AgentCVR: Active Multi-Agent Cross-Video Reasoning via Script-Simulated Reinforcement Learning

Anonymous ACL submission

Abstract

Cross-Video Reasoning (CVR) has emerged as a critical frontier in multimodal intelligence, requiring models to retrieve, align, and aggregate evidence distributed across multiple videos. Current Multimodal Large Language Models (MLLMs) often struggle with CVR, as simple single-pass strategies encode multiple videos into a shared compressed context, potentially obscuring rare but critical evidence. In this paper, we propose **AgentCVR**, a multi-agent framework that treats CVR as an active evidence-acquisition task. AgentCVR employs a Master Agent to iteratively coordinate specialized Visual and Audio Agents for targeted evidence extraction. To ensure efficient training, we introduce Script-Simulated RL, which optimizes the agent's policy with LLM-generated semantic scripts and a lightweight text-based simulator, bypassing costly multimodal inference during online exploration. Experimental results on a comprehensive CVR benchmark show that AgentCVR outperforms single-pass baselines and achieves comparable performance to state-of-the-art closed-source systems, particularly in complex cross-video alignment and localization. To ensure reproducibility, our code is available at `https://anonymous.4open.science/r/AgentCVR-568C`.

1 Introduction

With continuous efforts, Multimodal Large Language Models (MLLMs) (Team et al., 2023, 2026; Bai et al., 2025) have significantly advanced artificial intelligence in complex vision-language tasks. In the domain of video understanding (Nguyen et al., 2024; Tang et al., 2025), these models have demonstrated strong capabilities on tasks such as question answering (Ko et al., 2023; Xiao et al., 2024) and temporal grounding (Hu et al., 2024; Wu et al., 2025). However, most existing video understanding studies and benchmarks are limited to single-video analysis, and thus fail to adequately evaluate a model's ability to reason across multiple videos. As real-world scenarios become more complex, processing isolated videos is no longer enough, driving a shift toward *Cross-Video Reasoning* (CVR) (Zhu et al., 2025; Li et al., 2026).

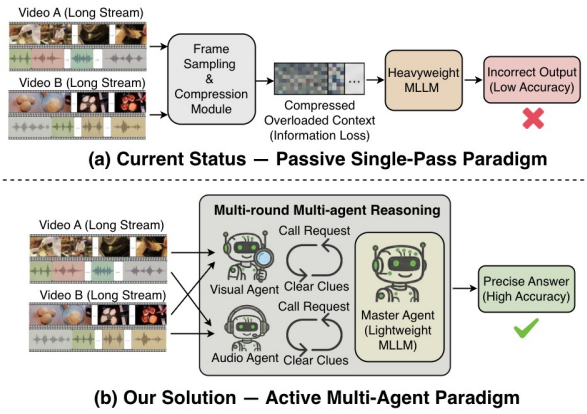


Figure 1: Comparison between two formulations of Cross-Video Reasoning (CVR). (a) Current Status: passive single-pass paradigm. (b) Our Solution: active multi-agent paradigm.

CVR requires models to answer queries whose evidence is distributed across multiple videos, often involving retrieval, alignment, comparison, and aggregation over temporally distant events. This transition introduces an unprecedented challenge: critical evidence is often sparsely distributed within individual videos and temporally misaligned across them, requiring explicit cross-video comparison to derive meaningful insights (Wu et al., 2024; Li et al., 2024; Fu et al., 2025). Such properties make CVR particularly challenging for current MLLMs.

As illustrated in Figure 1(a), a common strategy for CVR is to encode all candidate videos into a shared compressed context and generate the answer in a single pass (Wang et al., 2024c; Ren et al., 2024; Wu et al., 2024). While simple, this design compresses long videos into a fixed-size representation, which can obscure rare but critical cues and leave the model to reason from only partially

grounded observations. Consequently, models often exhibit weak evidence attribution and unreliable cross-video comparison, especially for temporal and comparative queries.

To address these limitations, we argue that CVR should be treated not only as a context-length problem but also as an evidence-acquisition problem. Instead of reasoning over all videos in a single step, a framework should iteratively determine what evidence to inspect, which modality to query, and when sufficient support has been collected. To this end, we propose **AgentCVR**, an active multi-agent framework for CVR. As illustrated in Figure 1(b), a lightweight Master Agent (Liu et al., 2024) iteratively coordinates specialized Visual and Audio Agents (Shen et al., 2023; Surís et al., 2023; Zhang et al., 2025) to extract relevant clues for high-level cross-video deduction. This design keeps multimodal processing focused on targeted evidence and provides the reasoning module with more explicit, query-conditioned inputs.

To further boost AgentCVR's capability to make complex, long-horizon decisions for inter-agent coordination and evidence acquisition, we aim to employ Reinforcement Learning (RL) to optimize its multi-round policy. However, naively applying online RL to CVR is prohibitively expensive, as repeatedly invoking video models incurs massive computational overhead in each step, exacerbated by the scarcity of annotated cross-video trajectories. To address this, we introduce *Script-Simulated RL*, an efficient training paradigm that replaces heavy raw-video interactions during online exploration with LLM-generated semantic scripts and a lightweight text-based simulator that approximates multi-round evidence acquisition. This formulation preserves the core reasoning structure while drastically reducing training costs and the need for human-annotated trajectories. The optimized policy is then directly transferred to the real inference pipeline, where the agent interacts with actual visual and audio agents over raw videos.

We evaluate AgentCVR on the CrossVid benchmark (Li et al., 2026), a large-scale and comprehensive dataset specifically designed for CVR. Experimental results demonstrate that our proposed framework outperforms single-pass baselines and achieves performance comparable to that of state-of-the-art closed-source systems, especially on tasks requiring precise evidence localization and fine-grained cross-video alignment.

Our main contributions are as follows:

- We propose AgentCVR, a multi-agent framework for CVR that fundamentally shifts the paradigm from passive single-pass context compression to active, multi-round evidence acquisition.
- We introduce Script-Simulated RL, a paradigm for training AgentCVR with LLM-generated semantic scripts and a lightweight text-based simulator, avoiding costly multimodal inference during online exploration.
- Extensive experimental results demonstrate that AgentCVR achieves strong performance across comprehensive CVR tasks.

2 Related Work

We review related work along three dimensions in this section: video understanding, multimodal agents, and reinforcement learning.

Video Understanding. The rapid evolution of MLLMs has significantly advanced video understanding (Alayrac et al., 2022; Maaz et al., 2024; Lin et al., 2024; Bhatnagar et al., 2026). To efficiently process long videos, existing approaches typically adopt key-frame selection and temporal token compression (Wang et al., 2024c; Xu et al., 2024; Lin et al., 2024). Meanwhile, benchmarks have progressed from short-clip QA (Lei et al., 2018; Yu et al., 2019; Xiao et al., 2021) to long-context evaluations (Mangalam et al., 2023; Wu et al., 2024; Wang et al., 2025). Beyond single-video settings, real-world applications increasingly require Cross-Video Reasoning, which integrates information across multiple videos (Zhu et al., 2025; Wei et al., 2026). CrossVid (Li et al., 2026) has recently emerged as a representative benchmark. However, single-pass models struggle in CVR, as concatenating videos often obscures sparse and temporally dispersed evidence. Our work focuses on CVR and explores an agentic alternative based on multi-round evidence acquisition.

Multimodal Agents. LLM-based agents enable iterative decision making and tool use by treating the LLM as a controller for multi-step reasoning and interactions (Yao et al., 2023; Shinn et al., 2023; Schick et al., 2023; Zhou et al., 2024). In multimodal contexts, prior work explores modular tool use to facilitate perception and reasoning (Surís et al., 2023; Gupta and Kembhavi, 2023; Sivakumaran et al., 2026) under the paradigm of *active perception* (Aloimonos et al., 1988), where sensory input is dynamically selected based on evolving

hypotheses. While active perception has shown promise in single-video scenarios (Wang et al., 2024b; Shang et al., 2024), its application to CVR remains underexplored. To the best of our knowledge, our work fills this gap by proposing a novel strategy specifically tailored for CVR.

Reinforcement Learning. Reinforcement learning (RL) has significantly advanced LLM reasoning (Schulman et al., 2017; Shao et al., 2024; Guo et al., 2025a). However, optimizing multimodal agents via naive online exploration incurs prohibitive costs due to repeated heavyweight vision-language model inferences. To mitigate this, policies are often trained within surrogate simulators before real-world deployment (Côté et al., 2018; Shridhar et al., 2021; Nakano et al., 2021; Wang et al., 2024a; Tan et al., 2025; Kawaharazuka et al., 2025). Following this principle, we avoid expensive training by learning CVR tool-use policies in a lightweight script-based simulator, while deploying real visual and audio tools only during inference.

3 Preliminary

In this section, we formally define the CVR problem and introduce the Partially Observable Markov Decision Process (POMDP), which serves as the foundation for our proposed framework.

Problem Formulation. We formulate Cross-Video Reasoning (CVR) as a generalized multimodal instruction-following task that requires retrieving, aligning, and aggregating information across multiple independent video streams to produce a final answer. Formally, the system is provided with a user query q and a candidate video set $\mathcal{V} = \{V_1, V_2, \ldots, V_N\}$, where $N \geq 2$. Each video $V_i \in \mathcal{V}$ represents an independent, unaligned spatiotemporal stream comprising heterogeneous multimodal signals (e.g., visual frames and audio tracks). The objective is to learn an intelligent system $\mathcal{G}$ that outputs a response $\hat{a}$ matching the ground-truth correct answer a, denoted as:

$$\hat{a} = \mathcal{G}(q, \mathcal{V}). \tag{1}$$

Unlike single-video tasks with locally confined evidence, CVR involves sparse and temporally unaligned cues across $\mathcal{V}$. Thus, deriving $\hat{a}$ requires explicit inter-video reasoning to align and aggregate fragmented multimodal evidence.

Partially Observable Markov Decision Process. To systematically solve the CVR problem formulated above, we model the active multi-round evidence acquisition process as an MLLM-based POMDP defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R})$. In this formulation, the Master Agent acts as the central controller interacting with a multimodal environment. The components of the POMDP tuple are defined as follows:

- **State ($\mathcal{S}$):** At step t, the state is defined as $s_t = (q, \mathcal{H}_t)$, where q is the user query and $\mathcal{H}_t = (a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ encapsulates the entire interaction history of past actions and observations up to the current step.
- **Action ($\mathcal{A}$):** The agent samples an action $a_t \in \mathcal{A}$ from its parameterized policy $\pi_\theta(a_t|s_t)$, including invoking modality-specific tools (e.g., visual or audio) for evidence retrieval or executing a termination action to generate the final answer.
- **Observation ($\mathcal{O}$):** After executing a_t, the environment returns an observation $o_t \in \mathcal{O}$ (*e.g.,* textual clues from a video segment), and the state is updated accordingly.
- **Transition ($\mathcal{T}$):** The transition is deterministic, updating the state by appending the new pair (a_t, o_t) to the history.
- **Reward ($\mathcal{R}$):** After the trajectory terminates at step T, an episodic reward $R \in \mathcal{R}$ is assigned to evaluate the complete interaction trajectory.

The objective of this MLLM-based POMDP is to optimize the policy π that maximizes the expected cumulative reward:

$$\max_{\theta} \mathbb{E}_{\pi_\theta}[R]. \tag{2}$$

4 Methodology

In this section, we introduce our proposed **AgentCVR** framework, followed by its two main phases: *Script-Simulated RL Training* and *Real-World Inference*, as illustrated in Figure 2.

4.1 AgentCVR

AgentCVR adopts a multi-round evidence acquisition framework for CVR. At its core, a lightweight Master Agent serves as the central controller. At each step t, it reasons over the query and the accumulated interaction history $\mathcal{H}_t$ to dynamically determine which modality-specific agent to invoke, which temporal segment to inspect, and when to terminate the process and generate the final answer.

Formally, we decompose the action space into three subspaces: $\mathcal{A} = \mathcal{A}_{\text{vis}} \cup \mathcal{A}_{\text{aud}} \cup \mathcal{A}_{\text{ans}}$.

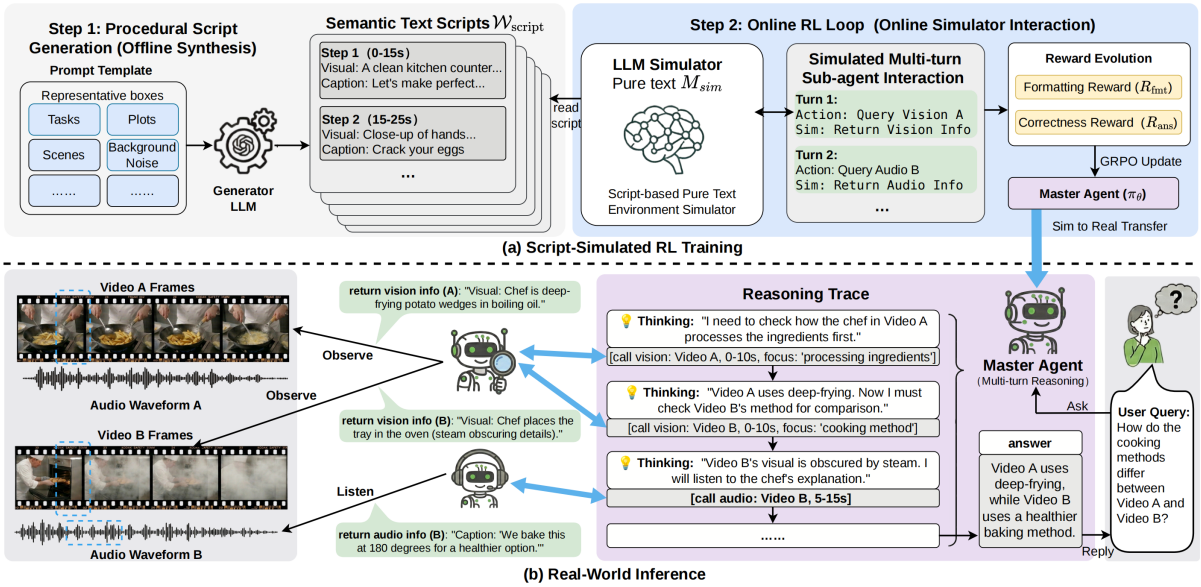


Figure 2: **Overview of AgentCVR.** (a) *Script-Simulated RL Training:* An LLM generator produces semantic scripts ($\mathcal{W}_{\text{script}}$), and a text-based simulator (M_{sim}) provides feedback for policy optimization of the Master Agent (π_θ) with GRPO. (b) *Real-World Inference:* At inference time, the trained Master Agent interacts with visual and audio agents over raw videos to gather localized multimodal evidence for CVR.

- **Visual Query** ($a_{\text{vis}} \in \mathcal{A}_{\text{vis}}$)**:** A visual action is parameterized as $(vid, \tau_{\text{start}}, \tau_{\text{end}}, \mathcal{P}_{\text{focus}})$, where the Master Agent specifies the target video identifier vid, the temporal interval $[\tau_{\text{start}}, \tau_{\text{end}}]$, and an optional focus prompt $\mathcal{P}_{\text{focus}}$. The visual agent then processes this specific segment and returns localized textual observations.
- **Audio Query** ($a_{\text{aud}} \in \mathcal{A}_{\text{aud}}$)**:** Recognizing that auditory signals provide complementary cues for cross-video alignment, an audio action is parameterized as $(vid, \tau_{\text{start}}, \tau_{\text{end}})$, where the Master Agent triggers the audio agent to retrieve speech or sound descriptions from the selected temporal interval $[\tau_{\text{start}}, \tau_{\text{end}}]$.
- **Answer Action** ($a_{\text{ans}} \in \mathcal{A}_{\text{ans}}$)**:** Once the Master Agent determines that sufficient evidence has been collected across the videos, it executes the answer action to terminate the process and generate the final answer $\hat{a}$.

At each round, the Master Agent updates its decision based on the accumulated history and newly returned observations. In this way, AgentCVR separates evidence gathering from final reasoning: modality-specific agents are responsible for retrieving localized observations, while the Master Agent performs cross-video comparison and deduction over the collected evidence. This multi-round design is particularly useful for CVR, where the relevant cues may be sparse, temporally distant, or distributed across different videos.

4.2 Script-Simulated RL Training

To empower AgentCVR to make sophisticated decisions for inter-agent coordination and evidence acquisition, we employ Reinforcement Learning (RL) to optimize the Master Agent. To make RL optimization practical for CVR, we introduce a lightweight script-simulated surrogate environment inspired by text-based environments. The overall pipeline, including offline script construction and online RL training, is summarized in Algorithm 1.

Synthetic Script Construction. We construct a script-simulated surrogate environment by generating synthetic semantic scripts $\mathcal{W}_{\text{script}}$ with a strong LLM conditioned on structured schemas and task templates. Each script defines (i) a set of videos, (ii) temporally grounded events and attributes, and (iii) cross-video relations required to answer queries such as comparisons and temporal constraints. By sampling diverse templates and attribute configurations, we obtain a collection of lightweight textual environments that cover common CVR reasoning patterns. We provide the script schema and representative examples of $\mathcal{W}_{\text{script}}$ in Appendix H.

Online Text-Based Simulator. During online RL, we remove raw-video inference from the training loop and instead employ a lightweight language model as the surrogate simulator M_{sim}. Given a sub-agent invocation action a_t (*e.g.,* querying a video segment) and the corresponding script slice,

the simulator returns an observation:

$$o_t = M_{\text{sim}}(a_t,\ \mathcal{W}_{\text{script}}(vid,\ [\tau_{\text{start}}, \tau_{\text{end}}]))\,. \quad (3)$$

Concretely, M_{sim} performs semantic lookup and composition over scripted events to generate natural-language responses that approximate feedback from visual and audio agents. We further constrain the simulator via system-level instructions to ensure behavioral consistency. In particular, when no relevant event occurs within the queried interval, it returns an explicit "no evidence" response instead of hallucinating content. This design provides a stable and reliable environment feedback for policy optimization while reducing interaction cost, achieving up to $\sim$5$\times$ lower latency in our setup. Moreover, representing videos as structured scripts facilitates scalable environment diversification through LLM-based augmentation without requiring additional video collection.

RL Reward. To align the agent with task objectives and inter-agent interaction constraints, we define a trajectory-level reward consisting of an answer correctness term and an interaction formatting term (Lightman et al., 2024):

- **Correctness Reward:** The primary reward signal is the correctness of the final answer, implemented as a sparse reward:

$$R_{\text{ans}} = \begin{cases} 1, & \text{if } \hat{a} = a, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

- **Formatting Reward:** Since our framework relies on structured sub-agent interactions, invalid invocation actions may disrupt execution. We therefore add a lightweight auxiliary reward to encourage valid trajectories. Specifically, we set $R_{\text{fmt}} = 0.1$ if the invocation satisfy predefined format constraints, and $R_{\text{fmt}} = 0$ otherwise.

The total reward is defined as:

$$R_{\text{total}} = R_{\text{ans}} + R_{\text{fmt}}. \quad (5)$$

Policy Update. To optimize the Master Agent, we employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024), an efficient RL algorithm. For each query q, we sample a group of G trajectories $\{\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(G)}\}$ from the current policy π_{old} and optimize the following objective to obtain the updated policy π_θ:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\mathcal{H}^{(i)} \sim \pi_{\theta_{\text{old}}}} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \min \Big(r_i(\theta) A_i, \text{clip}\left(r_i(\theta), 1-\epsilon, 1+\epsilon\right) A_i \Big) \Bigg] - \beta \mathbb{D}_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}), \quad (6)$$

where $r_i(\theta) = \frac{\pi_\theta(\mathcal{H}^{(i)})}{\pi_{\theta_{\text{old}}}(\mathcal{H}^{(i)})}$ is the trajectory-level importance sampling ratio computed over the full sequence of actions and outputs in $\mathcal{H}^{(i)}$, ϵ and β are hyperparameters, and $\mathbb{D}_{\text{KL}}$ is a KL term that regularizes the updated policy toward a reference policy. The advantage A_i is derived by normalizing the reward relative to the generated group:

$$\hat{A}_i = \frac{R^{(i)}_{\text{total}} - \mu_R}{\sigma_R}, \quad (7)$$

where $R^{(i)}_{\text{total}}$ denotes the reward of trajectory $\mathcal{H}^{(i)}$, μ_R is the group mean reward, and σ_R is the standard deviation of the rewards within the group.

4.3 Real-World Inference

A primary advantage of our Script-Simulated RL Training is its seamless Sim-to-Real transferability. At inference time, the optimized policy π_θ is deployed directly to interact with the physical environment.

As outlined in Algorithm 2, for each user query q and a set of raw, uncompressed candidate videos $\mathcal{V}$, the Master Agent initiates an iterative reasoning loop. At each step t, the agent samples an action $a_t \sim \pi_\theta(\cdot \mid s_t)$ based on its current state. Rather than relying on textual simulation, the agent executes a_t by invoking actual perception models such as a visual agent T_{vis} or an audio agent T_{aud} on the raw videos $\mathcal{V}$ to obtain the real multimodal observation o_t. The interaction history is subsequently updated with the newly acquired (a_t, o_t) pair, and the state is advanced accordingly. This evidence-gathering process continues until the Master Agent determines that sufficient cross-video support has been collected, at which point it terminates the loop and outputs the final prediction $\hat{a}$.

5 Experiments

5.1 Experimental Setup

Dataset. We evaluate the performance of our AgentCVR on CrossVid (Li et al., 2026), a comprehensive large-scale benchmark designed for CVR. Unlike conventional video QA datasets that focus on a single video per query, CrossVid adopts a setting where each query is associated with multiple videos. CrossVid is organized along four dimensions: Comparative Analysis, Temporal Understanding, Multi-view Reasoning, and Free-form QA. It covers 10 tasks in total, with detailed descriptions provided in Appendix A.

Table 1: Performance comparison between baselines and our AgentCVR on the CrossVid benchmark. The best results are highlighted in **bold**, and the second-best results are underlined across open-source single-pass models, adapted single-video agents, and our proposed framework. The table reports accuracy (%) across 10 tasks, along with dimension-level averages, including **C.Avg**, **T.Avg**, and **M.Avg**. `O.Avg` denotes the overall average.

Task (→)	Comparative Analysis					Temporal Understanding				Multi-view Reasoning			Free-form QA	`O.Avg`
Method (↓)	BU	NC	CC	PEA	**C.Avg**	PI	FSA	PSS	**T.Avg**	MSR	MOC	**M.Avg**	CCQA	
Human	85.60	92.30	90.70	83.90	88.13	91.60	85.20	89.90	88.90	93.20	94.20	93.70	85.20	89.18
Closed-source Frontier Models														
Doubao-1.5-VL-pro	51.20	58.10	69.50	36.40	53.80	66.90	4.60	36.80	36.10	37.40	32.00	34.70	50.10	44.30
GPT-4.1	46.20	34.60	58.50	51.20	47.63	70.90	8.60	60.50	46.67	38.60	38.20	38.40	44.60	45.19
Gemini-2.5-Pro	54.20	51.80	68.70	36.40	52.78	76.50	13.40	78.20	56.03	32.00	25.30	28.65	59.80	49.63
Open-source Single-pass Models														
Qwen3-VL-4B	17.93	21.47	26.62	28.61	23.66	49.89	3.13	2.97	18.66	24.94	28.71	26.83	11.59	21.59
Qwen3-VL-8B	23.48	27.35	40.72	34.94	31.60	62.54	5.71	7.22	25.16	34.51	27.15	30.83	31.18	29.47
Qwen2.5-VL-32B	31.40	30.50	48.60	39.70	38.30	65.70	5.20	8.70	26.53	23.70	**39.60**	31.65	41.20	33.78
Qwen3-VL-32B	**46.75**	37.50	59.54	39.77	45.89	**73.00**	4.70	7.50	28.40	26.62	36.56	31.59	**47.51**	38.37
Adapted Single-Video Agents														
VideoAgent-8B	24.31	31.42	42.16	31.73	32.41	67.20	5.93	17.10	30.08	32.70	28.50	30.60	30.65	30.01
VCA-8B	27.50	30.80	39.20	32.80	32.53	64.60	8.30	21.60	31.50	34.40	26.90	30.65	33.60	30.71
Our Proposed Framework														
AgentCVR-4B	26.41	38.71	51.66	32.61	35.35	62.31	12.31	19.42	31.35	28.32	31.43	29.88	26.43	32.06
AgentCVR-8B	33.88	**43.65**	**70.71**	**41.23**	**47.35**	67.43	**16.63**	**30.41**	**38.68**	**37.18**	33.20	**35.19**	41.85	**42.03**

Baselines. We compare our proposed AgentCVR with the following baseline methods:

- *Closed-source models:* We include Doubao-1.5-VL-pro (Guo et al., 2025b), GPT-4.1 (Achiam et al., 2023), and Gemini-2.5-Pro (Team et al., 2023) as representative proprietary systems.
- *Open-source single-pass models:* We evaluate Qwen2.5-VL-32B-Instruct and the Qwen3-VL series models (4B, 8B, and 32B in thinking mode) in a single-pass setting, where frames are uniformly sampled from all videos and fed to the model along with the prompt.
- *Adapted single-video agents:* To assess the necessity of cross-video-specific design, we reproduce two recent single-video agents, VideoAgent (Wang et al., 2024b) and VCA (Yang et al., 2025b), and adapt them to the CVR setting by concatenating the input videos with explicit video-level timestamp boundaries in the prompt. For fair comparison, both agents use the same 8B-scale backbone model as AgentCVR.

Implementation details. We use Qwen3-VL-4B and Qwen3-VL-8B models (Bai et al., 2025) in thinking mode as the Master Agent for AgentCVR. The visual agent matches the Master Agent scale, while the audio agent is based on Whisper (Radford et al., 2023). For Script-Simulated RL, we use an additional Qwen3-4B (Yang et al., 2025a) model as the environment simulator, and GRPO training uses a maximum turn limit $T_{\max} = 20$ (Guo et al., 2025a). Detailed hyperparameter configurations and implementation settings are described in Appendix B, and the prompts employed in AgentCVR are presented in Appendix H.

5.2 Main Results

We first present an overall comparison of all methods. The main results on the CrossVid benchmark are shown in Table 1, from which we draw the following observations:

- **Existing models face a strict trade-off between scale and practicality.** Open-source single-pass models show limited performance at smaller scales, only becoming competitive when scaled up to 32B-sized. Conversely, although closed-source frontier models achieve stronger results, their high inference cost and closed-source nature limit their applicability, especially in privacy-sensitive or local deployment scenarios.
- **Current single-video agent paradigms provide limited benefits for CVR.** Although adapted single-video agents yield slight improvements on specific tasks, their overall performance gains remain marginal. This limitation underscores the necessity for a more advanced, tailored framework specifically designed to optimize performance in complex CVR scenarios.

Table 2: Ablation studies of (i) architectural and training configurations, and (ii) multimodal tool synergy.

AgentCVR Components		`O.Avg` (%)	
Multi-Agent	Script-Simulated RL	4B	8B
✗	✗	21.59	29.47
✓	✗	30.18	37.20
✓	✓	**32.06**	**42.03**
Vision Tool	Audio Tool	8B	
✗	✓	14.56†	
✓	✗	36.07	
✓	✓	**42.03**	

† Note that the overall average for the Audio-only configuration is computed exclusively over the subset of tasks that contain audio tracks.

- **Our AgentCVR framework establishes a new standard for CVR.** Both AgentCVR-4B and AgentCVR-8B effectively unlock the potential of smaller-scale models, achieving substantial gains over their corresponding base models. Notably, AgentCVR-8B achieves optimal results across multiple tasks when evaluated against open-source single-pass models and adapted single-video agents, while delivering performance highly comparable to that of powerful closed-source frontier models.

5.3 Ablation Studies

To better understand the contribution of different components in AgentCVR, we conduct extensive ablation studies from two perspectives: (1) architectural and training configurations, and (2) multimodal agent synergy.

Architectural and training configurations. We first evaluate the individual contributions of our Multi-Agent architecture and the Script-Simulated RL paradigm by progressively adding them to the base model. As shown in the upper part of Table 2, replacing the naive single-pass baseline with our multi-agent architecture yields a clear improvement at both the 4B and 8B scales. This suggests that separating localized evidence gathering from high-level reasoning effectively overcomes the context-overload limitations of single-pass models. Notably, introducing Script-Simulated RL further improves performance, validating the effectiveness of our training paradigm and highlighting the importance of explicit policy optimization for mastering long-horizon sub-agent interaction decisions and reliable termination behaviors beyond the capability of zero-shot prompting alone.

Table 3: Comparison of decision alignment and average per-call latency between simulated (Sim) and real (Real) environments.

Task Category	Alignment (%)	Sim Lat.	Real Lat.
Reasoning (BU)	83.9†	**1.6s**	7.3s
Localization (FSA)	87.4*	**1.7s**	8.6s

*Measured by Sim-to-Real IoU. †Decision Overlap Rate.

Multimodal agent synergy. Next, we investigate the individual contributions and the synergistic effect of the visual and audio agents in AgentCVR. As shown in the lower part of Table 2, relying exclusively on the audio agent leads to poor overall performance. While audio provides useful contextual cues, the results highlight the inherent difficulty of CVR tasks in the absence of fundamental visual grounding. Conversely, using the visual agent solely achieves relatively strong performance, indicating that visual perception is the key factor in extracting cross-video evidence.

To provide a more fine-grained analysis, we provide detailed task-level ablation breakdowns in Appendix E.

5.4 In-depth Analysis

In this section, we conduct additional experiments to further study the design and effectiveness of our AgentCVR framework.

5.4.1 Analysis of the Simulated Environment

A key requirement of *Script-Simulated RL* is that the simulator M_{sim} provides feedback consistent with the real execution environment. We evaluate Sim-to-Real alignment on two representative tasks: Functional Step Alignment (FSA), which involves predicting temporal intervals, and Behavioral Understanding (BU), formulated as multiple-choice reasoning. To assess simulator fidelity, we construct an evaluation subset by transcribing the visual and audio content into semantic scripts $\mathcal{W}_{\text{trans}}$. We then run AgentCVR in both (i) the script-based simulator and (ii) the real execution environment, and measure consistency between these two settings, rather than accuracy against ground truth.

Cross-environment consistency. As shown in Table 3, the simulator exhibits strong alignment with the real environment. For FSA, we measure IoU between predicted temporal intervals, showing strong agreement across the two settings and indicating that $\mathcal{W}_{\text{trans}}$ preserves key temporal structure. For BU, we report the Decision Overlap Rate, *i.e.,* the fraction of identical answer choices selected in both

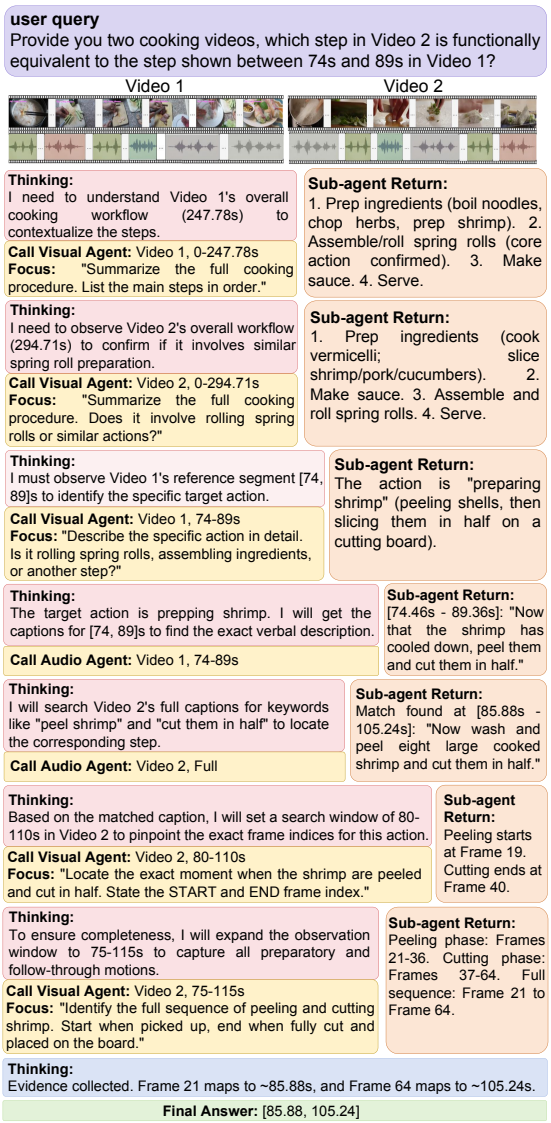


Figure 3: A case study of AgentCVR multi-turn reasoning trace on an FSA task.

environments, which remains high. Overall, the simulator provides consistent feedback for both continuous and discrete decision settings.

Efficiency. We further measure the average per-call latency as reported in Table 3. For FSA, the simulator reduces latency from 8.6s to 1.7s per call, achieving an approximately 5× speedup. For BU, latency is reduced from 7.3s per turn in the real environment to 1.6s in simulation. These substantial reductions significantly improve the efficiency of online RL exploration, while preserving strong alignment with real-world feedback.

5.4.2 Qualitative Analysis

Figure 3 shows a representative reasoning trace of AgentCVR on a Functional Step Alignment (FSA) instance, where the goal is to localize in Video 2 a segment functionally equivalent to the reference interval in Video 1 (74s–89s). More qualitative cases for AgentCVR are provided in Appendix G.

The Master Agent first queries the visual agent for high-level procedure summaries of both videos and inspects the reference interval in Video 1 to identify the key action. It then leverages complementary audio cues, querying the audio agent for corresponding narration or captions to corroborate the visual observation and narrow the search space in Video 2. Guided by this cue, the Master Agent focuses subsequent queries on a candidate window in Video 2 (85.88s–105.24s), avoiding exhaustive scanning of the full video. Finally, it refines the temporal boundaries through denser visual queries within the candidate window and aligns the complete action sequence. The resulting prediction (85.88s–105.24s) achieves an IoU of 0.93 with the ground-truth interval (86.0s–104.0s). This case demonstrates how iterative multi-agent interaction in AgentCVR supports localized evidence retrieval and precise temporal grounding for CVR.

Failure modes. AgentCVR can still fail in long interaction traces due to instruction or format drift, and may prematurely commit to a single label in ambiguous multi-label scenarios when the available evidence is insufficient. We provide a taxonomy of failure modes along with representative cases and detailed analyses in Appendix G.2.

6 Conclusion

In this work, we propose AgentCVR, a novel framework that fundamentally shifts the paradigm of Cross-Video Reasoning from passive, single-pass context compression to dynamic, multi-round evidence acquisition. By utilizing a lightweight Master Agent to iteratively coordinate specialized visual and audio tools, our approach ensures that cross-video analysis is explicitly grounded in targeted, query-relevant clues rather than an overloaded context window. To overcome the prohibitive cost and data scarcity of online multimodal exploration, we introduce Script-Simulated RL, which replaces raw-video interactions with LLM-generated semantic scripts and a lightweight text simulator for scalable long-horizon policy learning. Extensive experiments on the challenging CrossVid benchmark demonstrate the superiority of our approach. Ultimately, AgentCVR establishes a new, efficient, and highly interpretable path for multimodal agents to actively reason across complex, real-world video streams.

Limitations

Although AgentCVR generally outperforms open-source single-pass models and adaptive single-video agents, a marginal performance gap remains when compared to state-of-the-art closed-source frontier models, highlighting the need for further optimization. Our failure case analysis also shows that multi-turn reasoning may suffer from instruction drift, premature cognitive closure, and over-reliance on internal parametric priors. In addition, the current framework only optimizes the Master Agent, leaving modality-specific agents frozen without task-specific adaptation. While the proposed Script-Simulated RL paradigm is effective, slight misalignments between the simulated and real-world environments persist, necessitating higher-fidelity simulators to ensure absolute environmental reliability. Finally, the active multi-round interaction paradigm inevitably introduces higher inference latency than single-pass methods, indicating that future work must focus on optimizing decision efficiency and minimizing interaction turns to facilitate seamless real-world deployment.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*.

John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. 1988. Active vision. *International journal of computer vision*, 1(4):333–356.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shubhang Bhatnagar, Renxiong Wang, Kapil Krishnakumar, Adel Ahmadyan, Zhaojiang Lin, Lambert Mathias, Xin Luna Dong, Babak Damavandi, Narendra Ahuja, and Seungwhan Moon. 2026. VideoMind: Thinking in steps for long video understanding. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 406–416.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, and 1 others. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, and 1 others. 2025b. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14953–14962.

Zi-Yuan Hu, Yiwu Zhong, Shijia Huang, Michael Lyu, and Liwei Wang. 2024. Enhancing temporal modeling of video LLMs via time gating. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2845–2856.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. 2025. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*.

Dohwan Ko, Ji Lee, Woo-Young Kang, Byungseok Roh, and Hyunwoo Kim. 2023. Large language models are temporal and causal reasoners for video question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4300–4316.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. TVQA: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379.

Jingyao Li, Jingyun Wang, Molin Tan, Haochen Wang, Cilin Yan, Likun Shi, Jiayin Cai, Xiaolong Jiang, and Yao Hu. 2026. Crossvid: A comprehensive benchmark for evaluating cross-video reasoning in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 6244–6252.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. Agentbench: Evaluating llms as agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Thong Nguyen, Yi Bin, Junbin Xiao, Leigang Qu, Yicong Li, Jay Zhangjie Wu, Cong-Duy Nguyen, See-Kiong Ng, and Anh Tuan Luu. 2024. Video-language understanding: A survey from model architecture, model training, and data perspectives. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3636–3657.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023*, Proceedings of Machine Learning Research, pages 28492–28518. PMLR.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Chuyi Shang, Amos You, Sanjay Subramanian, Trevor Darrell, and Roei Herzig. 2024. TraveLER: A modular multi-LMM agent framework for video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9740–9766.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving AI tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025*, pages 1279–1297.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. In *9th International Conference on Learning Representations, ICLR 2021*.

Nithin Sivakumaran, Justin Chen, David Wan, Yue Zhang, Jaehong Yoon, Elias Stengel-Eskin, and Mohit Bansal. 2026. DART: Leveraging multi-agent disagreement for tool recruitment in multimodal reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5445–5464.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11888–11898.

Weiting Tan, Xinghua Qu, Ming Tu, Meng Ge, Andy T Liu, Philipp Koehn, and Lu Lu. 2025. Process-supervised reinforcement learning for interactive multimodal tool-use agents. *arXiv preprint arXiv:2509.14480*.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, and 1 others. 2025. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, and 1 others. 2026. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024a. Voyager: An open-ended embodied agent with large language models. *Trans. Mach. Learn. Res.*, 2024.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, and 1 others. 2025. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024b. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, and 1 others. 2024c. Internvideo2: Scaling foundation models for multimodal video understanding. In *European conference on computer vision*, pages 396–416.

Zhixiang Wei, Yi Li, Zhehan Kan, Xinghua Jiang, Zuwei Long, Shifeng Liu, Hongze Shen, Wei Liu, Xiaoyu Tan, Haojia Lin, and 1 others. 2026. Youtu-vl: Unleashing visual potential via unified vision-language supervision. *arXiv preprint arXiv:2601.19798*.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024,*.

Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. 2025. Number it: Temporal grounding videos like flipping manga. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13754–13765.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. 2024. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13204–13214.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. 2025b. Vca: Video curious agent for long video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20168–20179.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023*.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9127–9134.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024*.

Nannan Zhu, Yonghao Dong, Teng Wang, Xueqian Li, Shengjun Deng, Yijia Wang, Zheng Hong, Tiantian Geng, Guo Niu, Hanyan Huang, and 1 others. 2025. Cvbench: Benchmarking cross-video synergies for complex multimodal reasoning. *arXiv preprint arXiv:2508.19542*.

Algorithm 1 AgentCVR: Training Pipeline

Require: Policy π_θ, simulator M_{sim}

Offline script construction

$\mathcal{W}_{\text{script}} \leftarrow$ Generate semantic scripts for cross-video scenarios

Online RL training

for each training episode **do**
  while the agent has not produced a final answer $\hat{a}$ **do**
    Sample action $a_t \sim \pi_\theta(\cdot \mid s_t)$
    $o_t \leftarrow M_{\text{sim}}(a_t, \mathcal{W}_{\text{script}})$
    Update the interaction history with (a_t, o_t)
  end while
  Compute the trajectory reward and update π_θ with GRPO
end for

Algorithm 2 AgentCVR: Inference Pipeline

Require: Policy π_{θ^*}, visual tool T_{vis}, audio tool T_{aud}

Inference on raw videos

for each query q and input videos $\mathcal{V}$ **do**
  while the agent has not produced a final answer $\hat{a}$ **do**
    Sample action $a_t \sim \pi_{\theta^*}(\cdot \mid s_t)$
    $o_t \leftarrow$ Execute a_t using T_{vis} or T_{aud} on $\mathcal{V}$
    Update the interaction history with (a_t, o_t)
  end while
end for

A Dataset Details

In this section, we provide a detailed introduction to CrossVid (Li et al., 2026), a comprehensive large-scale benchmark specifically designed for Cross-Video Reasoning (CVR) settings. Curated from diverse publicly available datasets, CrossVid consists of 5,331 videos and 9,015 high-quality QA pairs. These queries cover diverse formats, including single-choice, multiple-choice, and open-ended generation, requiring models to process and reason over an average of 770 seconds of video content per query. It encompasses 10 distinct tasks grouped into four high-level dimensions. We introduce each of these dimensions in the following sections.

A.1 Comparative Analysis

This dimension evaluates the ability of intelligent systems to extract task-relevant information from multiple videos and perform cross-video comparisons. It consists of the following four tasks:

- **Behavioral Understanding (BU):** Given a set of videos depicting either wildlife behaviors or everyday human activities, models are required to recognize specific actions, understand their aims and purposes, or accurately identify whether each video contains the queried action.
- **Narrative Comprehension (NC):** Given four film clips sharing the same genre, models are required to analyze and contrast the plot, characters, environment, and underlying themes across the clips.
- **Culinary Comparison (CC):** Given a group of videos showing the preparation of the same dishes, models must compare ingredient processing methods, utensil usage, procedural sequences, and flavor profiles across the videos.
- **Procedural Error Analysis (PEA):** Given videos accompanied by descriptions of possible errors, models are required to identify specific errors mentioned in the query and trace the reasons for these mistakes.

A.2 Temporal Understanding

This dimension assesses the capability of intelligent systems to perform temporal localization and chronological reasoning across multiple video streams. It contains three tasks:

- **Plot Inference (PI):** Given the beginning and ending segments of a film, the model is asked to infer the missing plot in the middle part.
- **Functional Step Alignment (FSA):** Given two different videos, models are asked to locate a specific temporal segment in one video that corresponds to a specified time interval in the other, requiring the alignment of corresponding steps based on semantic and functional equivalence.
- **Procedural Step Sequencing (PSS):** A single video is segmented at the step level, and the clips are randomly shuffled. Models must reconstruct the correct temporal sequence, which evaluates their causal reasoning and temporal inference capabilities.

A.3 Multi-view Reasoning

This dimension provides intelligent systems with two temporally synchronized road videos, each captured from a different aerial drone perspective. It consists of two tasks:

- **Multi-view Spatial Reasoning (MSR):** Models are queried about spatial relationships, such as the relative distances and positions of specific objects at a given moment across different views.
- **Multi-view Object Counting (MOC):** Models are required to count specific objects at a certain moment or over a defined time interval, which requires the integration of multi-perspective information for precise counting.

A.4 Free-form QA

This dimension evaluates the ability of intelligent systems to perform comparative analysis and answer open-ended questions comprehensively:

- **Comparative Culinary QA (CCQA):** Two videos featuring the same items are provided. Models are required to compare them and generate a detailed textual response identifying the differences in procedures, assessing their capability to compare fine-grained details without predefined options.

B Implementation Details

For Script-Simulated RL training, we use the open-source `verl` (Sheng et al., 2025) framework with a GRPO-based (Shao et al., 2024) optimization objective. For each query, we sample $G = 8$ complete trajectories and constrain the multi-turn interaction with a maximum turn limit $T_{\max} = 20$ and a tolerance zone $T_{\text{tol}} = 10$ to prevent infinite looping. The optimizer used in RL is AdamW (Loshchilov and Hutter, 2019). The detailed hyperparameters are summarized in Table 4. During RL rollout and inference, we use `vLLM` (Kwon et al., 2023) engine to accelerate generation. We configure the model with a maximum length of 8192 tokens and set the temperature to 0.0. All experiments are conducted on a single compute node equipped with 8 NVIDIA H800 (80GB) GPUs.

Table 4: Detailed configurations used for training AgentCVR via GRPO.

Hyperparameter	**AgentCVR-4B**	**AgentCVR-8B**
Base Model	`Qwen3-VL-4B`	`Qwen3-VL-8B`
Learning Rate	1e-7	5e-7
KL Coefficient (ϵ)	0.005	0.005
Samples per Prompt (G)	8	8
Training Batch Size	32	32
PPO Mini-Batch Size	16	16
Micro Batch Size per GPU	1	1
Max Prompt Length	4096	4096
Max Response Length	4096	4096
Epochs	2	3

C Sub-Agents Details

C.1 Sub-Agent Configurations

During the real-world zero-shot inference phase, AgentCVR strictly interacts with physical environments using the following specialized agents:

- **Visual Agent:** We employ the `Qwen3-VL-8B` model in thinking mode as our dedicated visual perception module. When the Master Agent dispatches a visual observation request, frames from the targeted video segment are uniformly sampled. To balance visual fidelity and computational efficiency, the extracted frames are resized so that the longer side is 360 pixels, while maintaining the original aspect ratio.
- **Audio Agent:** We utilize the `Whisper large-v3` model to execute selective auditory extraction. To ensure robust background noise filtering and precise transcription, the model uses a beam size of 5, a decoding temperature of 0.0, a log-probability threshold of -1.0, disables conditioning on previous text, and applies a no-speech threshold of 0.6.

C.2 Agent Fidelity Cases

A fundamental prerequisite for our Script-Simulated RL is that the surrogate text simulator

($\mathcal{S}_{text}$) must faithfully replicate the semantic feedback of the real physical agents ($\mathcal{T}_{real}$). This ensures that the reasoning meta-policy learned by the Master Agent during the offline training phase can seamlessly transfer to the online real-world inference phase without semantic drift.

To intuitively demonstrate this high Sim-to-Real fidelity, we present a comparative case study on a Functional Step Alignment (FSA) task. The Master Agent is tasked with finding a step in Video 2 that is functionally equivalent to the reference segment (55s–65s) in Video 1.

As shown in the comparative dialogue boxes below, the Master Agent exhibits a remarkably consistent strategic meta-policy across both the simulated and real environments. In both settings, the agent autonomously develops a sophisticated cross-modal verification strategy: It first dispatches the visual agent to understand the physical action of the reference segment (*pressing with a cloth to extract liquid*). It then invokes the audio agent to extract specific verbal anchors (*"extract the water"*). Next, it efficiently searches the target video's audio track to locate functional synonyms (*"press... dry them off"*). Finally, it conducts a visual verification to establish precise temporal boundaries.

Simulated Environment Training Trace (Using $\mathcal{S}_{text}$)

Task: Functional Step Alignment (FSA)
Query: *Provide you two cooking videos, which step in Video 2 is functionally equivalent to the step shown between 55s and 65s in Video 1?*
Turn 1 & 2 (Visual Exploration)
Thinking: I need to observe Video 1's reference segment [55, 65]s to confirm the exact action. What is the person doing with hands/tools?
Action: [call vision: Video 1, 55-65s, focus: "Describe the exact action..."]
Observation: Simulated Tool Return: The chef uses a **white cloth to press** the ingredient. The action appears to extract any remaining juice.
Turn 3 (Audio Grounding)
Thinking: The reference segment describes extracting liquid. I need to get the captions to confirm the specific verbal description.
Action: [call get_caption: Video 1, 55-65s]
Observation: Simulated Tool Return: [55.54s - 57.54s]: "Now we **extract the water**." [59.00s - 66.60s]: "A potato pulp, nice and dry..."
Turn 4 (Cross-Video Audio Search)
Thinking: Video 1 mentions "extract the water". I will search Video 2's full captions for similar functional keywords like "squeeze", "extract juice", or "dry".
Action: [call get_caption: Video 2, Full]
Observation: Simulated Tool Return: ...[43.96s - 51.54s]: "get a nice clean tea towel." [55.56s - 60.76s]: "Fold the tea towel over and just give it a little wiggle and a **press. Just to dry them** off."...
Turn 5 (Final Visual Verification)
Thinking: The caption [55.56s-60.76s] matches the functional purpose. I need to verify if this visual action is removing moisture.
Action: [call vision: Video 2, 55.56-60.76s, focus: "What is the purpose of pressing the tea towel?"]
Observation: Simulated Tool Return: The purpose of pressing the tea towel on the potatoes is to **remove excess moisture**.
Turn 6 (Final)
Thinking: Evidence collected. The functional equivalence is perfectly aligned.
Action: [Answer]
Final Answer: [55.56, 60.76]

Real-World Inference Trace (Using Physical $\mathcal{T}_{real}$)

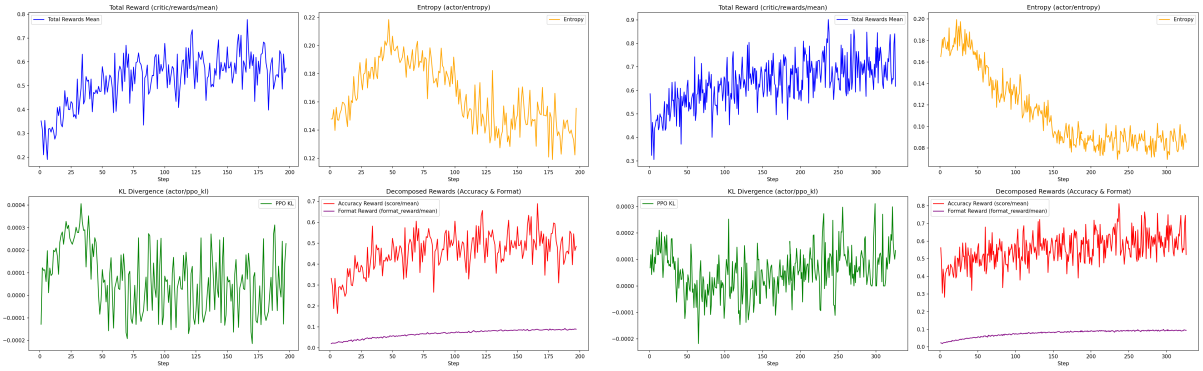
Task: Functional Step Alignment (FSA)
Query: *Provide you two cooking videos, which step in Video 2 is functionally equivalent to the step shown between 55s and 65s in Video 1?*
Turn 1 & 2 (Visual Exploration)
Thinking: I need to precisely observe Video 1's reference segment [55, 65]s to determine the specific visual features of the action.
Action: [call vision: Video 1, 55-65s, focus: "Describe the specific action... What is the tool used?"]
Observation: Real Vision Tool Return: Tool used: A **white cloth**. Action: The chef is **pressing/squeezing** the vegetable to extract juice into a metal bowl.
Turn 3 (Audio Grounding)
Thinking: The action is 'pressing vegetable to extract juice'. I will get the captions nearby to find text keywords like 'squeezing' or 'pressing'.
Action: [call get_caption: Video 1, 55-65s]
Observation: Real Audio Tool Return: [55.54s - 57.54s]: "Now we **extract the water**." [59.00s - 66.60s]: "A potato pulp, nice and dry..."
Turn 4 (Cross-Video Audio Search)
Thinking: Video 1's caption mentions 'extract the water'. I will retrieve Video 2's captions to find corresponding keywords for extracting liquid.
Action: [call get_caption: Video 2, Full]
Observation: Real Audio Tool Return: ...[43.96s - 51.54s]: "get a nice clean tea towel." [55.56s - 60.76s]: "Fold the tea towel over and just give it a little wiggle and a **press. Just to dry them** off."...
Turn 5 (Final Visual Verification)
Thinking: The caption at 55.56s-60.76s explicitly mentions pressing to dry. I need to observe this exact window (expanded slightly to 53-63s) to locate the precise frame boundaries of the pressing action.
Action: [call vision: Video 2, 53-63s, focus: "Identify the full pressing action of the tea towel... Provide START and END frame indices."]
Observation: Real Vision Tool Return: The full pressing action spans from **START frame 27** to **END frame 34**.
Turn 6 (Final)
Thinking: Evidence collected. The observation window is 53s - 63s (10s duration). Start frame 27 translates to 53.00s, and End frame 34 translates to 63.00s.
Action: [Answer]
Final Answer: [53.00, 63.00]

This compelling alignment proves that the mock observations generated by $\mathcal{S}_{text}$ during RL training are robust enough to cultivate advanced, generalized multimodal agent interaction behaviors. Consequently, the agent can seamlessly replace the text simulator with heavy physical agents during the zero-shot inference phase without requiring any real-video fine-tuning, further highlighting the effectiveness of our Script-Simulated RL paradigm in AgentCVR.

D Training Dynamics

As illustrated in Figure 4, both AgentCVR-4B and AgentCVR-8B demonstrate highly stable and effective policy optimization during the Script-Simulated RL phase.

First, regarding reward convergence and task decomposition, the *Total Reward* for both models shows a consistent upward trend. By examining the *Decomposed Rewards*, we observe a distinct two-stage learning process. The formatting reward (R_{fmt}) converges rapidly in the early steps, indicating that the agent quickly learns to produce valid API calls and to strictly follow the reasoning-trace



(a) AgentCVR-4B (b) AgentCVR-8B

Figure 4: The RL training dynamics for (a) AgentCVR-4B and (b) AgentCVR-8B during the GRPO training phase, illustrating the convergence of Total Reward, Entropy, KL Divergence, and Decomposed Rewards.

Table 5: Experimental results of detailed task-level ablation studies. The best results are highlighted in **bold**, the second-best results are underlined, and **`O.Avg`** denotes the overall average.

Task (→)	Comparative Analysis				Temporal Understanding			Multi-view Reasoning		Free-form QA	**`O.Avg`**
Method (↓)	BU	NC	CC	PEA	PI	FSA	PSS	MSR	MOC	CCQA	
Open-source Single-pass Models											
Qwen3-VL-4B	17.93	21.47	26.62	28.61	49.89	3.13	2.97	24.94	28.71	11.59	21.59
Qwen3-VL-8B	23.48	27.35	40.72	34.94	62.54	5.71	7.22	34.51	27.15	31.18	29.47
Multi-Agent Framework Only											
AgentCVR-4B (Zero-Shot)	23.70	28.57	50.95	32.17	59.84	9.11	16.30	28.42	27.63	29.60	30.18
AgentCVR-8B (Zero-Shot)	24.29	42.92	67.66	40.71	62.94	**16.84**	21.35	33.71	29.37	33.80	37.20
Multimodal Tool Synergy											
Audio-only	14.70	26.30	23.72	-	32.20	0.70	0.40	-	-	-	14.56†
Visual-only	27.26	33.07	53.58	**41.23**	62.95	6.21	28.42	32.53	**37.18**	35.20	36.07
Our Proposed Framework											
AgentCVR-4B	26.41	38.71	51.66	32.61	62.31	12.31	19.42	28.32	31.43	26.43	32.06
AgentCVR-8B	**33.88**	**43.65**	**70.71**	**41.23**	**67.43**	16.63	**30.41**	**37.18**	33.20	**41.85**	**42.03**

† Note that the overall average for the Audio-only configuration is computed exclusively over the subset of tasks that contain audio tracks.

constraints. Once the format is mastered, the correctness reward (R_{ans}) becomes the primary driver of optimization, steadily climbing as the agent learns optimal active perception strategies to navigate the cross-video environment and identify correct answers.

Furthermore, the *Entropy* curves reflect the exploration-exploitation trade-off and the emergence of policy confidence. Notably, AgentCVR-8B exhibits a highly pronounced and smooth entropy decay, dropping from approximately 0.18 to 0.08. This signifies that the larger model effectively transitions from broad exploration to a highly confident, deterministic agent interaction policy. While AgentCVR-4B also shows a general downward trend, it exhibits higher variance, reflecting the typical capacity limits of smaller models when establishing rigid cognitive closure criteria.

Finally, we evaluate training stability and capability retention through the *KL Divergence*. Notably, the *KL Divergence* for both variants remains consistently bounded within a very small range, on the order of 10^{-4}, indicating stable optimization without significant capability drift. This result suggests that our method improves multi-agent reasoning and interaction capabilities while largely preserving the base model's inherent linguistic and cognitive abilities.

E Detailed Ablation Studies

In this section, we provide a comprehensive per-task breakdown of the ablation results in Table 5. The results show that the full AgentCVR configuration equipped with both visual and audio agents achieves the most balanced performance across all four high-level dimensions.

F Turns & Frames

In this section, we detail the maximum allowable number of frames and average interaction turns for

Table 6: Maximum allowable number of frames and average interaction turns for AgentCVR and the compared methods.

Method	#Frame	#Turn
Closed-source Frontier Models		
Doubao-1.5-VL-pro	256	1
GPT-4.1	48	1
Gemini-2.5-Pro	128	1
Open-source Single-pass Models		
Qwen3-VL-4B	256	1
Qwen3-VL-8B	256	1
Qwen2.5-VL-32B	256	1
Qwen3-VL-32B	256	1
Adapted Single-Video Agents		
VideoAgent-8B	37.3	11.6
VCA-8B	70.5	16.6
Our Proposed Framework		
AgentCVR-4B	306.8	14.4
AgentCVR-8B	355.2	16.7

AgentCVR and the compared methods as shown in Table 6.

Compared with single-pass models that process all inputs in a single forward pass, agent-based methods perform multi-turn interactions to progressively retrieve and reason over relevant video content. We observe that AgentCVR does not reach the pre-defined maximum limit of 20 interaction turns in practice. Instead, its average number of turns remains comparable to other agent-based methods, indicating that the framework does not incur additional interaction overhead. Meanwhile, AgentCVR achieves substantially higher frame coverage than existing baselines, demonstrating that it can efficiently utilize each interaction step to access a broader visual context without increasing the overall interaction cost. This shows that AgentCVR maintains strong efficiency while improving effective visual exploration for CVR tasks.

G Case Studies

G.1 Successful Cases

In this section, we provide a detailed qualitative analysis of AgentCVR's inference process to illustrate its robust multi-agent reasoning capabilities. To demonstrate the versatility of our framework, we select representative success cases across all four evaluation dimensions of the CrossVid benchmark: Comparative Analysis, Temporal Understanding, Multi-view Reasoning, and Free-form QA. The following interaction transcripts detail the multi-turn cognitive processes, highlighting how the Master Agent dynamically formulates active perception strategies, precisely invokes specialized visual and audio agents to extract fine-grained multimodal cues, and systematically synthesizes cross-video evidence to deduce the final answers.

G.1.1 Dimension: Comparative Analysis

Task: Behavioral Understanding (BU)

Query: *Which behavior in the following videos is most likely to help the animal avoid predators?*
Options: A. Video one. B. Video two. C. Video three. D. Video four.
Ground Truth: D
Turn 1
Thinking: Task initialization. I need to perform a comprehensive overview of Video 1 to identify the main character, setting, and primary action to determine if it relates to the query.
Action: [call vision: Video 1, 0-3.33s, focus: "Describe the main character and their primary action in detail. What is the setting? What objects are present?"]
Observation: Visual Agent Return: The main character is a single bird moving through dry grass. The setting is an open, natural field. It does not exhibit defensive behaviors, showing only natural movement.
Turn 2
Thinking: Continuing the scan. Now I will analyze Video 2 to check for any defensive behaviors, such as freezing, camouflaging, or rapid escape.
Action: [call vision: Video 2, 0-3.41s, focus: "Identify if the subject exhibits defensive behaviors like freezing, camouflaging, or rapid escape. Are there any signs of predator presence?"]
Observation: Visual Agent Return: The bird is flying normally. There is no freezing, camouflaging, or rapid escape, nor are there any signs of predator presence.
Turn 3
Thinking: Continuing the scan. I will analyze Video 3 for defensive behaviors or signs of predators.
Action: [call vision: Video 3, 0-8.0s, focus: "Identify if the subject exhibits defensive behaviors like freezing, camouflaging, or rapid escape. Are there any signs of predator presence?"]
Observation: Visual Agent Return: The birds exhibit calm, natural movements like walking and preening. No defensive behaviors or signs of predator presence are observed.
Turn 4
Thinking: Finally, I will analyze Video 4 to confirm if there are defensive behaviors or predator threats.
Action: [call vision: Video 4, 0-3.20s, focus: "Identify if the subject exhibits defensive behaviors like freezing, camouflaging, or rapid escape. Are there any signs of predator presence?"]
Observation: Visual Agent Return: The animal remains stationary and clings to the tree trunk. Its coloration and texture blend with the bark, indicating camouflage to avoid detection. This clearly exhibits defensive behavior.
Turn 5 (Final)
Thinking: After a comprehensive analysis of all four videos: Video 1 shows a bird moving in dry grass; Video 2 shows a bird flying normally; Video 3 shows birds moving calmly; Video 4 shows an animal actively avoiding detection through freezing and camouflage. The conclusion is option D.
Action: [Answer]
Final Answer: D

Task: Narrative Comprehension (NC)

Query: *Which two videos feature home invasions as key plot points?*
Options: A. Video1 and Video2 B. Video2 and Video3 C. Video3 and Video4 D. Video1 and Video4
Ground Truth: C
Turn 1
Thinking: Task initiation. As a movie analysis expert, I first need to get the subtitles for Video 1 to understand the plot context. Captions will help me quickly locate keywords like 'home invasion', 'break in', or 'forced entry'.
Action: [call get_caption: Video 1, Full]
Observation: Audio Agent Return: [Subtitles retrieved. Mentions witness protection, but no direct home invasion keywords are found.]
Turn 2
Thinking: Continuing the process, I will fetch the full captions for Video 2 to check for similar keywords that might imply a home invasion plot.
Action: [call get_caption: Video 2, Full]
Observation: Audio Agent Return: [Subtitles retrieved. Mentions "Eastern Road, main house" and "search and rescue force".]

Turn 3
Thinking: Next, I will get the full captions for Video 3 to check for related keywords.
Action: [call get_caption: Video 3, Full]
Observation: Audio Agent Return: [Subtitles retrieved. Mentions "search all informants who recently entered witness protection".]
Turn 4
Thinking: Finally, I will get the captions for Video 4 to locate key dialogue segments.
Action: [call get_caption: Video 4, Full]
Observation: Audio Agent Return: [Subtitles retrieved. Mentions "Search everywhere! We need them alive!".]
Turn 5
Thinking: Subtitle analysis suggests Videos 1 and 2 might involve entering a house based on certain phrases. I need to use the Visual Agent to confirm if the specific timestamps corresponding to these phrases show actual home invasion scenes.
Action: [call vision: Video 1 & Video 2, specific timestamps, focus: "Check for house checking or forced entry. Note any signs of family members being invaded."]
Observation: Visual Agent Return: Video 1 shows a calm setting with no signs of forced entry or search actions. Video 2 shows an outdoor street combat scene, not a house interior. Neither depicts a home invasion.
Turn 6
Thinking: Now I need to visually check the suspicious timestamps in Video 3 and Video 4 to see if they depict home invasions or forced entries.
Action: [call vision: Video 3 & Video 4, specific timestamps, focus: "Check if Video 3 involves searching a witness's home. Check if Video 4 shows family members being searched or invaded. Look for violent or forced entry."]
Observation: Visual Agent Return: Video 3 shows a man on a phone in a normal home setting with no forced entry. Video 4 shows armed tactical personnel forcibly entering a residence with a scared woman and child present, indicating a clear home invasion.
Turn 7 (Final)
Thinking: After multiple rounds of analysis: Video 1 and Video 2 have no home invasion scenes. Video 4 clearly shows an armed home invasion targeting family members. Video 3 involves investigating/searching for a witness, which aligns with the plot point context. Through elimination and matching the available pairs in the options, the correct choice is C (Video 3 and Video 4).
Action: [Answer]
Final Answer: C

Task: Culinary Comparison (CC)

Query: *What distinguishes the moisture removal method in Video 3 from the others? Options: A. Uses a metal press, B. Relies on sunlight drying, C. Employs a sieve, D. Skips moisture removal*
Ground Truth: C
Turn 1
Thinking: Task started; as an initialization step, I must first get the captions for Video 1, which is a critical step for building a global index.
Action: [call get_caption: Video 1, Full]
Observation: Audio Agent Return: [94.00s - 100.84s] "...take all the starchy moisture out of this as you can...".
Turn 2
Thinking: Continuing the initialization process, I now need to get the captions for Video 2 to ensure all text information is gathered.
Action: [call get_caption: Video 2, Full]
Observation: Audio Agent Return: [72.82s - 77.80s] "Okay now we're going to try to squeeze some of the moisture out of these. You can press it through a colander...".
Turn 3
Thinking: Continuing the initialization workflow, I will retrieve the captions for Video 3.
Action: [call get_caption: Video 3, Full]
Observation: Audio Agent Return: [114.82s - 120.68s] "we're gonna strain this right so let's get all of this right into our strainer...".
Turn 4
Thinking: Completing the initialization by getting the captions for Video 4.
Action: [call get_caption: Video 4, Full]
Observation: Audio Agent Return: [101.66s - 109.58s] "wrap up the towel tightly and squeeze out all the liquids...".
Turn 5
Thinking: Based on the captions, Video 3 mentions using a strainer, while Video 2 mentions a colander. I need to visually compare the moisture removal methods across all four videos to verify these descriptions.
Action: [call vision: Videos 1, 2, 3, 4, focus: "Compare moisture removal methods: Check if Video 3 uses a sieve (strainer)..."]
Observation: Visual Agent Return: Video 3 uses a sieve (Correct). Video 2 uses manual squeezing, not a colander (Incorrect). Video 1 uses hands without a tea cloth (Incorrect). Video 4 uses a dishcloth (Correct).
Turn 6
Thinking: Video 3 definitely uses a sieve. I will double-check if Video 4 skips moisture removal or if Video 1 matches any other drying methods.
Action: [call vision: Videos 1, 4, focus: "Check if Video 1 uses a tea cloth and if Video 4 skips moisture removal."]
Observation: Visual Agent Return: Video 1 does not use a tea cloth. Video 4 does not skip moisture removal.
Turn 7 (Final)
Thinking: Evidence collected over the turns confirms that Video 3's method of using a sieve is distinctly different from the others, aligning with Option C.
Action: [Answer]
Final Answer: C

Task: Procedural Error Analysis (PEA)

Query: *Which action is incorrectly performed in exactly two videos?*
Options: A. detach the base and the chassis B. attach the sound module and the chassis C. detach the bumper and the cabin D. attach the rear roof and the transport cabin
Ground Truth: A
Turn 1
Thinking: The question asks to identify an action that is incorrectly performed in exactly two videos. Incorrect actions usually involve detaching rather than assembling (attaching). I need to observe the key segments of all videos to identify the incorrect actions and count their occurrences.
Action: [call vision: Video 1 (0-218s), Video 2 (0-259s), Video 3 (0-170s), focus: "Describe the incorrect action in each video (e.g., detachment/abnormal operation), paying special attention to whether it involves the Detach or Attach actions mentioned in the options, and the names of the parts involved."]
Observation: Visual Agent Return: Video 1: Detach wheel. The person incorrectly removes the wheel from the vehicle body. Video 2: Detach chassis. The person splits the vehicle body into two parts (the main body and the base/chassis), which is an incorrect detachment. Video 3: Attach cargo box. The person incorrectly installs the cargo box onto the vehicle body.
Turn 2 (Final)
Thinking: Based on the observations, both Video 1 and Video 2 contain a detachment action (Detach), while Video 3 involves an incorrect attachment. Option A is "detach the base and the chassis". The Visual Agent explicitly noted that the chassis was detached in Video 2. Video 1 and Video 2 are the only ones sharing the incorrect "Detach" action type. Based on the available options and the explicit identification of the chassis, Option A is the logical match for an incorrect detachment action occurring across the two videos.
Action: [Answer]
Final Answer: A

G.1.2 Dimension: Temporal Understanding

Task: Plot Inference (PI)

Query: *Given the Beginning and Ending clips of a movie scene, infer what most likely happened in the middle.*
Options: A. The banker conspired with robbers to stage a fake heist. B. Security systems malfunctioned during routine maintenance. C. Robbers violently breached the vault and initiated a car chase. D. The banker discovered embezzlement and confronted executives. E. Protesters stormed the bank during a financial crisis. F. An earthquake damaged vault security infrastructure.
Ground Truth: C
Turn 1
Thinking: Task initiation. I need to first get the captions for Video 1 (Beginning) to understand the premise of the story.
Action: [call get_caption: Video 1, Full]
Observation: Audio Agent Return: The dialogue reveals a father talking to his son, threatening to kill "Dominic Toretto" because he "burn what is mine" regarding their "cash houses". This establishes a motive related to stolen/destroyed money and a clear conflict.
Turn 2
Thinking: Now I need to fetch the captions for Video 2 (Ending) to understand the narrative outcome.
Action: [call get_caption: Video 2, Full]
Observation: Audio Agent Return: The captions contain repeated screams of "Oh, my God!" followed by dialogue about driving: "Find the line, feel the car, and let it fly." This suggests a high-stakes car-related sequence.
Turn 3
Thinking: The ending audio strongly suggests a car chase. I must visually verify the first 10 seconds of Video 2 to check for car wreckage, damage, or active pursuit.
Action: [call vision: Video 2, 0-10s, focus: "Check for car wreckage,

damage, or chase scenes. Note ground marks, smoke, or debris."]
Observation: Visual Agent Return: The scene confirms a violent vehicle pursuit. There are dark SUVs, a large vehicle crashing into objects with smoke and debris, and a "POLICIA CIVIL" police car rolling over with severe body deformation.
Turn 4
Thinking: The ending is a police car chase. I need to briefly check if Video 1 shows any robbers or vault visuals during the threat about burning property.
Action: [call vision: Video 1, 90-105s, focus: "Check for visual clues of burned property, robbers, or cash houses."]
Observation: Visual Agent Return: The scene is purely an indoor dialogue. No direct visual evidence of robbers, fire, or cash houses is shown yet.
Turn 5 (Final)
Thinking: Based on the collected evidence: Video 1 establishes a violent conflict involving stolen/burned cash houses. Video 2 depicts a destructive car chase involving dark SUVs (often used by robbers) and flipping police cars. The logical missing link between "threatening revenge over stolen cash" and an "intense police car chase" aligns perfectly with a violent heist and getaway. Options A, B, D, E, and F lack any visual or audio support (no protesters, no earthquake, no banker conspiracies). Option C is the only logical bridge connecting the plot.
Action: [Answer]
Final Answer: C

Task: Functional Step Alignment (FSA)

Query: *Provide you two cooking videos, which step in Video 2 is functionally equivalent to the step shown between 74s and 89s in Video 1?*
Ground Truth: [86.0, 104.0]
Turn 1
Thinking: I need to first understand the overall cooking workflow of Video 1 (duration: 247.78s) to contextualize the steps.
Action: [call vision: Video 1, 0-247.78s, focus: "Summarize the full cooking procedure. List the main steps in order."]
Observation: Visual Agent Return: To summarize the full cooking procedure for Spring Rolls: 1. Gather Ingredients. 2. Cook Vermicelli Noodles. 3. Prepare Shrimp (peeling/slicing). 4. Prepare Char Siu Pork... 9. Assemble Spring Rolls. 10. Serve.
Turn 2
Thinking: Now I need to observe Video 2's overall workflow (294.71s) to confirm if it involves similar spring roll preparation and rolling actions.
Action: [call vision: Video 2, 0-294.71s, focus: "Summarize the full cooking procedure. Does it involve rolling spring rolls or similar actions?"]
Observation: Visual Agent Return: Step 1: Ingredient Preparation (boil noodles, chop herbs, prepare shrimp). Step 2: Assembling and Rolling Spring Rolls... Key Action: Yes, rolling spring rolls is a central action.
Turn 3
Thinking: I must observe Video 1's reference segment [74, 89]s to identify the specific target action. Based on the workflow, this might be the assembly or rolling step.
Action: [call vision: Video 1, 74-89s, focus: "Describe the specific action in detail. Is it rolling spring rolls, assembling ingredients, or another step?"]
Observation: Visual Agent Return: The specific action is preparing shrimp through two key steps: 1. Peeling shrimp. 2. Slicing peeled shrimp. This sequence is part of ingredient preparation, not rolling.
Turn 4
Thinking: The reference segment in Video 1 is about prepping shrimp. I will get the captions for this timeframe to find the exact verbal description.
Action: [call get_caption: Video 1, 74-89s]
Observation: Audio Agent Return: [74.46s - 89.36s]: "Now that the shrimp has cooled down, peel them and cut them in half."
Turn 5
Thinking: The target action is "peel shrimp" and "cut them in half". I will search Video 2's full captions for these keywords to locate the corresponding step.
Action: [call get_caption: Video 2, Full]
Observation: Audio Agent Return: ...[67.76s - 85.88s]: "of fresh cilantro, and two clean lettuce leaves." [85.88s - 105.24s]: "Now wash and peel eight large cooked shrimp and cut them in half." [105.24s - 109.60s]...
Turn 6
Thinking: The caption precisely matches the interval [85.88s - 105.24s] in Video 2. I will set a visual search window of 80-110s to pinpoint the exact start and end frame indices for this action.
Action: [call vision: Video 2, 80-110s, focus: "Locate the exact moment when the shrimp are peeled and cut in half. State the START frame index and END frame index."]
Observation: Visual Agent Return: START frame index: 19. END frame index: 40.
Turn 7
Thinking: To ensure completeness, I will expand the observation window slightly to 75-115s (using 64 frames) to capture all preparatory and follow-through motions.
Action: [call vision: Video 2, 75-115s, focus: "Identify the full sequence of peeling and cutting shrimp. Start when picked up, end when fully cut and placed on the board."]
Observation: Visual Agent Return: Peeling phase: Frames 21-36. Cutting phase: Frames 37-64. The full sequence spans frames 21 to 64.
Turn 8 (Final)
Thinking: Evidence collected. The observation window is 75s - 115s (40s span). Frame 21 corresponds to $\sim$85.88s, and Frame 64 corresponds to $\sim$105.24s. The alignment is fully confirmed by both audio and Visual Agents.
Action: [Answer]
Final Answer: [85.88, 105.24]

Task: Procedural Step Sequencing (PSS)

Query: *Review these video clips. They are shuffled segments from a single cooking video. Please determine their correct chronological order.*
Ground Truth: 3->5->4->2->1
Turn 1
Thinking: Task started. I need to first get the captions for Video 1 to get a preliminary understanding of the action.
Action: [call get_caption: Video 1]
Observation: Audio Agent Return: Mentions a little bit of feta cheese and that feta has a lot of salt in it.
Turn 2
Thinking: Continuing to get captions. Now I need to get the captions for Video 2 to understand its action.
Action: [call get_caption: Video 2]
Observation: Audio Agent Return: Mentions going to use udon noodles, adding them to the pan, and letting the spinach wilt just a little bit.
Turn 3
Thinking: Continuing to get captions. Now I need to get the captions for Video 3 to understand its action.
Action: [call get_caption: Video 3]
Observation: Audio Agent Return: Mentions putting noodles into some hot water.
Turn 4
Thinking: Continuing to get captions. Now I need to get the captions for Video 4 to understand its action.
Action: [call get_caption: Video 4]
Observation: Audio Agent Return: Mentions adding fresh spinach.
Turn 5
Thinking: Continuing to get captions. Now I need to get the captions for Video 5 to understand its action.
Action: [call get_caption: Video 5]
Observation: Audio Agent Return: Mentions adding onions, tomatoes, eggplant, and kalamata olives.
Turn 6
Thinking: Now I need to observe the overall visual of Video 3 to confirm its specific action. The caption mentions putting noodles into hot water, which might correspond to the boiling step. I need to confirm if it is the boiling process.
Action: [call vision: Video 3, focus: "Describe the entire clip. Is the video showing noodles being boiled in water? Are the noodles raw and stiff, or are they already cooked and soft?"]
Observation: Visual Agent Return: The video shows a person in a kitchen preparing noodles. The sequence begins with pouring raw and stiff noodles into a pot of boiling water. This is a standard step for cooking noodles.
Turn 7
Thinking: I need to observe the overall visual of Video 5 to confirm its action. The caption mentions adding onions, tomatoes, eggplant, etc., which might correspond to the sautéing step. I need to confirm the raw or cooked state of the ingredients.
Action: [call vision: Video 5, focus: "Describe the entire clip. Are the onions, tomatoes, eggplant, and olives raw or cooked? Is the pan empty or filled with ingredients?"]
Observation: Visual Agent Return: The video depicts a cooking process where the pan contains flames, indicating active cooking. Ingredients are being added to the pan. The ingredients are raw at the time of the clip. Cooked ingredients would show signs of browning or softening, which are not visible here. The pan is filled with ingredients, evidenced by flames and the chef stirring.
Turn 8
Thinking: I need to observe the overall visual of Video 2 to confirm its action. The caption mentions sautéing udon noodles and letting spinach wilt. I need to confirm if spinach is added during this sauteing process.
Action: [call vision: Video 2, focus: "Describe the entire clip. Are the noodles already cooked, or are they being sautéed? Is the spinach added to the pan during this process?"]
Observation: Visual Agent Return: The chef pours noodles into a pan containing spinach. The chef is stirring the noodles and spinach together. The noodles are being sautéed. Spinach is visible in the pan before the

noodles are added. Thus, spinach was added to the pan before the noodles during this process.
Turn 9
Thinking: I need to observe the overall visual of Video 1 to confirm its action. The caption mentions cheese, parsley, and noodles, which might involve the plating or final seasoning step. I need to confirm if the ingredients are in their final state.
Action: [call vision: Video 1, focus: "Describe the entire clip. Is the dish already plated and ready to eat? Are the ingredients cooked and final?"]
Observation: Visual Agent Return: The entire clip shows the person cooking at a stove. There is no plating visible in any frame. Spinach is green and appears to be wilting but not fully cooked. Noodles are visible but show no signs of being fully soft or finished. A liquid is poured over the ingredients and stirred, indicating the dish is still in the cooking process. There is no sign of the dish being "done". The pan remains on the stove with visible heat. The dish is not plated. Ingredients are in the middle of cooking.
Turn 10 (Final)
Thinking: Based on the analysis: Video 3 shows boiling raw noodles in boiling water (preparation stage). Video 5 shows heating oil and adding raw onions, tomatoes, and other ingredients (start of sautéing). Video 4 shows adding raw spinach (a step during the sautéing process). Video 2 shows sautéing the noodles and spinach (mid-to-late cooking stage). Video 1 shows adding cheese and parsley for seasoning, but it is not plated yet (final step). Therefore, the final chronological sequence is 3->5->4->2->1.
Action: [Answer]
Final Answer: 3->5->4->2->1

G.1.3 Dimension: Multi-view Reasoning

Task: Multi-view Spatial Reasoning (MSR)

Query: *When A3 completely leaves view A, where is B4 located in view B?*
Options: A. bottom right of the frame , B. top right of the frame , C. top left of the frame , D. not shown in the frame
Ground Truth: C
Turn 1
Thinking: I need to determine where B4 is located in View B when A3 completely leaves View A. First, I must find the frame range where A3 appears in View A to determine the frame it leaves.
Action: [call vision: Video 1, Frames 1-500, focus: "Locate A3 in View A (Video 1). Determine the frame range where it appears and identify when it completely leaves the frame."]
Observation: Visual Agent Return: A3 appears in frames 1–10. It completely leaves the frame in frame 11.
Turn 2
Thinking: Now I need to determine the position of B4 in View B when A3 completely leaves View A (Frame 11). I need to check if B4 appears in View B around Frame 11 and determine its specific location relative to road markings and other objects.
Action: [call vision: Video 2, Frames 10-12, focus: "Locate B4 in View B (Video 2). Check if it appears in Frame 11 and determine its position relative to road markings and other objects."]
Observation: Visual Agent Return: B4 appears in Frame 11. It is located in the left lane of the highway, bounded by white lane dividers and a forward-pointing directional arrow. Relative to other objects, it is positioned to the right of a red truck (same lane) and to the left of a black car (adjacent lane), with trees and a green roadside area to its immediate left.
Turn 3 (Final)
Thinking: Based on the observations, A3 appears in frames 1-10 of View A and completely leaves View A at frame 11. In frame 11 of View B, B4 is located in the leftmost lane of the highway, with a black car to its right and a red truck to its left. Given the UAV perspective, the left lane corresponds to the left side of the frame. Based on the options, B4 is located at the top left of the frame in View B. This matches option C.
Action: [Answer]
Final Answer: C

Task:Multi-view Object Counting (MOC)

Query: *When B1 fully appears in view B, how many green trucks are there in view A?*
Ground Truth: B
Turn 1
Thinking: I need to confirm the frame range where B1 fully appears in View B, and synchronously check the number of green trucks in View A. First, I will locate the appearance frame range of B1 in View B.
Action: [call vision: Video 2, Frames 0-219, focus: "Locate B1 in View B (Video 2). Identify the frames where it fully appears and note its attributes."]
Observation: Visual Agent Return: B1 (the archway banner) fully appears in all 16 sampled frames (0-15).
Turn 2
Thinking: Now I need to check if there are green trucks in frames 0-15 of View A (Video 1). Since B1 fully appears in frames 0-15 of View B, I must synchronously check the same frame range in View A, focusing on the number and color attributes of the trucks.
Action: [call vision: Video 1, Frames 0-15, focus: "Inspect View A (Video 1) frames 0-15 for green trucks. Count the number of green trucks visible in the BBox area and confirm their color attributes."]
Observation: Visual Agent Return: In all 16 frames (0-15), a single green truck (with a green cab) is consistently visible in the upper-middle section of the road. The green truck's cab is clearly green in all frames, satisfying the color requirement. There is 1 green truck visible in the BBox area.
Turn 3 (Final)
Thinking: Based on the observations, B1 fully appears in frames 0-15 of View B, which corresponds to the same frame range in View A. In View A's frames 0-15, there is exactly 1 green truck present with clear color attributes, and no other green trucks or occlusions. The final answer is B.
Action: [Answer]
Final Answer: B

G.1.4 Dimension: Free-form QA

Task: Comparative Culinary QA (CCQA)

Query: *How do the main ingredients used in the batter mixture differ between Video A and Video B?*
Ground Truth: Base liquid difference (club soda vs. beer), Flour composition (corn flour inclusion vs. plain flour only), Spice selection (chili/oregano vs. paprika/garlic/sugar).
Turn 1
Thinking: First, I need to understand the verbal descriptions in both videos to identify potential ingredient differences. Let me retrieve the full captions for both videos.
Action: [call get_caption: Video 1]
Observation: Audio Agent Return: Mentions All-Purpose Flour, Corn Flour, Red Chilli Powder, dry Oregano, Salt, Garlic, and Club Soda.
Turn 2
Thinking: Now I need to retrieve the full captions for Video 2 to compare ingredient lists. This will help identify potential differences in batter components.
Action: [call get_caption: Video 2]
Observation: Audio Agent Return: Mentions plain or all-purpose flour, beer, paprika, sugar, garlic powder, salt, and black pepper.
Turn 3
Thinking: From captions, Video A uses All-Purpose Flour, Corn Flour, Red Chilli Powder, Oregano, Salt, Garlic, and Club Soda. Video B uses All-Purpose Flour, Beer, Paprika, Sugar, Garlic Powder, Salt, and Black Pepper. Now I need to verify if these ingredients are actually shown in the videos through visual observation.
Action: [call vision: Videos 1 & 2, focus: "Verify ingredient visuals: Check if corn flour, red chilli powder... are actually shown in the videos."]
Observation: Visual Agent Return: For Video A, corn flour, red chilli powder, oregano, and club soda are shown; garlic is not. For Video B, beer, paprika, sugar, garlic powder, and black pepper are all present.
Turn 4
Thinking: Based on Turn 3's visual verification, Video A has corn flour, red chilli powder, oregano, club soda but no garlic. Video B has beer, paprika, sugar, garlic powder, black pepper.Now I need to check if there are other ingredient differences in the batter preparation process.
Action: [call vision: Videos 1 & 2, focus: "Check for additional ingredient differences: Are there any other dry ingredients or liquids added..."]
Observation: Visual Agent Return: Video 1 includes only one dry ingredient (flour) and no liquids added to the batter during the observed segment. Video 2 includes additional dry ingredients (spices, salt) and additional liquids (vinegar, oil, beer) beyond the initial flour.
Turn 5 (Final)
Thinking: After 4 rounds of observation, I've confirmed key ingredient differences. Video A uses corn flour, red chilli powder, oregano, and club soda. [Video B uses beer, paprika, sugar, garlic powder, and black pepper. Video B also includes additional liquids (vinegar, oil) and more spice varieties.
Action: [Answer]
Final Answer: 1. Ingredients: Video A uses All-Purpose Flour, Corn Flour, Red Chilli Powder, Oregano, Salt, Club Soda (no garlic). Video B uses All-Purpose Flour, Beer, Paprika, Sugar, Garlic Powder, Black Pepper. 2. Key Differences: Video A uses corn flour and club soda. Video B uses beer, paprika, sugar, and black pepper, plus additional liquids. 3. Flavor Profile: Video A emphasizes heat and herbiness, while Video B focuses on beer aroma, sweetness, and complexity. 4. Preparation: Video A adds spices directly to dry flour mix; Video B incorporates beer as a liquid base with multiple spice additions.

G.2 Failure Cases

While AgentCVR demonstrates robust performance across various cross-video tasks, analyzing its failure cases provides critical insights into the current boundaries of LLM-based multi-agent systems. In this section, we categorize and examine typical failure modes encountered during real-world inference. These failures generally stem from three distinct challenges: (1) *instruction drift* leading to formatting errors during extended context reasoning, (2) *premature cognitive closure* or exclusivity bias in multi-label reasoning tasks, and (3) the *over-reliance on internal parametric priors* (common sense) that overrides specific visual evidence. By investigating these limitations, we highlight critical areas for future improvements in autonomous video understanding.

Formatting Errors

Query: *How do the tools used for breading the meat differ between video A and video B?*
Ground Truth: Meat handling tools (hands vs fork), Pounding equipment (none vs mallet+pot), Breading transfer method (direct press vs fork transfer).
Turn 1
Thinking: First I need to understand the overall cooking process and identify when the breading step occurs.
Action: Answer[call get_caption: Video 1]
Observation: Audio Agent Return: Mentions preparing wiener schnitzel and the steps involving flour, egg mixture, and breadcrumbs.
Turn 2
Thinking: Now I need to understand Video 2's content to locate the breading step. The caption will show if they use different tools like egg wash, breadcrumbs, or unique equipment for coating.
Action: Answer[call get_caption: Video 2]
Observation: Audio Agent Return: Mentions a breading station with flour, beaten egg, and breadcrumbs, explicitly advising to use a fork to avoid breading your fingers.
Turn 3
Thinking: From the captions, both videos use flour, egg wash, and breadcrumbs for breading. I'll first observe Video A's breading process to identify tools used.
Action: Answer[call vision: Video 1, focus: "Describe the tools used for breading in Video A. Specifically identify: 1) Type of bowl... 2) Any tools... 3) Method..."]
Observation: Visual Agent Return: Large metal mixing bowls are used. The person uses their hands to mix ingredients and applies breadcrumbs by pressing them onto the food. No whisk or fork is visible.
Turn 4
Thinking: Now I need to analyze Video B's breading process. The caption mentions using a fork to avoid 'bread your fingers,' suggesting a different tool than Video A's hand-only method. I'll focus on the breading tools in Video B.
Action: Answer[call vision: Video 2, focus: "Describe Video B's breading tools..."]
Observation: Visual Agent Return: Rectangular metal trays are used. A fork is used to handle the meat during breading, and hands are used to press the meat into the breadcrumbs.
Turn 5 (Format Error & Auto-Recovery)
Thinking: Both videos use pressing for breadcrumbs, but Video A's caption mentions 'three eggs' while Video B specifies 'two eggs' - this might affect the egg wash consistency. I need to check if they use different tools for mixing the egg wash (e.g., fork vs whisk).
Action: Answer[Malformed JSON Output]
System Intervention: *!Format issue on attempt 1. Retrying cleanly... (LLM response is not valid JSON.) Attempt 2: Sending gentle Chinese reminder...*
Action (Retry): Answer[call vision: Video 1, focus: "Describe Video A's egg wash preparation..."]
Observation: Visual Agent Return: The egg wash is prepared in large metal mixing bowls. The person uses hands to mix the egg wash.
Turn 6
Thinking: Now I need to check Video B's egg wash preparation. The caption mentions using a fork. I'll focus on their egg wash process.
Action: Answer[call vision: Video 2, focus: "Describe Video B's egg wash preparation..."]
Observation: Visual Agent Return: The egg wash is contained in a rectangular metal tray. A fork is used to stir the egg wash.
Turn 7 (Final)
Thinking: After 6 rounds of observation, I've collected sufficient evidence to compare the breading tools between Video A and Video B. The key differences lie in container types, mixing tools, and egg wash preparation methods.
Action: [Answer]
Final Answer: 1. Breading Containers: Video A uses large metal mixing bowls; Video B uses rectangular metal trays. 2. Mixing Tools: Video A uses hands only; Video B uses a fork to stir egg wash and hands for pressing breadcrumbs. 3. Egg Wash Preparation: Video A mixes beaten eggs by hands; Video B uses a fork in a metal tray. 4. Breadcrumb Application: Both use the pressing method, but Video B explicitly mentions using a fork.

Failure Analysis Note: This case highlights the vulnerability of LLMs to "instruction drift" during extended multi-turn interactions. As the context window fills with lengthy multimodal observations, the Master Agent occasionally loses its strict JSON formatting constraints. However, this example also effectively demonstrates the necessity and robustness of our system-level auto-recovery mechanism. By detecting the malformed output and injecting a targeted prompt intervention, the system successfully nudges the agent back on track without causing a catastrophic failure of the entire reasoning chain.

Incomplete Answer Identification

Query: *Which animals' current behaviors make them more likely to be detected by predators that hunt using visual movement cues?*
Options: A. Video one. B. Video two. C. Video three. D. Video four.
Ground Truth: A, B, C
Turn 1
Thinking: I need to scan Video 1 to confirm if the animal is stationary or moving regularly (e.g., foraging), as such behaviors might reduce the risk of predation due to a lack of suddenness. Frequent head turning or sudden sprints might expose it.
Action: Answer[call vision: Video 1, focus: "Identify if the animal is stationary, slowly moving, or exhibiting sudden movements. Note any head-turning, tail-twitching, or abrupt directional changes."]
Observation: Visual Agent Return: The walruses are not entirely stationary, as their positions shift slightly. They are slowly moving with gradual shifts on the rocky shore. There are no rapid or jerky motions , and no sudden movements or abrupt directional changes are observed. Water splashes are smooth and incremental. Head-turning is a slow, deliberate action. They lack prominent tails. The conclusion is that they are slowly moving.
Turn 2
Thinking: I need to scan Video 2. Since the video is only 1.83 seconds long, I must quickly identify key behaviors. Stationary or regular actions (like slow chewing) are less likely to expose the animal, whereas sudden actions (like fast escape or rapid blinking) make it easier for visual predators to lock on.
Action: Answer[call vision: Video 2, focus: "Is the animal stationary or moving? Are there sudden movements, rapid eye blinks, or erratic motions? Note any abrupt changes in posture or direction."]
Observation: Visual Agent Return: The shrimp-like creature is moving. There are subtle changes in its posture, limb positioning, and antennae orientation. The motion is slow and incremental. There are no sudden movements, and changes are gradual and smooth. Rapid eye blinks are irrelevant as it uses antennae for sensing. There are no erratic motions. The conclusion is that the animal is moving gradually and non-erratically.
Turn 3
Thinking: I need to scan Video 3, which is 4.83 seconds long, focusing on any sudden actions or regular movement. Fast running, frequent head turning, or sudden steering significantly increase the probability of dis-

cover by visual predators.
Action: Answer[call vision: Video 3, focus: "Detect any sudden movements, rapid directional changes, or frequent head-turning. Are the animal's motions smooth and predictable, or are there abrupt, erratic actions?"]
Observation: Visual Agent Return: The crab-like arthropod maintains a steady forward trajectory with no abrupt shifts in direction. It progresses smoothly along the mossy surface with a rhythmic, repetitive gait. Each step is gradual without jerky actions. The head and antennae show minimal, slow adjustments. The conclusion is that its motions are smooth and predictable.
Turn 4
Thinking: I need to scan Video 4, which is 4.625 seconds long, checking for sudden actions or regular movement. Behaviors like fast running, frequent head turning, or sudden steering increase the risk of being spotted.
Action: Answer[call vision: Video 4, focus: "Detect any sudden movements, rapid directional changes, or frequent head-turning. Are the animal's motions smooth and predictable, or are there abrupt, erratic actions?"]
Observation: Visual Agent Return: The animal, likely a wild boar in a forest , remains largely consistent in position with no abrupt shifts or rapid directional changes. Its head is mostly lowered for foraging, with slow and gradual minor adjustments. Its movements are smooth and predictable, maintaining a steady posture with subtle, incremental motions. No sudden jumps or rapid head-turning are observed.
Turn 5 (Final)
Thinking: Based on the analysis: 1. Video 1 (Walrus): Moving slowly, no sudden movements, not easily detected. 2. Video 2 (Shrimp): Moving slowly, but limbs are adjusting frequently, which might attract predator attention due to regular actions. 3. Video 3 (Crab): Steady direction, repetitive actions, highly predictable, which might lower the risk of detection. 4. Video 4 (Boar): Mostly stationary, only slow foraging actions, not easily exposed. Conclusion: None of the animals exhibit obvious sudden actions, but the shrimp in Video 2 is easier to detect by visual predators due to frequent limb adjustments.
Action: [Answer]
Final Answer: B

Failure Analysis Note: Although the Master Agent accurately perceived the subtle visual cues (movement vs. stationary) across all four videos, it failed at the final reasoning stage due to "Premature Cognitive Closure." The model exhibited an exclusivity bias—treating a multiple-choice question (where multiple answers could be correct) as a single-best-choice question. It erroneously concluded that only the most active subject (Video 2) was the answer, ignoring Videos 1 and 3 which also met the detection criteria. This reveals a critical gap between accurate low-level multimodal perception and complex high-level logical alignment in multi-label scenarios.

Incorrect Answer

Query: *Review these video clips. They are shuffled segments from a single cooking video. Please determine their correct chronological order.*
Ground Truth: 5->3->4->2->1
Model Prediction: 3->5->4->2->1
Turn 1
Thinking: Task started. I need to first get the captions for Video 1 to get a preliminary understanding of the action.
Action: Answer[call get_caption: Video 1]
Observation: Audio Agent Return: Mentions adding watercress, putting udon on a platter, and adding garnishes like carrots, green onions, and cilantro.
Turn 2
Thinking: Continuing to get captions. Now I need to get the captions for Video 2 to understand its action.
Action: Answer[call get_caption: Video 2]
Observation: Audio Agent Return: Mentions adding a sauce mixture, taking blanched udon noodles, and lightly stir-frying them.
Turn 3
Thinking: Continuing to get captions. Now I need to get the captions for Video 3 to understand its action.
Action: Answer[call get_caption: Video 3]
Observation: Audio Agent Return: Mentions making a quick dressing using soy sauce, oyster sauce, ponzu, red miso paste, and sake, and dissolving it.
Turn 4
Thinking: Continuing to get captions. Now I need to get the captions for Video 4 to understand its action.
Action: Answer[call get_caption: Video 4]
Observation: Audio Agent Return: Mentions adding red pepper, onions, more oil, ginger, garlic, and green onion.
Turn 5
Thinking: Continuing to get captions. Finally, get the captions for Video 5 to understand its action.
Action: Answer[call get_caption: Video 5]
Observation: Audio Agent Return: Mentions heating a skillet, adding oil, and cooking shiitake mushrooms on high heat.
Turn 6
Thinking: Now I need to observe the overall visual of Video 1 to confirm the plating action and the final state of the ingredients. I will check for any obviously unfinished actions or leftover ingredients.
Action: Answer[call vision: Video 1, focus: "Describe the final state of the dish..."]
Observation: Visual Agent Return: The udon is fully cooked. Watercress and other ingredients are arranged on the platter. Raw watercress is visible in the final plated dish.
Turn 7
Thinking: I need to observe the overall visual of Video 2 to confirm whether the udon noodles are already cooked and in the stir-frying stage. I'll focus on the noodle state and cooking actions.
Action: Answer[call vision: Video 2, focus: "Describe the state of the udon noodles..."]
Observation: Visual Agent Return: The udon noodles are fully cooked (blanched) and are being stir-fried in a preheated pan with the sauce and vegetables.
Turn 8
Thinking: I need to observe the overall visual of Video 3 to confirm the sauce preparation process. I'll focus on mixing and dissolving actions, and the state of the ingredients.
Action: Answer[call vision: Video 3, focus: "Describe the preparation of the sauce..."]
Observation: Visual Agent Return: The ingredients are mixed together, the miso is dissolved, and there are no visible raw or unprocessed ingredients in the sauce preparation.
Turn 9
Thinking: I need to observe the overall visual of Video 4 to confirm the processing of red peppers and onions. I'll check if they are chopped, being sautéed, and mixed with sauce.
Action: Answer[call vision: Video 4, focus: "Describe the processing of red peppers and onions..."]
Observation: Visual Agent Return: The red peppers and onions are sliced, being sautéed in a pan, and interacting with oil/sauce during the cooking process.
Turn 10
Thinking: I need to observe the overall visual of Video 5 to confirm the heating and initial cooking steps. I'll check if oil is heated, the order of adding ingredients, and if it logically connects to subsequent steps.
Action: Answer[call vision: Video 5, focus: "Describe the initial cooking steps..."]
Observation: Visual Agent Return: The pan is heated with oil. Oil is poured first, then mushrooms are added. There is no visible interaction with the sauce or ingredients from Video 4.
Turn 11 (Final)
Thinking: Based on the analysis: Video 3 prepares the seasoning (mixing raw sauces), which is an early stage. Video 5 heats oil and sautés mushrooms (initial cooking). Video 4 sautés red peppers and onions, which should follow mushrooms to utilize the oil. Video 2 stir-fries udon (mixed cooked noodles + sauce) which follows the vegetable sautéing. Video 1 is plating (final step). The logical chain is: Seasoning (3) -> Heat/Mushrooms (5) -> Sauté veggies (4) -> Stir-fry noodles (2) -> Plating (1).
Action: [Answer]
Final Answer: 3->5->4->2->1

Failure Analysis Note: The model failed due to a rigid assumption about standard cooking workflows. It incorrectly assumed that all prep work (making the sauce in Video 3) must chronologically precede the heating of the pan (Video 5). In the ground truth, the chef first heats the oil and mushrooms (Video 5), takes a moment to mix the sauce while the pan heats (Video 3), then returns to the

pan to add peppers (Video 4), noodles and sauce (Video 2), and finally plates the dish (Video 1). The model missed the visual and chronological continuity between the steps, relying too heavily on general procedural common sense rather than the specific continuity cues between the video clips.

H Prompt Summary

H.1 Prompts for Script Synthesis ($\mathcal{W}_{script}$)

In this section, we detail the prompt templates used to drive LLMs to synthesize video scripts for various tasks. To ensure the complexity, visual density, and logical rigor of the generated data, we designed tailored prompt structures for different CVR tasks.

H.1.1 Multi-view Spatial Reasoning (MSR)

For the Multi-view Spatial Reasoning task, the prompt strictly mandates the introduction of "Environmental Occlusions" to ensure that the scripts generated for the two views are complementary in terms of visual information.

MSR Prompt

Role
You are the **Lead Data Simulator for the MSR (Multi-view Spatial Reasoning) Benchmark**.
Your task is to generate high-difficulty synthetic data that tests if an AI can spatially reason across two synchronized camera views despite visual obstructions.
Scenario Settings
* **Scene**: <scenario>
* **Traffic Density**: <density>
* **Camera Configuration**: <camera_desc>* **Total Duration**: <duration>seconds.
* **Sampling Rate**: Detailed description every 2 seconds.

CRITICAL INSTRUCTION: OCCLUSION & HARD MODE
To make this task challenging, you MUST introduce **Environmental Occlusions**:
1. **Define Obstacles**: Explicitly state what blocks the view (e.g., "A large oak tree covers the NW corner of View A", "A parked delivery truck blocks the lower lane in View B").
2. **The "Hidden" Entity**: Ensure at least one key object passes BEHIND an obstacle in one view but remains visible in the other.
3. **Entity Consistency**: Use {A1}={B1}, {A2}={B2} mapping.

Task 1: Generate Synchronized Video Scripts
Generate a timeline (0s to <duration>s) with a step of 2 seconds.
For EACH timestamp, provide:
* **View A Visual**: Describe positions, relative distances, and **occlusions**.
* **View B Visual**: Describe the same reality from the other angle.

Task 2: Generate 1 Complex MSR Question
Create ONE single-choice question focused on **Spatial Reasoning**.
* **Focus**: Relative Position, Trajectory Prediction, or Blind Spot Inference.
* **Requirement**: The answer MUST derive from understanding **BOTH** videos.

Output JSON Format
Strict JSON format specifying meta, video_scripts, and question with reasoning

H.1.2 Multi-view Object Counting (MOC)

The prompt for the Multi-view Object Counting task focuses on cross-view object deduplication and the design of distractors to test the model's tracking and counting capabilities in complex scenarios.

MOC Prompt

Role
You are the **Lead Data Simulator for the MOC (Multi-view Object Counting) Benchmark**.
Your task is to generate high-difficulty synthetic data that tests if an AI can correctly count objects across synchronized videos, handling duplicates and occlusions.

Scenario Settings
* **Scene**: <scenario>
* **Traffic Density**: <density>
* **Camera Configuration**: <camera_desc>
* **Total Duration**: <duration>seconds.

CRITICAL INSTRUCTION: COUNTING COMPLEXITY & OCCLUSION
To make this challenging:
1. **Occlusion Zones**: Define zones where objects vanish in one view but are visible in the other.
2. **Cross-View Transition**: Objects must move from View A's field of view into View B's.
3. **Distractors**: Include objects that SHOULD NOT be counted (e.g., "Count only Red Cars", but include Red Trucks and Blue Cars).
4. **Deduplication**: {A1} and {B1} are the SAME physical car. The AI must count it as 1, not 2.

Task 1: Generate Synchronized Video Scripts
Generate a timeline with a step of 2 seconds. Provide View A Visual, View B Visual, and a Tracking Note (e.g., "{A1} is now {B1}").

Task 2: Generate 1 Complex MOC Question
Create ONE single-choice question focused on **Object Counting** (Global Unique Count or Conditional Count).

Output JSON Format
Strict JSON format specifying meta, video_scripts, and question

H.1.3 Process Sorting

This prompt is utilized to generate unordered segments of continuous processes, such as cooking, and requires the model to provide dense visual detail descriptions for subsequent use in the temporal sorting task.

Sorting Prompt

Role
You are the **Director of a High-Precision Video Reasoning Benchmark**. Your goal is to generate a **chronologically ordered** cooking script for "<recipe>".

Strict Configuration
* **Total Phases**: Exactly **<num_steps>**
* **Difficulty**: <difficulty>
* **Visual Focus**: <focus>

Task
Break down the cooking process into **<num_steps>Distinct Sequential Phases**.
Inside EACH Phase, list **8-15 Atomic Visual Events** (Micro-Actions).

Critical Requirements
1. **Strict Structure**: You MUST generate exactly <num_steps>phases.
2. **Focus Adherence**: Your visual descriptions MUST emphasize "<focus>".
3. **Logical Flow**: The phases must follow the correct timeline (Irreversible logic).
4. **Atomic Detail**: Do NOT write "He cooks the meat". Break it down: "Places pan", "Meat hits pan", "Searing sound", "Flipping meat".
5. **Dense Visuals**: Describe the **texture, color, and consistency** of the food at that exact moment. Show progression explicitly.

Output JSON Format
Strict JSON format with phases, events, visuals, and captions

H.1.4 Plot Inference (Missing Middle)

This task requires the generation of a narrative structure containing the beginning and ending segments but with a missing middle segment, in order to test the model's causal inference abilities.

Plot Inference Prompt

Role
You are a **Screenwriter**.
Task: Create a **Plot Inference (Missing Middle)** challenge.
* **Genre**: <genre>

* **Setting**: <setting>
* **Core Conflict (The Missing Event)**: <conflict>

Structure
* **Act 1 (Beginning)**: Setup.
* **Act 2 (HIDDEN Middle)**: Duration <gap_duration>seconds. The "<conflict>" happens here. DO NOT write the script, but ensure Act 3 reflects its consequences.
* **Act 3 (Ending)**: Aftermath. Visual state must change drastically due to Act 2.

INSTRUCTION: FILL THE TIMELINE
I have pre-calculated the time slots. **You must generate 'visual' and 'caption' for EACH slot.**

Act 1 Slots (Beginning):
<begin_slots_str>
Act 3 Slots (Ending):
<end_slots_str>

Content Requirements (Strict)
1. **Visual**: Focus ONLY on Narrative Content (Characters, Actions, Environment, Plot). NO camera/lighting jargon.
2. **Caption**: Dialogue ONLY. Empty if silent.

Option Generation (Crucial)
Generate 6 distinct plot summaries for the missing Act 2. One Correct Answer, 5 plausible but incorrect Distractors.

Output JSON Format
Strict JSON format with question, options, correct_answer, and scripts

H.1.5 Movie Understanding (Hard Single-Choice)

Aimed at the retrieval and understanding of specific clues in long videos, this requires the generation of four video scripts that are highly similar in setting and action but differ in critical details.

Movie Understanding Prompt

Role
You are a **Benchmark Dataset Designer**.
Your task is to create a **Hard Single-Choice Video Understanding Problem**.
Genre: **"<genre>"**.

Input Elements (The Target Scenario)
The **Correct Answer** must depict this specific scenario:
<keywords_str>

CRITICAL CONSTRAINT: THE LOGIC PUZZLE (Strict Single-Choice)
You are generating 4 different video clips (A, B, C, D), but **ONLY ONE** is the correct answer.
1. **Target Video**: Must perfectly integrate ALL the Input Elements into the plot.
2. **Distractor Videos**: Must belong to the same Genre and style. They should be confusingly similar (e.g., same setting but different action), but MUST FAIL to match the full description.

VISUAL & FORMAT CONSTRAINTS
1. **Independent Durations**: <video_instructions>
2. **Variable Pacing**: Use natural editing (Short Cuts 3-8s + Long Takes 15-30s).
3. **Content Requirements**: A natural, descriptive sentence of what is seen (Action + Camera + Environment) and subtitles/sound effects.

Output JSON Format
Strict JSON format with logic_adaptation, question, correct_answer, and timelines for 4 videos

H.1.6 Video Grounding / Alignment

This is used to generate two parallel video scripts with subtle visual or action differences, to test spatial and temporal alignment as well as grounding capabilities.

Grounding Prompt

Role
You are the **Chief Editor of a Video Alignment Benchmark**.
Your goal is to generate a "Parallel Script" for two different videos (Video A and Video B) cooking the SAME recipe: "<recipe>".

Diversity Injection
* **Alignment Granularity**: <granularity>
* **Variation Style**: <variation>
* **Difficulty**: <difficulty>

Task
Break the cooking process into **<num_steps>Distinct Functional Steps**.
For EACH step, generate TWO variations (Video A vs Video B) that perform the **SAME FUNCTION** but look **DIFFERENT**.

Critical Requirements
1. **Enforce Variation**: You MUST apply the "<variation>" rule rigorously.
2. **Enforce Granularity**: The descriptions must focus on the "<granularity>" level.
3. **Dense Visuals**: Describe texture, color, and consistency. Show progression.
4. **Micro-Events**: For each step, list 3-8 micro-events for Video A and 3-8 for Video B.

Output JSON Format
Strict JSON format with alignment_steps, function_desc, and dual video events

H.1.7 Cooking Action (Hard Negative)

Generates dense scripts containing atomic-level micro-actions, used to distinguish highly similar distractor videos.

Cooking Prompt

Role
You are the **Director of a High-Precision Video Reasoning Benchmark**. Your goal is to generate "Hard Negative" training data that mimics real-world cooking videos.

Task Assignment
- **Recipe**: <recipe>
- **Focus**: <difference_type>
- **Target**: <config_label>
- **Style**: <config_style>

1. The Question (Single-Choice)
Create a difficult **Single-Choice Question** where Option <config_letter>is the ONLY correct answer based on visual details or caption nuances.

2. Video Scripts (CRITICAL: HYPER-GRANULARITY & VISUAL DENSITY)
* **Atomic Action Decomposition (3-8s per segment)**: Break actions down into micro-steps.
* **Dense Visuals**: Describe the State (texture, color) and Show Progression explicitly. Include Visual Noise (tools, steam).
* **The Logical Trap**: The Target must contain the specific evidence. Distractors must look 90% similar but fail on the specific detail required.

Output JSON Format
Strict JSON format with multiple video timelines mapping dense visuals

H.1.8 Behavior and Intent Understanding

By defining different intents and context modifiers, this generates action scripts that capture complex differences in human or animal intentions.

Behavior Prompt

Role
You are an **Expert Examiner** creating a Behavior Understanding dataset.

Task Specs
* **Type**: <task_type>
* **Target Behavior/Activity**: "<target_activity>"
* **Context/Nuance**: "<context_modifier>"
* **Distractor Candidates**: <distractors>

Step 1: Scenario Expansion
1. **Target Scenario**: Describe the target specifically performed in the required manner.
2. **Distractor Scenarios**: Choose behaviors that fit the same Subject/Setting but imply a **different intent**.

Step 2: Formulate Question (Intent-Based)
Write a concise question focusing on the **Specific Intent** or **Nuance**. (e.g., "Which video depicts a lion resting due to exhaustion rather than just sleeping?")

Step 3: Script Generation
Generate 4 videos. Exactly <num_correct>Videos must depict the Target Scenario. The others must be Distractors. Provide dense visual descriptions with no captions.

Output JSON Format
Strict JSON format with expanded_scenario, question, correct indices, and video content

H.1.9 Assembly Task Error Identification

Based on specific assembly operations, performers are assigned different personas to generate action scripts containing specific assembly errors.

Assembly Prompt

Role
You are an expert scriptwriter for Video Understanding Benchmarks.
Task: Generate detailed visual scripts for 3 videos of different people assembling the same toy.

Context:
- Toy: <toy_id>
- Parts Sequence (Standard SOP): [<parts_str>]
- **Goal:** In ALL videos, the user MUST eventually complete the FULL assembly.

SCRIPTING RULES (STRICT):
1. **Full Process**: Start from loose parts ->Finish with the completed toy.
2. **Granularity**: ONE clip = ONE distinct action (‘ATTACH‘, ‘DETACH‘, ‘ADJUST‘, ‘INSPECT‘, ‘SEARCH‘, ‘IDLE‘).
3. **Visual Detail**: Describe the hands, specific parts, and physics.

ERROR LOGIC (Apply based on Config):
- **wrong_order**: Attaches B. Realizes A is missing. DETACHES B. Attaches A. Re-attaches B.
- **previous_one_is_mistake**: Fails to attach C because B is loose. DETACHES B. Fixes it. Attaches C.
- **shouldn’t_have_happened**: Picks up unnecessary object, realizes error, puts it back.
- **wrong_position**: Attaches to wrong slot. DETACHES. Attaches to correct slot.

Video Configs:
<video_requirements_and_personas>

QA GENERATION:
Design a Single Choice Question (SCQ) based on observing the *process* or *errors* (e.g., Error attribution, Persona comparison).

Output JSON Format:
Strict JSON format with meta, question, options, and structured clips with action_type

H.2 Prompts for the Dynamic Simulator (M_{sim})

During the construction of the dynamic simulator (M_{sim}), we adopted differentiated processing strategies for tools of different modalities. For audio tools, to ensure the complete accuracy and objectivity of the content, we directly extracted and utilized the subtitle information from the video script as the output. For visual tools, we employed preset system prompts to guide the large language model, enabling it to accurately simulate the analytical behavior of advanced computer vision tools based on the video script. The complete prompt used for the visual tool simulation is as follows:

Prompt

You are a video analysis simulator. Your task is to act as an advanced computer vision tool.
You will receive a "video script (Ground Truth)" and a user’s "query". Please simulate the output of a computer vision tool based on the content of the video script.

Note:

Do not reveal that you are reading a script; respond as if you have actually watched the video.

Generate your responses strictly according to the timestamps and action descriptions provided in the script.

If the user asks about a time frame where no action occurs in the script, please reply: "No significant action observed."

H.3 Master Agent Prompts

This section details the Master Agent prompts to guide MLLMs in executing multi-turn video analysis and reasoning tasks. We categorize these tasks into four main dimensions: Comparative Analysis, Temporal Understanding, Multi-view Reasoning, and Free-form QA.

H.3.1 Comparative Analysis

BU (Behavior Understanding) Prompt

You are a behavioral science and ethology expert. Your task is to handle a **Behavior (behavioral intent recognition)** problem.

Task Objective
Your goal is to **answer the question correctly**.
Note: This is a multiple-select task.
Videos that meet the description may be **1, or 2 to 3**, but you cannot select all four ABCD options.
Your task is not to "select the best one", but to **"find all videos that meet the criteria"**.
You need to analyze the **intent** and **function** behind actions through visual evidence.

Available Tools
1. **observe** (the only visual tool)
· **Parameters**: *observation_targets* (list), *focus_prompt* (instructions describing what you are looking at).
· **Limitation**: The total number of frames (*num_frames* sum) in a single operation **must not exceed 128**.

Operation Examples (Json Examples)

Example 1: Global Horizontal Comparison (Establishing Baseline Behavior)
Scenario: Task begins.
The question asks "Which video shows a **defensive** posture?". First need to scan all 4 videos to see what everyone is doing and establish a baseline.
{
"action": "observe",
"thought": "The question asks about ’defensive posture’. I first need to overview all 4 videos to establish baseline behavior. The subjects in Videos 1, 2, 3, 4 all seem to be ’dogs’. I need to compare their first reactions when ’a stranger approaches’: tail wagging (friendly), ignoring (neutral), or backing away/showing teeth (defensive/fearful)?",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 0, "end_time": 10, "num_frames": 16},
{"video_index": 2, "start_time": 0, "end_time": 10, "num_frames": 16},
{"video_index": 3, "start_time": 0, "end_time": 10, "num_frames": 16},
{"video_index": 4, "start_time": 0, "end_time": 10, "num_frames": 16}
],
"focus_prompt": "Compare the reaction to the approaching human. Focus on tail position (wagging high vs tucked between legs) and ear position (perked up vs flattened back)."
}
}

Example 2: Cross-Video Differential Analysis (Screening Candidates)
Scenario: Turn 1 observation found that dogs in Videos 1 and 3 are both backing away.
Videos 2 and 4 are wagging tails (excluded). Now need to compare Videos 1 and 3 to distinguish "fear" from "aggressive defense".
{
"action": "observe",
"thought": "Videos 2 and 4 show friendliness, already excluded. Now focus on comparing Videos 1 and 3. Both are backing away, but the question is looking for ’defensive’ rather than pure ’escape’. I need to compare the body center of gravity and facial orientation of these two dogs. Who maintains eye contact with the target while backing away (ready to counterattack), and who directly turns and runs away?",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 5, "end_time": 15, "num_frames": 32},
{"video_index": 3, "start_time": 5, "end_time": 15, "num_frames": 32}
],
"focus_prompt": "Compare the retreat style. Is the dog turning away to flee (Flight), or backing up while maintaining eye contact and baring teeth (Defense)?"
}
}

Example 3: Single Video Detail Drilling (Verifying Micro-expressions/Key Features)
Scenario: Turn 2 leans towards Video 3 being defensive because it faces the target.
But to be absolutely sure, need to zoom in on Video 3’s neck fur and pupils, which is decisive evidence for confirming "high alert state".
{
"action": "observe",
"thought": "[Turn 3] Video 3 looks most like ’defensive’ behavior. For final confirmation, I need to drill into Video 3 alone. I need to zoom in on its back and neck area to confirm whether there is ’hackles raised’ phenomenon, which is a physiological feature of extreme tension and defense.",

"params": {
"observation_targets": [
{"video_index": 3, "start_time": 8, "end_time": 12, "num_frames": 32}
],
"focus_prompt": "Zoom in closely on the dog's neck and back ridge. Are the hairs standing up (piloerection)? Also check if the lips are curled to show gums."
}
}

Example 4: Final Answer (Multiple Selection Scenario)
Scenario: The question may allow double selection.
Assume in the just check, it was found that although Video 1 is backing away, it also showed defensive posture in the last few seconds.
{
"action": "answer",
"thought": "Comprehensive analysis:
1. Global comparison: Videos 2/4 are friendly.
2. Differential analysis: Video 3 shows standard defensive posture (staring, showing teeth, hackles raised).
3. Detail review: Video 1 initially was escaping, but when cornered at the end, also turned towards the target and lowered its center of gravity, fitting 'passive defense' characteristics.
4. Conclusion: Both Videos 1 and 3 exhibited defensive behavioral intent.",
"final_answer": ["A", "C"]
}

Final Answer Format:
Must be in JSON format.
The *final_answer* field must be a string composed of option letters, directly concatenated without spaces or symbols, and arranged in alphabetical order.
Select 1-3 options, no need to not select or select all four options
Correct examples: "A" or "AC" or "BCD"

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object.
Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

NC (Narrative Comprehension) Prompt

You are an extremely insightful **film narrative and emotional analysis expert**.
Your task is to observe **4 input videos** (Videos 1, 2, 3, 4) through multiple rounds based on the user's question (Query) and optional answers (Options), collect plot clues, and finally give the correct answer.

Working Mode Instructions:
- This is a **multi-turn dialogue task**, you need to **call tools step by step** to collect evidence, rather than giving an answer all at once.
- **Only output one JSON operation at a time**.
- **Strictly prohibited to output multiple JSONs at once**.

Available Tools
1. **observe**: Observe video frame details (main tool).
2. **get_caption**: Get subtitle information (auxiliary tool, used to obtain dialogues and narrations).

Tool Details
1. observe
Purpose: Observe video frames
Parameters:
· *observation_targets*: List[Dict] (*video_index*, *start_time*, *end_time*, *num_frames*)
* **Tip**: List contains 1 object = single video deep dive; List contains >1 objects = multi-video comparison.
· *focus_prompt*: str (tell the vision model what to look at, must be specific!)

2. get_caption
Purpose: Get subtitle text
Parameters:
· *video_index*: int (1, 2, 3, or 4)
· *start_time*: float (optional)
· *end_time*: float (optional)

Operation Examples (Json Examples)

Example 1: Forced Initialization
Scenario: Task begins, indiscriminately retrieve dialogues to build the plot skeleton.
{
"action": "get_caption",
"thought": "Task begins. As a film analysis expert, I must first get Video 1's dialogues to understand the plot background.",
"params": { "video_index": 1 }
}

Example 2: Visual Overview Based on Subtitle Clues
Scenario: The question is "Which video tells the story of an undercover police officer's identity being exposed?". Subtitles show that both Videos 2 and 4 have arguments involving "police" and "traitor".
{
"action": "observe",
"thought": "According to subtitle clues, Videos 2 and 4 both involve 'police' and 'traitor' dialogues, they are key suspects. I need to visually overview these two videos, describe the plot development in detail, see who was discovered, and what the consequences were.",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 0, "end_time": 300, "num_frames": 32},
{"video_index": 4, "start_time": 0, "end_time": 300, "num_frames": 32}
],
"focus_prompt": "Describe the narrative arc of these clips in detail regarding the 'undercover' theme. Does a character get caught? Is there an interrogation or a chase? Describe the key plot points chronologically."
}
}

Example 3: Single Video Detail Drilling
Scenario: The question is "What is the protagonist's true emotion when reading the letter?". In Video 1, he reads the letter expressionlessly, need to drill deeper.
{
"action": "observe",
"thought": "At 120s-130s in Video 1, the protagonist is reading a letter. Although he neither cries nor smiles, I need to capture his micro-expressions to judge whether he is 'numb and desperate' or 'secretly pleased'. This will determine his true attitude towards the plot.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 120, "end_time": 130, "num_frames": 64}
],
"focus_prompt": "Zoom-in on the man's face while reading. Analyze his micro-expressions. Are his eyes trembling? Is his jaw clenched? Does he look relieved or devastated? Describe the subtle emotional cues."
}
}

Example 4: Multi-Video Side-by-Side Comparison
Scenario: The question is "Which film most extensively uses night scenes to create an oppressive atmosphere?" (degree comparison question).
{
"action": "observe",
"thought": "This is a degree comparison question. Can't judge 'most' by looking at one video alone. I need to simultaneously input key segments from all 4 videos, compare their tones and lighting horizontally, and find the one with the highest proportion of night scenes and darkest atmosphere.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 50, "end_time": 150, "num_frames": 32},
{"video_index": 2, "start_time": 50, "end_time": 150, "num_frames": 32},
{"video_index": 3, "start_time": 50, "end_time": 150, "num_frames": 32},
{"video_index": 4, "start_time": 50, "end_time": 150, "num_frames": 32}
],
"focus_prompt": "Compare the lighting and atmosphere side-by-side. Which video predominantly uses low-key lighting, shadows, and night settings? Rank them from darkest to brightest."
}
}

Example 5: Final Answer
{
"action": "answer",
"thought": "After analysis:
1. Subtitles suggest V2 and V4 have undercover plots.
2. Turn 5 visual overview confirms Video 4 is undercover escaping successfully, while Video 2 is undercover being executed.
3. Turn 7 detail drilling shows V2 protagonist's expression before death is relieved.
4. The question asks 'which video ends in tragedy with protagonist's sacrifice', only Video 2 meets the criteria.",
"final_answer": "B"
}

Key Constraints
Final Answer Format:
Must be in JSON format.
If it's a multiple-choice question, the *final_answer* field must contain only one uppercase letter (such as "A", "B", "C", "D").
✓Correct example: "final_answer": "D"
All reasoning processes and reasons must be written in the *thought* field.

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object.
Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe* or *get_caption*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

CC (Culinary Comparison) Prompt

You are a high-precision culinary detail and process monitoring expert with eagle eyes.
Your task is to address fine-grained single-choice questions (usually involving ingredient texture, tool material, or subtle differences in action sequences) by comparing and observing 4 highly similar input videos (Hard Negatives) through multiple rounds, eliminating extremely similar distractors, and finally precisely locking onto the only correct answer that matches the description.

Working Mode Instructions:
· This is a **multi-turn dialogue task**, you need to **call tools step by step** to collect evidence, rather than giving an answer all at once.
· **Only output one JSON operation at a time**.
· **Strictly prohibited to output multiple JSONs at once**.

Available Tools
1. **observe**: Observe video frame details (main tool).
2. **get_caption**: Get subtitle information (auxiliary tool, used to obtain dialogues and narrations).

Tool Details

1. observe
Purpose: Observe video frames
Parameters:
· *observation_targets*: List[Dict] (*video_index*, *start_time*, *end_time*, *num_frames*)
* **Tip**: List contains 1 object = single video deep dive; List contains >1 objects = multi-video comparison.
· *focus_prompt*: str (tell the vision model what to look at, must be specific!)

2. get_caption
Purpose: Get subtitle text
Parameters:
· *video_index*: int (1, 2, 3, or 4)
· *start_time*: float (optional)
· *end_time*: float (optional)

Operation Examples (Json Examples)

Example 1: Forced Initialization and Subtitle Retrieval
Scenario: Task just started, to understand what each of the four videos is doing (or their subtle differences in narration), first retrieve the dialogue text.
{
"action": "get_caption",
"thought": "Task begins. The question asks about differences in 'seasoning order'. I first need to get the subtitles of all videos to see if the chef explicitly mentions 'add salt first' or 'add salt later' when explaining steps, to build a preliminary timeline.",
"params": { "video_index": 1 }
}
Note: In actual operation, may need to retrieve subtitles from different videos multiple times

Example 2: Visual Overview Based on Subtitle Clues
Scenario: The question asks "Which video uses **Fresh Basil** instead of dried basil?". Subtitles in Videos 2 and 3 both mention "Adding basil". Need visual confirmation whether it's green leaves (fresh) or crumbs (dried).
{
"action": "observe",
"thought": "According to subtitles, Videos 2 and 3 both mention 'adding basil' at 45s-60s. Whether fresh or dried, subtitles may just say 'Basil'. I need to locate this time segment and judge the physical form of the ingredient through visual overview (whole green leaves or bottled crumbs).",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 45, "end_time": 60, "num_frames": 16},
{"video_index": 3, "start_time": 45, "end_time": 60, "num_frames": 16}
],
"focus_prompt": "Identify the ingredient being added. Is it fresh green leaves being torn by hand, or dry flakes shaken from a jar?"
}
}

Example 3: Single Video Detail Drilling (State Check)
Scenario: The question asks "In which video is the butter cooked to a browned state (Brown Butter)?". Preliminary investigation found that the butter in Video 1 looks darker, need to confirm whether it's browned or just a lighting issue.
{
"action": "observe",
"thought": "The butter in the pan in Video 1 at 30s looks darker. I need to zoom in on the foaming state and sediment to confirm whether it shows 'hazelnut color' with brown sediment (characteristics of browned butter), or just melted.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 28, "end_time": 35, "num_frames": 32}
],
"focus_prompt": "Zoom in on the melting butter. Look for 'nutty' brown milk solids at the bottom and specific foaming. Confirm if it is 'beurre noisette'."
}
}

Example 4: Multi-Video Side-by-Side Comparison (Difference Analysis)
Scenario: The question asks "Which video uses a serrated knife to cut tomatoes?". This is a tool detail question, it's hard to judge by looking at one video alone, need to put the "cutting tomatoes" segments from all 4 videos together for horizontal comparison of blade characteristics.
{
"action": "observe",
"thought": "This is a tool detail question. All videos are cutting tomatoes, I need to simultaneously capture the moment when 'the blade contacts the tomato skin' in all 4 videos and compare blade characteristics horizontally. I'm looking for a knife with a serrated edge, not a smooth-edged chef's knife.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 15, "end_time": 25, "num_frames": 16},
{"video_index": 2, "start_time": 18, "end_time": 28, "num_frames": 16},
{"video_index": 3, "start_time": 12, "end_time": 22, "num_frames": 16},
{"video_index": 4, "start_time": 20, "end_time": 30, "num_frames": 16}
],
"focus_prompt": "Focus strictly on the knife blade. Is the edge smooth or serrated (saw-like)? Compare the blade shapes side-by-side."
}
}

Example 5: Final Answer
{
"action": "answer",
"thought": "Comprehensive analysis:
n1. Subtitles suggest all videos made scrambled eggs with tomatoes.
n2. Turn 3 observation excluded V2 and V3 (used dried basil, doesn't match the question).
n3. Turn 7 side-by-side comparison shows that Videos 1, 2, 4 all use smooth-edged chef's knives, only Video 3 clearly uses a bread knife/tomato knife with wavy serrations to process ingredients.
n4. The question asks to find the video using a 'serrated knife', Video 3 is the only match.",
"final_answer": "C"
}

Key Constraints
Final Answer Format:
Must be in JSON format.
If it's a multiple-choice question, the *final_answer* field must contain only one uppercase letter (such as "A", "B", "C", "D").
Correct example: "final_answer": "D"
All reasoning processes and reasons must be written in the *thought* field.

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object.
Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe* or *get_caption*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

PEA (Procedural Error Analysis) Prompt

You are an industrial process and behavioral analysis expert. Your task is to observe 3 silent videos, each showing different users assembling the same toy (such as an excavator or truck) from start to finish.
You need to precisely identify logical errors in the assembly process through pure visual cues to answer a single-choice question about toy assembly.
Your core ability is to identify the reason behind the key action of "Detach" - whether it is to correct an error or a normal disassembly step.

Working Mode Instructions:
· This is a **multi-turn dialogue task**, you need to **call tools step by step** to collect evidence, rather than giving an answer all at once.
· **Only output one JSON operation at a time**.
· **Strictly prohibited to output multiple JSONs at once**.

Available Tools
1. **observe** (the only visual tool)
· **Parameters**: *observation_targets* (list), *focus_prompt* (instructions describing what you are looking at).
· **Limitation**: The total number of frames (*num_frames* sum) in a single operation **must not exceed 128**.

Operation Examples (Json Examples)

Example 1: Forced Initialization and Proficiency Profiling (Persona Profiling)
Scenario: Task begins.
The question asks "Which user is a Novice?". You need to quickly scan the first 30 seconds of the three videos to establish a baseline profile through the fluency of actions.
{
"action": "observe",
"thought": "The question requires identifying the 'novice'. This is a pure visual task. I need to compare the first 30 seconds of the three videos. I need to observe: Who hesitates before picking up parts? Who frequently looks back and forth between the manual and parts? Whose actions are 'pick and install' (Expert)?",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 0, "end_time": 30, "num_frames": 16},
{"video_index": 2, "start_time": 0, "end_time": 30, "num_frames": 16},
{"video_index": 3, "start_time": 0, "end_time": 30, "num_frames": 16}
],
"focus_prompt": "Analyze hand movement confidence. Count distinct pauses (Idle). Check gaze shifts between 'Table Parts' and 'Manual'. Look for 'search' behavior."
}
}

Example 2: Locating Key Abnormal Actions (The "Detach" Event)
Scenario: The question asks "Why did the user in Video 2 **detach** the wheel they just installed around 120s?".
You need to zoom in on the actions before the detachment to find the root cause of the error.
{
"action": "observe",
"thought": "This is a question about 'error correction'. Users detaching parts usually means they made a mistake. I need to trace back to the 10 seconds before the detachment action (110s-120s) to see what happened during installation. Was it

because the screw wasn't aligned? Or because they missed installing an internal washer?",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 110, "end_time": 125, "num_frames": 32}
],
"focus_prompt": "Focus on the 'Wheel' attachment point. Did the part click into place? Did the user try to insert a screw but fail? Look for a 'missing internal part' lying on the table."
}
}

Example 3: Cross-Video Specific Step Comparison
Scenario: The question asks "Which video made a 'direction error' when installing the **bulldozer bucket**?".
You need to locate the time segments where the "bucket" appears in all three videos for horizontal comparison.
{
"action": "observe",
"thought": "I need to find the segments where all videos install the 'bucket'. Since assembly speeds differ, I first need to search for the corresponding time points in each video based on the part shape (yellow, large, bucket-shaped), then compare whether the bucket orientation is correct (the concave side should face forward).",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 50, "end_time": 150, "num_frames": 32},
{"video_index": 2, "start_time": 50, "end_time": 150, "num_frames": 32},
{"video_index": 3, "start_time": 50, "end_time": 150, "num_frames": 32}
],
"focus_prompt": "Locate the 'Bucket' installation. Check its orientation relative to the Chassis arms. Does the user install it, pause, examine it, and then flip it over?"
}
}

Example 4: Single Video Detail Verification (Physical Interaction Analysis)
Scenario: Turn 3 suspects Video 1 installed it backwards because they pushed hard several times but it didn't snap in. Need to zoom in on the physical feedback at the "snap" point.
{
"action": "observe",
"thought": "The user in Video 1 seems to be struggling when installing the bucket. I need to closely observe the connection point. Is it a tolerance issue with the parts, or did they completely misalign the slot? Observe the user's hand force - is it 'smooth guidance' or 'forceful push'?",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 95, "end_time": 105, "num_frames": 32}
],
"focus_prompt": "Zoom in on the joint connection. Is there a physical obstruction? Does the user shake the part in frustration? Does the part eventually snap in, or is it removed?"
}
}

Example 5: Final Answer
{
"action": "answer",
"thought": "Comprehensive process analysis:
1. Video 1 (Expert): Actions are smooth and fluid, picks up the bucket and directly aligns it with the slot, succeeds in one attempt, no extra actions.
2. Video 3 (Novice): Hesitates for 5 seconds after picking up the bucket, finds it won't close after installation, then performs a 'Detach' action, flips the bucket and reinstalls successfully.
3. Video 2: Actions are fast but rough, forcefully jams the bucket in, no correction.
4. The question asks 'who made a direction error and corrected it', only Video 3 matches the description.",
"final_answer": "C"
}

Key Constraints
Final Answer Format:
Must be in JSON format.
If it's a multiple-choice question, the *final_answer* field must contain only one uppercase letter (such as "A", "B", "C", "D").
Correct example: "final_answer": "D"
All reasoning processes and reasons must be written in the *thought* field.

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object.
Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

H.3.2 Temporal Understanding

PI (Plot Inference) Prompt

You are a logical reasoning and narrative reconstruction detective. Your task is to infer from the options what key events occurred in the (missing middle part) that led to these results, by keenly capturing "irreversible changes" in character states (injuries, emotions), environmental backgrounds (destruction, weather), and object positions, based on the video's (beginning) and (ending) segments.

Working Mode Instructions:
· This is a **multi-turn dialogue task**, you need to **call tools step by step** to collect evidence, rather than giving an answer all at once.
· **Only output one JSON operation at a time**.
· **Strictly prohibited to output multiple JSONs at once**.

Available Tools
1. **observe**: Observe video frame details (main tool).
2. **get_caption**: Get subtitle information (auxiliary tool, used to obtain dialogues and narrations).

Tool Details

1. observe
· **Purpose**: Observe video frames
· **Parameters**:
- *observation_targets*: List[Dict] (*video_index*, *start_time*, *end_time*, *num_frames*)
· **Tip**: List contains 1 object = single video deep dive; List contains >1 objects = multi-video comparison.
- *focus_prompt*: str (tell the vision model what to look at, must be specific!)

2. get_caption
· **Purpose**: Get subtitle text
· **Parameters**:
- *video_index*: int (1, 2, 3, or 4)
- *start_time*: float (optional)
- *end_time*: float (optional)

Operation Examples (Json Examples)

Example 1: Forced Initialization and Getting Key Dialogues
Scenario: Task begins. You need to first listen to what people are saying in the ending (Act 3), because dialogues often contain direct references or complaints about "what just happened".
{
"action": "get_caption",
"thought": "The task objective is to infer the missing middle plot. By getting the ending segment's (Video 2) dialogues, I hope to hear the characters' discussion about the accident that just happened, such as 'this damn trap' or 'who took the map', to narrow down the reasoning range.",
"params": {
"video_index": 2,
"start_time": 0,
"end_time": 60
}
}

Example 2: Visual Overview Based on Subtitle Clues
Scenario: In the subtitles, the protagonist shouts "Fire! Put out the fire!" at the ending. You need to quickly browse the ending frames to confirm the source and extent of the fire, whether it's sparks from electrical short circuit, or a big fire from arson?
{
"action": "observe",
"thought": "Subtitles explicitly mention 'fire'. Now I need to observe Video 2's (ending) overall environment. I need to confirm the fire source location - is it limited to a specific electrical device (suggesting short circuit, Option A), or is the entire room burning (suggesting explosion or arson, Option C).",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 0, "end_time": 30, "num_frames": 16}
],
"focus_prompt": "Describe the fire and smoke. Is it coming from a specific device (e.g., a computer) or is it a general room fire? Look for scorch marks on the walls."
}
}

Example 3: Single Video Detail Drilling (Micro Trace Analysis)
Scenario: The question asks what fight happened in the middle. The protagonist's hand is bleeding at the ending. You need to zoom in on the wound to judge whether it's a sharp object cut (corresponding to "knife fight") or scrape/bruise (corresponding to "hand-to-hand combat").
{
"action": "observe",
"thought": "The protagonist's hand injury is key evidence. I need to zoom in on Video 2's frame where the protagonist is holding the wound. Through the wound shape (clean cut vs irregular edges) and bleeding amount, to judge what weapon caused it.",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 15, "end_time": 20, "num_frames": 32}
],
"focus_prompt": "Zoom in on the hand injury. Is it a clean slice (indicating a knife/glass) or a bruise/scrape (indicating a fall or punch)? describe the blood pattern."
}
}

Example 4: Multi-Video (Cross-Time) Side-by-Side Comparison
Scenario: The question core is about object loss or position movement. You need to put Video 1 (beginning) and Video 2 (ending) of the same scene together for comparison to find the inconsistency.
{
"action": "observe",
"thought": "This is an object state change question. In Video 1, there's a red vase on the table. In Video 2, the table is overturned. I need to simultaneously observe the 'table area' at these two time points to confirm whether the vase is ***shattered on the floor*** *(suggesting impact) or* ***disappeared into thin air*** *(suggesting theft).",*
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 50, "end_time": 60, "num_frames": 16},
{"video_index": 2, "start_time": 0, "end_time": 10, "num_frames": 16}
],
"focus_prompt": "Compare the 'table area' in both clips. In the second clip, are there ceramic shards on the floor? Or is the vase completely missing?"
}
}

Example 5: Final Answer
{
"action": "answer",
"thought": "Reasoning chain:
1. Subtitle clue: At the ending, the protagonist mentions 'it suddenly jumped out from the ventilation shaft'.
2. Visual comparison: In Video 1, the ventilation cover is intact, in Video 2 the cover is on the floor, and there's green slime around the ventilation shaft.
3. Detail analysis: The protagonist has scratches on their arm, not burns.
4. Option matching: These evidences rule out 'earthquake' (Option A) and 'gas leak' (Option B), perfectly pointing to 'alien creature attack' (Option D).",
"final_answer": "D"
}

Key Constraints & Final Answer Format:
Must be in JSON format.
If it's a multiple-choice question, the *final_answer* field must contain only one uppercase letter (such as "A", "B", "C", "D").
Correct example: "final_answer": "D"
All reasoning processes and reasons must be written in the *thought* field.

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object.
Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe* or *get_caption*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

FSA (Functional Step Alignment) Prompt

You are an extremely rigorous and precise video temporal grounding expert.
Your task is to find a **functionally equivalent** time segment in Video B based on a reference segment in Video A.

Working Mode Instructions:
· This is a **multi-turn dialogue task**, you need to **call tools step by step** to collect evidence, rather than giving an answer all at once.
· **Only output one JSON operation at a time**.
· **Strictly prohibited to output multiple JSONs at once**.

Available Tools
1. **observe**: Observe video frame details (main tool).
2. **get_caption**: Get subtitle information (auxiliary tool, used to obtain dialogues and narrations).

Tool Details

1. observe
Purpose: Observe video frames
Parameters:
· *observation_targets*: List[Dict] (*video_index*, *start_time*, *end_time*, *num_frames*)
* **Tip**: List contains 1 object = single video deep dive; List contains >1 objects = multi-video comparison.
· *focus_prompt*: str (tell the vision model what to look at, must be specific!)

2. get_caption
Purpose: Get subtitle text
Parameters:
· *video_index*: int (1, 2, 3, or 4)
· *start_time*: float (optional)
· *end_time*: float (optional)

Operation Examples (Json Examples)

Example 1: Forced Initialization and Reference Action Semantic Analysis
Scenario: Task begins. The question provides an interval [15.5, 22.0] in Video 1 (Reference). You must first figure out what Video 1 is doing during this time and extract the "functional semantics".
{
"action": "observe",
"thought": "The question requires aligning Video 1's [15.5, 22.0] to Video 2. I must first observe this reference segment in Video 1, ignore specific tool details, and extract the core action's 'functional semantics' (for example: is it whipping egg whites, or just mixing evenly?).",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 15.5, "end_time": 22.0, "num_frames": 16}
],
"focus_prompt": "Analyze the action strictly by its function. What is the goal? (e.g., 'Emulsifying the sauce' vs 'Boiling water'). Ignore that the chef is using a whisk."
}
}

Example 2: Rough Localization Based on Subtitle Clues
Scenario: Confirmed the reference action is "chopping onions". Video 2 is 10 minutes long, to narrow the search range, first pull Video 2's subtitles to see where "Chop the onion" is mentioned.
{
"action": "get_caption",
"thought": "The reference action is confirmed to be 'chopping onions'. Video 2 is very long, to avoid blind searching, I'll first get Video 2's subtitles, looking for keywords 'onion', 'dice', 'chop', to quickly lock onto potential time candidate areas.",
"params": {
"video_index": 2,
"start_time": 0,
"end_time": 600
}
}

Example 3: Visual Overview and Semantic Search (Ignoring Appearance Differences)
Scenario: Subtitles suggest 45s-90s may be relevant. Now need visual confirmation. The difficulty is that Video 2 may be completed instantly using a food processor, looking completely different from Video 1's manual chopping.
{
"action": "observe",
"thought": "Subtitles point to Video 2's 45s-90s. I need to scan this interval. Note: Video 2 is a 'high-tech version' variant, may have used a food processor. I need to look for the state transition moment when 'whole onion' becomes 'minced', rather than staring at the 'knife'.",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 45, "end_time": 90, "num_frames": 32}
],
"focus_prompt": "Find the segment where the onion changes state from whole to chopped. Watch for a machine being turned on or a quick cut. Ignore the absence of a knife."
}
}

Example 4: Dual Video Side-by-Side Verification and Boundary Refinement
Scenario: Found a candidate segment [52.0, 58.0] in Video 2. To ensure accuracy, compare Video 1 (Ref) and Video 2 (Candidate) side-by-side to align the "start point" and "end point" of the action.
{
"action": "observe",
"thought": "I've locked onto [52, 58] in Video 2. Now fine-tune the boundaries through side-by-side comparison. Video 1 starts from 'knife touching the skin'. The corresponding start point in Video 2 should be the moment of 'pressing the machine button'; Video 1 ends at 'putting down the knife', Video 2 corresponds to ending at 'pouring out the minced pieces'.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 15.5, "end_time": 22.0, "num_frames": 16},
{"video_index": 2, "start_time": 50.0, "end_time": 60.0, "num_frames": 32}
],
"focus_prompt": "Compare the 'Start of Action' and 'End of Action' events side-by-side. Align the functional start (processing begins) and functional end (processing stops)."
}
}

Example 5: Final Answer
{
"action": "answer",
"thought": "After analysis:
1. Video 1's reference segment [15.5, 22.0] shows 'manual onion chopping'.
2. Although Video 2 uses a food processor (Visual Variation), it completes the exact same 'onion crushing' task in the [53.2, 57.8] interval.
3. Start and end points have been semantically aligned.",
"final_answer": [53.2, 57.8]
}

Key Constraints
Final Answer: Only output [start, end] list.

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object.
Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe* or *get_caption*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

PSS (Procedural Step Sequencing) Prompt

You are an extremely rigorous culinary video temporal reconstruction expert. Your task is to receive a set of **shuffled video segments** (usually 3-6), and restore them to the correct chronological order by analyzing **Cooking Logic** and **Semantic Content**.

Working Mode Instructions:
- This is a **multi-turn dialogue task**, you need to **call tools step by step** to collect evidence, rather than giving an answer all at once.
- **Only output one JSON operation at a time**.
- **Strictly prohibited to output multiple JSONs at once**.

Available Tools
1. **observe**: Observe video frame details (main tool).
2. **get_caption**: Get subtitle information (auxiliary tool, used to obtain dialogues and narrations).

Tool Details

1. observe
Purpose: Observe video frames
Parameters:
· *observation_targets*: List[Dict] (*video_index*, *start_time*, *end_time*, *num_frames*)
* **Tip**: List contains 1 object = single video deep dive; List contains >1 objects = multi-video comparison.
· *focus_prompt*: str (tell the vision model what to look at, must be specific!)

2. get_caption
Purpose: Get subtitle text
Parameters:
· *video_index*: int (1, 2, 3, or 4)
· *start_time*: float (optional)
· *end_time*: float (optional)

Operation Examples (Json Examples)

Example 1: Forced Initialization and Action Labeling
Scenario: Received 4 shuffled video segments, first need to know what specific actions are happening in each segment for preliminary classification (is it preparing ingredients, cooking, or plating?).
{
"action": "observe",
"thought": "Task begins. I need to quickly browse all 4 segments and label each segment with an 'action tag'. This will help me establish a preliminary logical framework (for example: chopping vegetables must be before stir-frying).",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 0, "end_time": 10, "num_frames": 8},
{"video_index": 2, "start_time": 0, "end_time": 10, "num_frames": 8},
{"video_index": 3, "start_time": 0, "end_time": 10, "num_frames": 8},
{"video_index": 4, "start_time": 0, "end_time": 10, "num_frames": 8}
],
"focus_prompt": "Summarize the main action in each clip. Is it chopping, washing, frying, eating, or plating?"
}
}

Example 2: Using Physical State to Judge Pairwise Order (Pairwise Sorting)
Scenario: Turn 1 found Videos 1 and 3 are both processing meat. Video 1 is "meat sizzling in the pan, surface brown", Video 3 is "bright red raw meat pieces being put into the pan". Need to establish V3 -> V1 logical connection.
{
"action": "observe",
"thought": "Videos 1 and 3 have similar actions, but I need to determine the order. Based on physical common sense, raw meat (red) must come before cooked meat (brown). I need to compare their 'doneness' and 'pan state'.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 0, "end_time": 5, "num_frames": 16},
{"video_index": 3, "start_time": 0, "end_time": 5, "num_frames": 16}
],
"focus_prompt": "Assess the doneness of the meat. Is it raw/red or cooked/brown? Is the pan dry or is there already oil and steam?"
}
}

Example 3: Checking Irreversible Mixing Process (Entropy Check)
Scenario: Now need to determine the order of "beating eggs" and "stirring batter". Video 2 shows only flour in the bowl, Video 4 shows a mixture of flour, eggs and milk in the bowl.
{
"action": "observe",
"thought": "Check the mixing state of substances. Video 2 shows dry powder, Video 4 shows mixed liquid. According to the principle of entropy increase, once substances are mixed they cannot naturally separate. Therefore, 'pure state' (Video 2) must come before 'mixed state' (Video 4).",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 0, "end_time": 5, "num_frames": 16},
{"video_index": 4, "start_time": 0, "end_time": 5, "num_frames": 16}
],
"focus_prompt": "Inspect the contents of the bowl. Is it just dry flour, or a wet mixture of multiple ingredients? Look for traces of eggs."
}
}

Example 4: Using Subtitles to Assist in Judging Final Step
Scenario: Videos 1 and 4 both look like close-ups of the finished product. Hard to judge which is "taking out of the pan" and which is "tasting". Listening to the dialogues might be more direct.
{
"action": "get_caption",
"thought": "Videos 1 and 4 look visually similar. I need to listen to the audio. If someone says 'Finally done' or hears chewing sounds, that must be the last step.",
"params": {
"video_index": 1,
"start_time": 0,
"end_time": 10
}
}

Example 5: Final Answer
{
"action": "answer",
"thought": "Logical reconstruction chain:
n1. Video 2: Whole potato (original state).
n2. Video 4: Cutting potato (physical destruction, V2 -> V4).
n3. Video 3: Frying potato strips (raw to cooked, color changes from white to golden, V4 -> V3).
n4. Video 1: Sprinkling salt, plating and tasting (final decoration, V3 -> V1).
nThis order conforms to physical irreversibility logic.",
"final_answer": "2->4->3->1"
}

Key Constraints
Only output one JSON at a time.

Final Answer Format:
Must be in JSON format.
The *final_answer* field must be a string composed of numbers and arrows.
Strictly prohibited to include "Video" wording or explanations.
Correct example: "3->5->4->2->1"

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object. Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe* or *get_caption*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

H.3.3 Multi-view Reasoning

MSR (Multi-view Spatial Reasoning) Prompt

You are an aerial reconnaissance and spatial intelligence expert. Your task is to combine 2 synchronized UAV perspective videos to infer the precise spatial location or movement trajectory of targets.

Core Challenge: The question usually involves "blind spot reasoning". For example, the target is blocked by a building in View A, and blocked by a tree in View B. You must synthesize two incomplete information sources to deduce the target's true position in 3D space.

Available Tools
1. **observe** (the only visual tool)
· **Parameters**: *observation_targets* (list), *focus_prompt* (instructions describing what you are looking at).
· **Limitation**: The total number of frames (*num_frames* sum) in a single operation **must not exceed 128**.

Operation Examples (Json Examples)

Example 1: Forced Initialization and Blind Spot Construction
Scenario: Task begins. The question asks "What is the exact position of the red sedan at T=15s". First need to identify each perspective's "visual blind spots".
{
"action": "observe",
"thought": "This is a spatial reasoning task. View A is blocked by a tall building on the left, View B is blocked by a billboard on the right. I need to first establish an 'environmental occlusion model', clarifying which areas View A cannot see but View B can.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 0, "end_time": 5, "num_frames": 8},
{"video_index": 2, "start_time": 0, "end_time": 5, "num_frames": 8}
],
"focus_prompt": "Identify static obstacles (buildings, trees) and the shadow zones they create in each view."
}
}

Example 2: Dynamic Tracking and Vanishing Point Inference
Scenario: The target drives north in View A, then disappears behind a building. Need to use View B to confirm whether it passed through the building or stopped behind it.
{
"action": "observe",
"thought": "The target enters the building's blind spot in View A at T=12s. At this point View A cannot provide information. I need to immediately observe View B's frames at T=12s-15s. If View B shows no car coming out from the north side of the

building, then the car must have stopped at the loading dock behind the building.",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 12, "end_time": 16, "num_frames": 16}
],
"focus_prompt": "Watch the area immediately North of the building. Does the red car emerge? Or does it remain unseen?"
}
}

Example 3: Multi-Perspective Triangulation
Scenario: The question asks "What is the pedestrian's position relative to the bus". View A shows the pedestrian is "to the left or behind" the bus (perspective compression), View B shows the pedestrian is "behind" the bus. Comprehensive inference.
{
"action": "observe",
"thought": "A single perspective lacks depth information. View A (front view) shows the pedestrian and bus overlapping on the left-to-right axis. View B (side view) clearly shows the pedestrian is 2 meters behind the bus's rear. Combining both, I can determine the pedestrian is 'directly behind' the bus.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 20, "end_time": 25, "num_frames": 16},
{"video_index": 2, "start_time": 20, "end_time": 25, "num_frames": 16}
],
"focus_prompt": "Determine the relative depth. Use View B to resolve the depth ambiguity seen in View A."
}
}

Example 4: Final Answer
{
"action": "answer",
"thought": "Spatial logic deduction:
1. View A shows the target entered the tunnel entrance (disappeared).
2. View B shows no vehicles exited the tunnel exit during the same time period (T=15s-20s).
3. Combining both perspectives: the target is neither outside the entrance nor outside the exit.
4. Conclusion: The target is currently located inside the tunnel.",
"final_answer": "C"
}

Key Constraints
Only output one JSON at a time.
Final Answer Format: "final_answer": "C".

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object. Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

MOC (Multi-view Object Counting) Prompt

You are a UAV surveillance and traffic analysis expert. Your task is to observe 2 synchronized UAV perspective videos (View A and View B, covering the same area but from different angles) and accurately count the global unique quantity of specific targets.

Core Challenge: You need to solve the cross-camera re-identification (Re-ID) problem.
If a red car appears in View A and also appears in View B, you must determine whether they are the same car (if yes, count only 1 time; if not, count 2 times).
At the same time, you need to use complementary perspectives to solve occlusion problems.

Available Tools
1. **observe** (the only visual tool)
· **Parameters**: *observation_targets* (list), *focus_prompt* (instructions describing what you are looking at).
· **Limitation**: The total number of frames (*num_frames* sum) in a single operation **must not exceed 128**.

Operation Examples (Json Examples)

Example 1: Forced Initialization and Spatial Overlap Analysis
Scenario: Task begins.
The question asks "How many blue trucks passed through this area in total?".
First need to figure out the spatial relationship between View A and View B (are they adjacent, opposite, or perpendicular?), to determine the overlapping area.
{
"action": "observe",
"thought": "This is a dual-perspective counting task. First I need to understand the spatial relationship between the two cameras and determine where the 'overlapping area' is. If View A is top-down and View B is oblique, then the center of View A may correspond to the bottom of View B.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 0, "end_time": 10, "num_frames": 8}, # View A
{"video_index": 2, "start_time": 0, "end_time": 10, "num_frames": 8} # View B
],
"focus_prompt": "Analyze the scene layout. Identify common landmarks (e.g., a central roundabout or a large billboard) visible in BOTH views to establish the spatial overlap."
}
}

Example 2: Cross-Perspective Re-identification and De-duplication
Scenario: At T=20s, there's a blue car on the left side of View A, and also a blue car on the right side of View B. Need to determine whether they are the same car (Count=1) or two different cars (Count=2).
{
"action": "observe",
"thought": "Blue trucks appear in both View A and View B. I need to judge whether they are the same target through 'temporal synchronization' and 'driving direction'. If the car in View A is driving east, while the car in View B is driving west, then they must be two different cars.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 18, "end_time": 22, "num_frames": 16},
{"video_index": 2, "start_time": 18, "end_time": 22, "num_frames": 16}
],
"focus_prompt": "Compare the blue trucks. Check their exact movement vector and relative position to the landmarks identified in Turn 1. Are they the same physical object?"
}
}

Example 3: Using Complementary Perspectives to Solve Occlusion (Occlusion Handling)
Scenario: In View A, a car drives under a bridge and disappears.
View B can see under the bridge from the side. Need to confirm whether the car stopped under the bridge.
{
"action": "observe",
"thought": "View A shows the target 'disappeared' under the bridge (occluded). I need to switch to View B (side angle) to check the blind spot under the bridge, to confirm whether the vehicle drove out of the frame, or just stopped in the shadow under the bridge.",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 30, "end_time": 40, "num_frames": 32} # Only check View B
],
"focus_prompt": "Look specifically under the bridge structure. Is the vehicle visible there? Or is the area empty?"
}
}

Example 4: Final Answer
{
"action": "answer",
"thought": "Counting logic:
1. T=0-10s: View A and View B both see a blue truck (ID#1), determined to be the same target.
2. T=15s: View A sees a second blue truck (ID#2) entering, but View B is blocked by trees and didn't capture it, rely on View A for counting.
3. T=30s: View B sees a third car (ID#3) coming from the opposite direction.
4. Total: 3 unique blue trucks.",
"final_answer": "A"
}

Key Constraints
Only output one JSON at a time.
Final Answer Format: "final_answer": "C".

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object.
Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe*) to verify evidence before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".

H.3.4 Free-form QA

CCQA (Comparative Culinary QA) Prompt

You are an elite culinary comparative analyst and open-ended QA expert.
Your task is to observe 2 instructional cooking videos (Video 1 and Video 2) showing the preparation of the same recipe, and answer an open-ended question by deeply comparing their specific differences in ingredient processing, tool usage, procedural variations, timing, or flavor styles.

Working Mode Instructions:
· This is a **multi-turn dialogue task**, you need to **call tools step by step** to collect evidence, rather than giving an answer all at once.
· **Only output one JSON operation at a time**.
· **Strictly prohibited to output multiple JSONs at once**.

Available Tools
1. **observe**: Observe video frame details (main tool).
2. **get_caption**: Get subtitle information (auxiliary tool, used to obtain dialogues and narrations).

Tool Details
1. observe
Purpose: Observe video frames
Parameters:
· *observation_targets*: List[Dict] (*video_index*, *start_time*, *end_time*, *num_frames*)
* **Tip**: List contains 1 object = single video deep dive; List contains >1 objects = multi-video side-by-side comparison.
· *focus_prompt*: str (tell the vision model what to look at, must be specific!)

2. get_caption
Purpose: Get subtitle text
Parameters:
· *video_index*: int (1 or 2)
· *start_time*: float (optional)
· *end_time*: float (optional)

Operation Examples (Json Examples)

Example 1: Forced Initialization and Subtitle Retrieval
Scenario: Task begins. The open-ended question asks "How do the two videos differ in their methods of cooking the chickpeas?". You first need to find exactly when the chickpeas are handled in both videos.
{
"action": "get_caption",
"thought": "The task asks for a procedural comparison regarding 'chickpeas'. I need to first retrieve the subtitles for Video 1 to locate the exact timestamps where 'chickpeas' or 'beans' are mentioned, so I know where to focus my visual analysis.",
"params": {
"video_index": 1,
"start_time": 0,
"end_time": 600
}
}

Example 2: Visual Overview Based on Subtitle Clues (Video 1)
Scenario: You found the relevant timestamps for Video 1 in the subtitles. Now you need to visually confirm the exact cooking method used.
{
"action": "observe",
"thought": "Subtitles in Video 1 mention adding chickpeas around 45s-75s. I need to observe this segment to determine their exact cooking method. Do they boil them in a pot with a tea bag, roast them, or use a pressure cooker? I need concrete visual evidence.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 45, "end_time": 75, "num_frames": 32}
],
"focus_prompt": "Observe the cooking method of the chickpeas. What type of cookware is used? Are there any specific ingredients added to the water, like baking soda or tea bags?"
}
}

Example 3: Cross-Video Differential Analysis (Video 2)
Scenario: You have established Video 1's method. Now you must retrieve and observe the corresponding step in Video 2 to construct your comparative analysis.
{
"action": "observe",
"thought": "I know Video 1 boils the chickpeas with tea bags in a standard pot. Now I need to check Video 2's method. Subtitles indicate chickpea preparation happens between 20s and 50s. I need to observe this segment to see how it contrasts with Video 1.",
"params": {
"observation_targets": [
{"video_index": 2, "start_time": 20, "end_time": 50, "num_frames": 32}
],
"focus_prompt": "Identify the cooking vessel and method for the chickpeas. Is it a pressure cooker? Are they using canned chickpeas directly or boiling dried ones?"
}
}

Example 4: Side-by-Side Detail Drilling (Final Verification)
Scenario: You noticed a subtle difference in the final seasoning step and want to verify both videos simultaneously before generating the final text.
{
"action": "observe",
"thought": "Before concluding, I need to verify the final garnishing step. I will do a side-by-side comparison of the final plating moments in both videos to ensure my open-ended summary captures all nuances.",
"params": {
"observation_targets": [
{"video_index": 1, "start_time": 180, "end_time": 195, "num_frames": 16},
{"video_index": 2, "start_time": 240, "end_time": 255, "num_frames": 16}
],
"focus_prompt": "Compare the final garnish applied to the dish. Does Video 1 use cilantro while Video 2 uses parsley? Detail the visual differences."
}
}

Example 5: Final Answer (Open-Ended Comparative Text)
Scenario: You have collected enough evidence and are ready to answer the open-ended question.
{
"action": "answer",
"thought": "Comprehensive comparative analysis:
1. Subtitles and visual tracking confirmed the chickpea preparation steps in both videos.
2. In Video 1, the chef boils dried chickpeas in a standard pot and adds tea bags for color.
3. In Video 2, the chef uses a pressure cooker to speed up the process and does not add any tea bags.
4. Both videos ultimately use the chickpeas for chana masala, but the foundational prep completely differs. I will synthesize this into a clear descriptive paragraph.",
"final_answer": "Video 1 boils dried chickpeas in a standard pot along with tea bags to impart a darker color, whereas Video 2 utilizes a pressure cooker for a faster cooking process without the addition of tea bags."
}

Key Constraints
Final Answer Format:
Must be in JSON format.
Because this is an open-ended comparative task, the *final_answer* field MUST be a single descriptive string containing a comparative paragraph (3-5 sentences) summarizing your findings.
Do NOT output a single letter.
Correct example: "final_answer": "Video 1 uses pre-made Texas Toast fried in a pan, while Video 2 creates dough from scratch using wheat flour..."

CRITICAL EXECUTION RULES (MUST FOLLOW)
1. **JSON OUTPUT ONLY**: Every single response you generate must be a strict, valid JSON object. Do NOT output any conversational text or internal monologue outside the JSON.
2. **MINIMUM 4 TOOL CALLS**: You are REQUIRED to perform at least 4 distinct tool interactions (e.g., *observe* or *get_caption*) to thoroughly verify evidence across BOTH videos before you are allowed to call the *answer* tool.
3. **STRICT FORMAT COMPLIANCE**: Your output format must EXACTLY match the provided examples. Use specific keys: "action", "thought", and "params".