 28 is the CNN patch size. Thinking prompt is an additional reasoning cue. The metric is the average of \mathcal{J} and \mathcal{F} scores (%).

Model	Resolution	Thinking	VideoMME		MVBench Avg
			General	Short	
Qwen2.5-Omni	$128 \times 28 \times 28$	No	58.3	69.8	66.1
Qwen2.5-Omni	$256 \times 28 \times 28$	No	58.7	69.9	67.0
Qwen2.5-Omni	$128 \times 28 \times 28$	Yes	59.3	70.1	68.1
Qwen2.5-Omni	$256 \times 28 \times 28$	Yes	59.8	70.9	68.3
Omni-R1-AVS	$128 \times 28 \times 28$	No	59.0	71.9	68.3
Omni-R1-AVS	$256 \times 28 \times 28$	No	59.4	71.9	68.7
Omni-R1-AVS	$128 \times 28 \times 28$	Yes	59.9	72.1	69.4
Omni-R1-AVS	$256 \times 28 \times 28$	Yes	60.0	72.1	69.5
Omni-R1-VOS	$128 \times 28 \times 28$	No	59.7	72.3	68.9
Omni-R1-VOS	$256 \times 28 \times 28$	No	59.6	72.5	68.9
Omni-R1-VOS	$128 \times 28 \times 28$	Yes	59.8	72.5	69.8
Omni-R1-VOS	$256 \times 28 \times 28$	Yes	60.1	72.8	69.9
Omni-R1-VOS-AVS	$128 \times 28 \times 28$	No	60.1	72.5	69.1
Omni-R1-VOS-AVS	$128 \times 28 \times 28$	Yes	60.7	73.0	70.3

170 achieving a performance gain from 68.3% to 68.7%. These improvements suggest that higher spatial
 171 resolution enhances the model’s ability to capture fine-grained visual details, particularly beneficial
 172 for multi-object reasoning and scene comprehension.

173 In addition to resolution, the *thinking* prompt—designed to guide the model toward structured
 174 multi-step reasoning—further boosts performance across all tested models. Omni-R1-AVS with
 175 *thinking* achieves 69.4% on MVBench, outperforming its baseline by 1.1%. The combination of both
 176 higher resolution and *thinking* yields the best results overall, with Omni-R1-VOS-AVS + *thinking**
 177 reaching 60.1% on VideoMME (general) and 69.9% on MVBench. This indicates that resolution and
 178 prompting act as complementary strategies: resolution improves visual precision, while prompting
 179 enhances reasoning capability.

180 But we can also conclude that the performance gain obtained by increasing video resolution is faint,
 181 which suggests that in general understanding benchmarks, the impact of resolution may be limited
 182 and temporal understanding is more important compared to fine-grained spatial details. This is
 183 also consistent with our findings in the main paper, where we observed that the temporal reasoning
 184 capability of our model is significantly enhanced by the dual-system design and gain much advantage
 185 on reasoning VOS tasks.

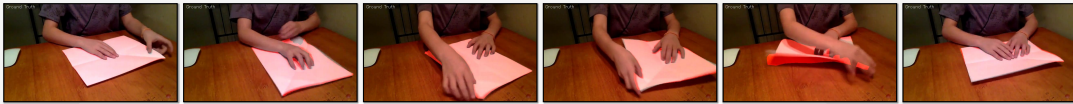
F Limitations and Future Work

Limitations Although our dual-system design significantly enhances the temporal reasoning capability of **System 1**, the complete functional decoupling between System 1 and System 2 introduces certain limitations. In particular, **System 2** lacks temporal context, which may affect consistency in temporally coherent tasks. This consideration partially motivates our selection of **VOS** as a primary training task: while VOS emphasizes temporal consistency, it also provides dense per-frame annotations that allow us to design stable training strategies to mitigate the context gap—such as frame-wise Hungarian matching loss and aggregated mask-based rewards. During inference, the missing temporal cues in System 2 are partially recovered through SAM2’s mask-based processing.

However, when extending to tasks requiring finer temporal sensitivity such as detecting and describing localized anomalous behaviors within a specific time span, our current architecture faces new challenges. While **System 1** can still progressively narrow down relevant temporal segments, the need for dynamic temporal granularity exposes the limitations of a fully decoupled, one-way reasoning architecture.

Discussion on Future Work Our coarse-to-fine reasoning pathway aligns closely with human cognitive intuition, yet we recognize the crucial importance of enabling a bidirectional flow of information between global and local levels. Our current design can be viewed as a context-constrained tree search structure, where only the root node (**System 1**) has access to full contextual information, while downstream nodes (**System 2**) operate solely on partial, local input. This diverges from typical hierarchical systems, which often allow child nodes to access aggregated information from their parent nodes.

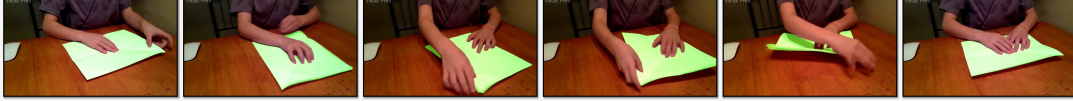
While the VOS task structure inherently compensates for this limitation by providing dense temporal supervision, to further enhance the flexibility of our two-system framework, future work should explore more interactive architectures that facilitate explicit information exchange between **System 1** and **System 2**. Supporting backtracking within the reasoning tree would empower **System 1** to refine its global reasoning using local insights from **System 2**. Conversely, it would enable **System 2** to operate with broader contextual awareness provided by **System 1**. We firmly believe that transforming this pipeline into a bidirectional, cooperative reasoning structure holds immense potential for advancing multi-modal temporal understanding abilities on more flexible and complex tasks.



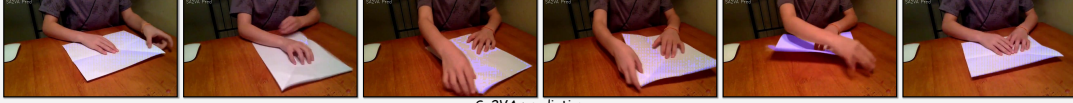
Which is the white object(s) made of cellulose that is/are held in people's hands?



<think> The instruction asks to segment the white object(s) made of cellulose that is/are held in people's hands. The video shows a person folding a white paper on a wooden table. The white object in question is the paper. The person's hands are visible as they manipulate the paper. The action of folding the paper is the key behavior here. The most relevant duration for this action is from 0.5 to 2.0 seconds, as it captures the initial folding of the paper.**</think>**
"answer": {"start_time": "00:00.00", "end_time": "00:00.50", "description": "white paper being folded"}



Omni-R1 prediction



Sa2VA prediction

(a) In this simple example, both our method and Sa2VA are able to select the right target object, but our method is able to segment the target object with a mask consistent with the ground truth, while Sa2VA generates a mask with holes and noise.



Which bottle(s) of beverage will be picked up soon?



<think> The instruction asks to identify which bottle(s) of beverage will be picked up soon. The video shows a group of men playing instruments and drinking beer. The beer bottles are placed on a small table in front of them. The instruction suggests that one of the bottles will be picked up soon. Based on the video, the bottle on the right side of the table is the one that will be picked up soon.**</think>**
"answer": {"start_time": "00:01.50", "end_time": "00:02.00", "description": "the bottle on the right side of the table"}



Omni-R1 prediction



Sa2VA prediction

(b) The target object can only be predicted after one watches the whole video and makes a prediction according to the video context.



Please find the wineglass in which the wine may be finished first?



<think> The instruction asks to segment the wineglass in which the wine may be finished first. The video shows two men sitting at a bar, each with a wineglass and a bottle of wine. The man on the left is gesturing with his hand, while the man on the right is holding his wineglass. The wine in the glass of the man on the left appears to be lower than the wine in the glass of the man on the right. Therefore, the wine in the glass of the man on the left is likely to be finished first.**</think>**
"answer": {"start_time": "00:00.00", "end_time": "00:00.50", "description": "the wineglass in which the wine may be finished first"}



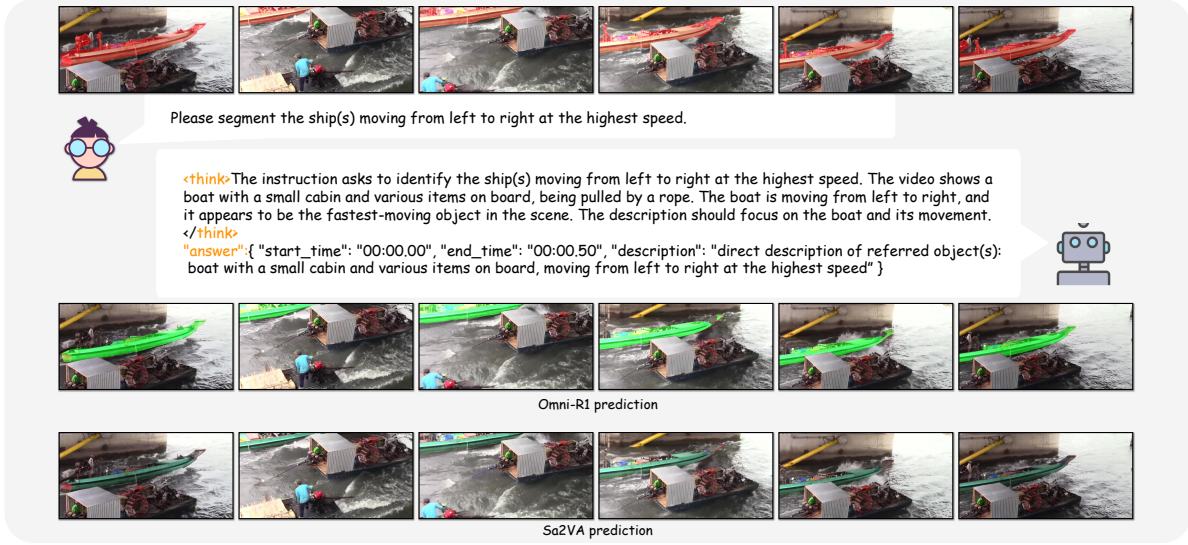
Omni-R1 prediction



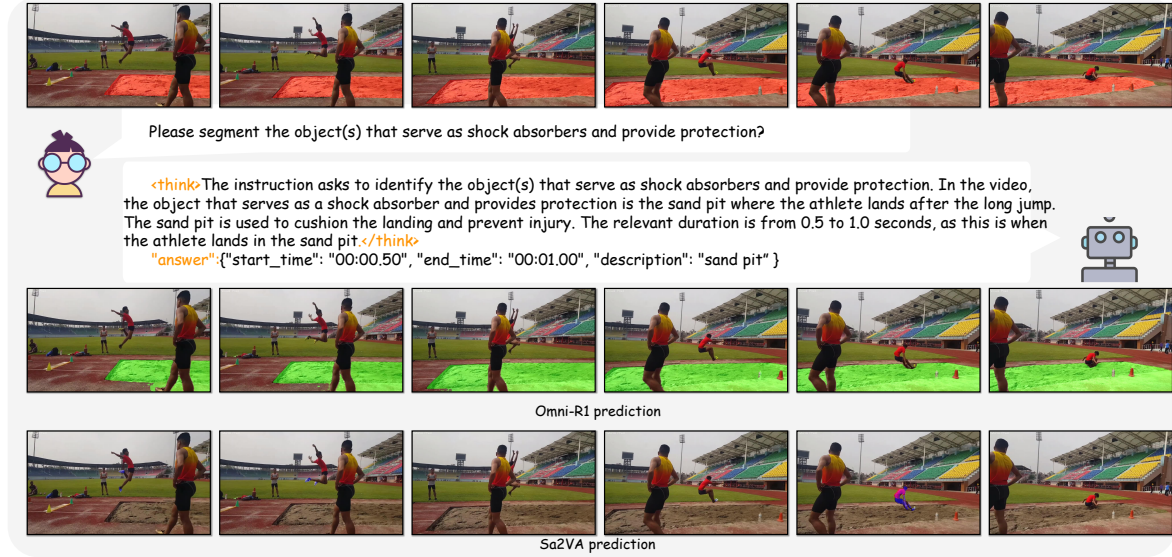
Sa2VA prediction

(c) Though our method is able to select the right wineglass at System 1, it still lets System 2 analyze the detailed information.

Figure 3: Visualization results of our method on three representative VOS cases. Each subfigure illustrates a different reasoning pattern.



(a) The target object can only be identified when one watches the whole video and analyses based on the temporal context to figure out the fast moving object.



(b) The case shows the video understanding abilities of our method. Our method analyses the function of the sand pit and is able to segment, while Sa2VA fails to do so.

Figure 4: More visualization results of our method on representative VOS cases.

216 **References**

- 217 [1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and
218 Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on*
219 *computer vision*, pages 213–229. Springer, 2020.
- 220 [2] David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on*
221 *Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.
- 222 [3] Kim Sung-Bin, Oh Hyun-Bin, JungMok Lee, Arda Senocak, Joon Son Chung, and Tae-Hyun
223 Oh. Avhbench: A cross-modal hallucination benchmark for audio-visual large language models.
224 *arXiv preprint arXiv:2410.18325*, 2024.