Videopus: PARALLEL MULTIMODAL EVIDENCE SEEKING FOR LONG-HORIZON VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

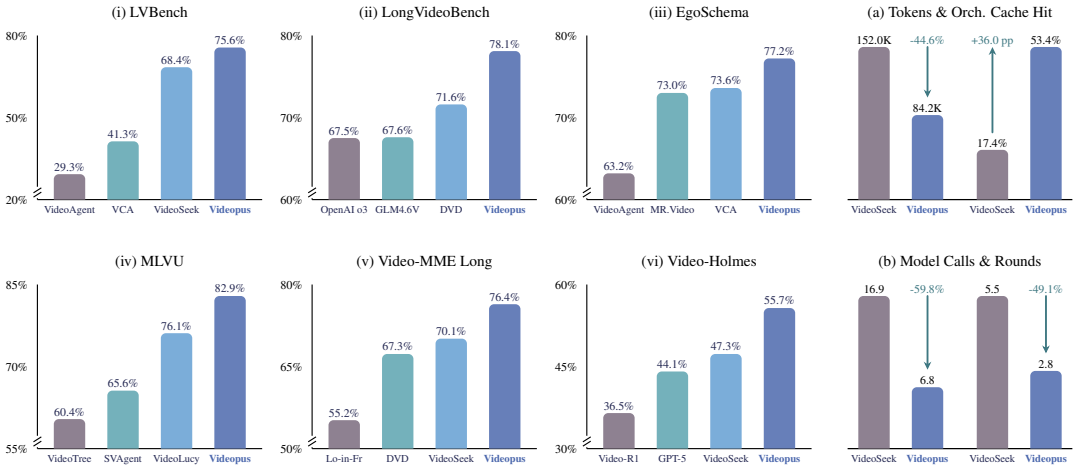


Figure 1: Panels (i)-(vi) report accuracy for representative methods; panels (a)-(b) report efficiency comparison on LVBench in average token use, Orchestrator cache hit, model calls, and round counts.

ABSTRACT

Long-horizon video understanding requires models to locate sparse evidence across extended timelines while preserving the visual detail needed for accurate reasoning. Existing agentic systems often inspect candidate intervals sequentially, repeatedly processing an interaction history. Moreover, local observations are usually summarized in text, making it difficult for the global reasoner to verify fine-grained visual details or connect evidence across intervals. We introduce Videopus, a multi-agent framework in which an Orchestrator coordinates interval-specific Executors. It has two components. Parallel Task Spawn groups inspections derived from the same evidence state into a concurrently executable batch. Multimodal Evidence Aggregation returns structured local findings with a compact set of supporting visuals selected from each Executor's input. The Orchestrator uses this evidence to resolve ambiguous descriptions and compare objects or event stages across the video. Local perception remains interval-specific, while global reasoning operates over a compact, visually inspectable evidence state. Videopus alternates parallel inspection with evidence-guided refinement. Across six video benchmarks, Videopus consistently outperforms the state-of-the-art methods while reducing token use and sequential rounds and increasing persistent-context reuse across reasoning turns.

1 INTRODUCTION

Long-horizon video understanding is fundamentally an evidence-allocation problem. Under a finite inference budget, a model must distribute perception across an extended timeline while preserving sufficient visual fidelity to resolve decisive local cues. These cues are often question-dependent and temporally sparse. An object-state change, a brief interaction, or an ordering relation may occupy only a few seconds of an hour-long video. Uniform sampling thus creates an inherent trade-off

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

between temporal coverage and spatial detail. Broader coverage reduces the visual capacity available to each observation, while finer detail requires concentrating the budget on fewer time points. Simply extending the context is not sufficient, since sampling, compression, and irrelevant content can still obscure decisive visual evidence under fixed budgets (Wang et al., 2025c; 2024a; Liu et al., 2024).

Existing methods address this problem either by preprocessing the full video before reasoning or by acquiring evidence interactively. Preprocessing methods first segment and process the entire video to construct captions, embeddings, or hierarchical memories. This requires full video-wide computation regardless of which evidence the question ultimately needs, while the resulting representation is fixed before question-dependent reasoning begins. Interactive agents instead access the video on demand and progressively localize useful intervals (Wang et al., 2024b; Lin et al., 2026; Zhang et al., 2025c). However, repeated tool calls often introduce serial orchestration, where each observation must return before the next search decision while the interaction history continues to grow. Even after useful intervals are located, local perception often reaches the global reasoner through captions or textual summaries (Yang et al., 2026b). Such text-only evidence can omit important visual attributes, hinder direct verification of ambiguous descriptions, and weaken entity correspondence across intervals.

These bottlenecks reinforce each other. Serial inspection forces the global reasoner to wait for each local observation before deciding what to inspect next, while text-only returns may still omit the fine-grained visual cues needed to resolve the question. Recovering missing cues then requires additional inspection rounds, further extending the interaction history and repeating global reasoning over prior evidence. Effective evidence acquisition should therefore combine parallel local inspection with selective visual return. Independent intervals can be processed concurrently, while a compact set of visuals allows the global reasoner to verify and compare their findings directly. This organization reduces serial interaction, repeated inspection, and token accumulation across reasoning cycles.

Based on this principle, we propose Videopus, a multi-agent framework for evidence-grounded long-video understanding. A bounded global sampling first provides coarse temporal orientation across the full video. Conditioned on the question and current evidence state, the Orchestrator then issues a batch of self-contained inspection requests. Each request specifies a candidate interval, an inspection objective, and either coverage mode or detail mode. Coverage mode emphasizes temporal coverage and event localization, whereas detail mode preserves the visual fidelity needed to examine short, ambiguous moments. Executors process these requests concurrently and return compact findings with selected supporting visuals. Tasks within a round share the same pre-round evidence state and can therefore execute independently, while evidence-dependent adaptation occurs between rounds. The Orchestrator aggregates the outputs of the batch, assesses the joint evidence against the unresolved question, and either answers or launches a more targeted batch to resolve the remaining uncertainty.

The main contributions of this work are threefold.

(1) We introduce Parallel Task Spawn, which groups objective-specific interval inspections from a shared pre-round evidence state into concurrent batches. This design preserves evidence-dependent refinement across rounds while reducing sequential global decisions and repeated context processing.

(2) We introduce Multimodal Evidence Aggregation, which pairs compact local findings with selected source visuals from each Executor's input. This allows the Orchestrator to verify ambiguous descriptions, preserve appearance and state cues, and compare evidence across distant intervals.

(3) We evaluate Videopus on six benchmarks spanning temporal, visual, and cross-segment reasoning tasks. Videopus consistently outperforms the latest methods in average accuracy. Efficiency analyses further show that Videopus requires fewer model calls, tokens, and reasoning rounds during inference.

2 Related Work

Video Long-Context Management. Video language models extend image-language backbones to temporal inputs (Lin et al., 2024; Zhang et al., 2024), while general-purpose VLMs support increasingly long multimodal contexts (Hurst et al., 2024; Comanici et al., 2025; Bai et al., 2025a). Long videos nevertheless remain difficult to process at uniform fidelity. Sparse sampling can miss brief cues, whereas dense inputs introduce redundancy and dilute relevant evidence (Wang et al., 2024a; Liu et al., 2024). Compression and memory-based methods reduce this load through token compression, adaptive hierarchies, or structured long-term representations (Song et al., 2024; Wang

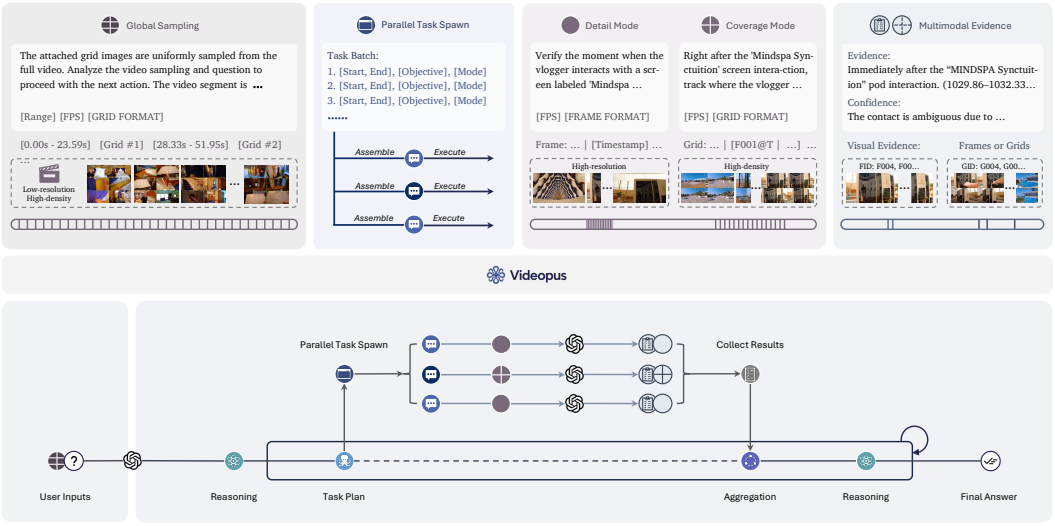


Figure 2: Videopus workflow. From a sparse overview, the Orchestrator forms interval-specific tasks and launches parallel inspections in coverage mode and detail mode. Executors return findings and source observations for batch aggregation before the Orchestrator answers or begins another round.

et al., 2025e;c; Li et al., 2026; Long et al., 2025; Lian et al., 2026; Chen et al., 2026). These designs reduce the active context, but later reasoning still depends on the information retained during compression, selection, or memory construction. Distributed processing offers another route by assigning different temporal regions to independent workers or sub-agents, reducing the context handled by each agent session (Pang & Wang, 2025; Yang et al., 2026b). The resulting local evidence must then be aggregated for global reasoning, and visual details omitted when local observations are reduced to textual summaries are unavailable for later verification or cross-interval comparison.

Adaptive Video Reasoning. End-to-end CoT and test-time scaling increase reasoning-time computation, while learned focusing adaptively allocates visual computation across the video (Xie et al., 2025; Wang et al., 2025d;a; Zhang et al., 2025b; Li et al., 2025b; Feng et al., 2025; Ding et al., 2025; Ouyang et al., 2025; Fu et al., 2025b). Agentic methods make evidence acquisition part of inference. Some construct video-wide memories, captions, or searchable representations before question-specific reasoning (Yeo et al., 2025; Pang & Wang, 2025; Zhang et al., 2025c; Ma et al., 2025). Processing the full video in advance incurs substantial preprocessing cost before the required evidence is known, while caption-based representations can omit visual details needed later. Other methods retrieve frames, documents, temporal intervals, or evolving memories on demand (Wang et al., 2024b; Yang et al., 2025a; Guo et al., 2025b; Lin et al., 2026; Yin et al., 2025), making evidence acquisition question-dependent without full-video preprocessing. Some agents search sequentially based on prior observations, with successive calls reprocessing a growing interaction history (Yang et al., 2025b; Zeng et al., 2026). Structured multi-agent systems improve coordination across specialized reasoning roles (Yang et al., 2026a; Liu et al., 2026). Multi-clip reasoning further reduces serial dispatch through concurrent interval inspection (Yang et al., 2026b), but these calls return textual summaries without interval-specific objective guidance or source visuals for verification.

3 METHODOLOGY

Figure 2 shows the workflow of Videopus. Given a video V of duration T and a question Q, the Orchestrator conditions on Q and bounded global sampling $S(V)$ to identify missing evidence and its likely temporal location. At reasoning round r, it uses $\mathcal{H}_{<r}$ from earlier rounds to either return an answer y or invoke Parallel Task Spawn to construct a batch of interval-specific inspections. Each inspection specifies a temporal interval, an inspection objective, and either coverage mode or detail mode for independent Executor processing. Executors process these requests in parallel and return structured findings together with selected source observations for global verification and comparison.

The Orchestrator maintains a persistent context containing the question, global sampling, and accumulated evidence across rounds. Each Executor receives only its interval-specific objective and sampled

Algorithm 1 Videopus inference with parallel inspection and multimodal evidence aggregation.

Require: Video V, question Q, budgets $J, B_{\text{glo}}, B_{\text{ret}}, R_{\max}$
$S(V) \leftarrow \textsc{GlobalSampling}(V, J, B_{\text{glo}}), \mathcal{H}_{<1} \leftarrow \varnothing$ ▷ coarse temporal orientation
for $r = 1, \ldots, R_{\max}$ **do**
  $d_r \leftarrow \textsc{Orchestrator}(Q, S(V), \mathcal{H}_{<r})$ ▷ answer required at $r = R_{\max}$
  if d_r is an answer y **then return** y ▷ early stop or final-round answer
  $\mathcal{B}_r \leftarrow d_r$, where $\mathcal{B}_r = \{I_i = (\tau_i, o_i, m_i)\}_{i=1}^{K_r}$
  for all $I_i \in \mathcal{B}_r$ **in parallel do** ▷ independent local inspections
    $X_i \leftarrow \text{Sample}(V, \tau_i, m_i)$ ▷ coverage grids or detail frames
    $(E_i, U_i, A_i) \leftarrow \text{Exec}(o_i, X_i)$ ▷ finding, reliability, selected IDs
    $\mathcal{V}_i \leftarrow \mathcal{R}(X_i, A_i, m_i)$, with $|\mathcal{V}_i| \le B_{\text{ret}}$ ▷ selected source visuals
  end for
  $\mathcal{H}_r \leftarrow \bigoplus_{i=1}^{K_r} [I_i, E_i, U_i, \mathcal{V}_i]$ ▷ temporally ordered batch evidence
  $\mathcal{H}_{<r+1} \leftarrow [\mathcal{H}_{<r}; \mathcal{H}_r]$ ▷ append to persistent context
end for

visual units in a temporary context that ends after execution. Multimodal Evidence Aggregation combines the batch of Executor outputs into a multimodal evidence state, which is then added to the persistent context. The Orchestrator assesses this updated evidence and either answers or begins a more targeted round based on the remaining uncertainty. This workflow combines parallel local perception with iterative evidence refinement. We define an image decoded from V as a source frame and each model image input as a visual unit, which contains either one source frame or a grid of timestamped source frames. Both model-input and returned-evidence budgets are measured in visual units. Algorithm 1 summarizes the synchronized round updates across the full inference trajectory.

3.1 Parallel Task Spawn

Global Sampling and Task Construction. Initial planning requires broad temporal coverage before relevant evidence can be localized. Sparse frame inputs leave large temporal gaps, while textual summaries can discard visual cues needed for later search. We therefore uniformly sample J timestamps from $[0, T)$ and pack the corresponding low-resolution source frames into at most B_{glo} chronological grids. Under a fixed visual-unit budget, this packing increases the number of observed timestamps at the cost of per-frame detail while preserving direct access to the visual content. The resulting global sampling $S(V)$ provides a question-independent visual reference for scene transitions, recurring activities, and their approximate temporal positions across the video. It remains accessible across reasoning rounds, allowing the Orchestrator to place newly acquired local evidence within the global timeline and identify candidate regions that require closer inspection. At round r, the Orchestrator constructs the next inspection batch conditioned on Q, $S(V)$, and $\mathcal{H}_{<r}$:

$$\mathcal{B}_r = \{I_i = (\tau_i, o_i, m_i)\}_{i=1}^{K_r}, \qquad \tau_i \subseteq [0, T), \qquad m_i \in \{\textsf{coverage}, \textsf{detail}\}, \tag{1}$$

where i indexes a task, τ_i is its temporal interval, o_i is its inspection objective, and m_i selects either coverage mode or detail mode. Each objective specifies the observable cue needed to resolve one uncertainty, such as an action, relation, text, or object-state change. The Orchestrator may inspect alternative locations for the same cue or seek complementary cues for one hypothesis. All requests use the same pre-round evidence and can therefore execute independently. Each objective carries the question-specific context needed to interpret local observations, so an Executor can identify the requested evidence within its assigned interval without accessing the global reasoning history.

Mode-Specific Inspection and Parallel Execution. Given a task I_i, its mode m_i determines how the visual budget is allocated within τ_i. The relevant uncertainty may concern insufficient temporal sampling or unclear spatial detail. A fixed observation configuration cannot adapt its allocation to these different needs, and higher resolution alone cannot recover events between sampled timestamps. Under the same visual-unit budget, our coverage mode packs more timestamped source frames into chronological grids, concentrating observations within τ_i to find relevant events and refine their temporal boundaries. The additional timestamps help distinguish nearby actions and follow event progression. Detail mode supplies fewer source frames as individual images, allocating more spatial

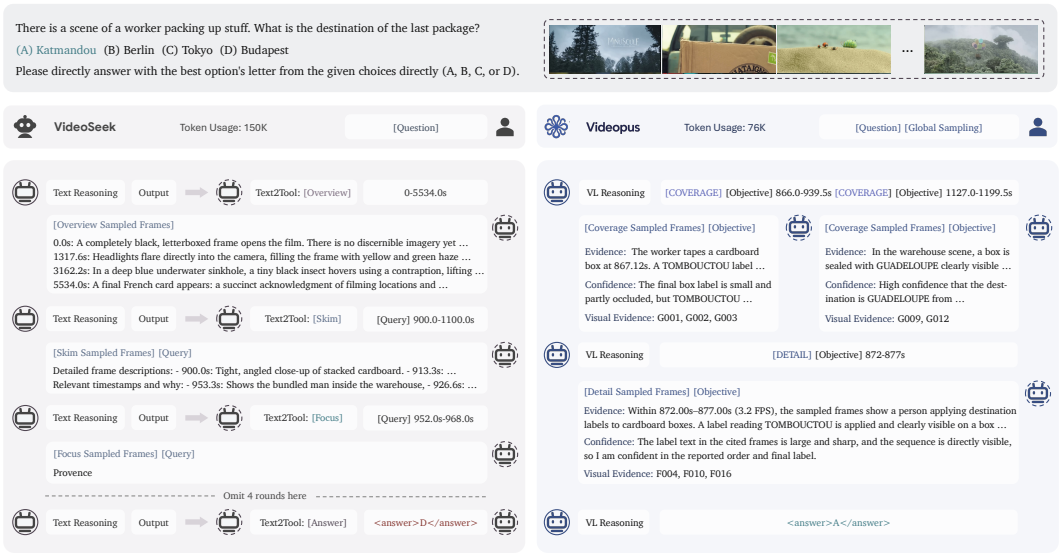


Figure 3: Comparison of VideoSeek and Videopus. The panels trace inspection path. Dashed and solid robots denote temporary and persistent contexts. The robot's eyes distinguish LLM and VLM.

capacity to small objects, scene text, and fine state changes that may be unclear in grids. A batch can combine coverage and detail requests when its objectives require different temporal and spatial evidence. Each request produces a mode-specific visual input, which the Executor inspects under the assigned objective to identify relevant observations and assess whether the requested cue is visible:

$$X_i = \text{Sample}(V, \tau_i, m_i), \qquad Z_i = \text{Exec}(o_i, X_i), \qquad i = 1, \ldots, K_r, \tag{2}$$

where X_i contains the locally indexed visual units sampled for task I_i, and Z_i is the resulting local inspection report. Executors process these tasks independently and concurrently. Each Executor operates on its assigned observations in a temporary context without cross-task communication. Grouping independent inspections before one global decision reduces serial orchestration and repeated context processing compared with deciding separately after each inspection. Appendix D analyzes how global input cost grows with reasoning rounds and accumulated evidence. The Orchestrator receives all reports before adapting its plan, so evidence-dependent decisions occur between rounds.

3.2 Multimodal Evidence Aggregation

Structured Multimodal Evidence. To support joint assessment of the parallel inspection results, each Executor structures its local report as $Z_i = (E_i, U_i, A_i)$, where E_i describes the task-relevant evidence visible in its sampled observations, U_i records reliability information such as visibility or sampling limitations, and A_i contains identifiers of the input observations most relevant to the finding. The objective o_i determines which observed cues are relevant to the requested inspection. When a cue remains unresolved, U_i records the sampling or visibility limitation to guide the next interval or mode choice. The identifiers in A_i link each finding back to its Executor inputs, allowing selected source observations to accompany the textual report for direct verification by the Orchestrator.

Selective Aggregation and Refinement. Using the identifiers A_i in each report, we retrieve the selected source observations as $\mathcal{V}_i = \mathcal{R}(X_i, A_i, m_i)$ in a mode-compatible format, with $|\mathcal{V}_i| \le B_{\text{ret}}$. Combining the reports with these observations yields the multimodal evidence for the current round:

$$\mathcal{H}_r = \bigoplus_{i=1}^{K_r} \left[I_i, E_i, U_i, \mathcal{R}(X_i, A_i, m_i) \right], \qquad |\mathcal{V}_i| \le B_{\text{ret}}, \tag{3}$$

where $\oplus$ orders reports by temporal interval and B_{ret} caps the returned visual units from each inspection. In detail mode, $\mathcal{V}_i$ contains individual frames; in coverage mode, it contains selected source grids so that each temporal anchor remains visible with its local neighborhood. Only these selected observations enter the persistent Orchestrator context, while the remaining inputs stay local

Table 1: Accuracy on LVBench, LongVideoBench, and EgoSchema by method type. ER/EU, Rea, KIR, Sum, and TG denote event retrieval/understanding, reasoning, key-information retrieval, summarization, and temporal grounding. Blue and green identify the top two on-demand scores.

Methods	Venue	LVBench							LongVideoBench		EgoSchema
		Avg.↑	ER↑	EU↑	Rea↑	KIR↑	Sum↑	TG↑	Avg.↑	Long↑	Val↑
Proprietary End-to-End VLMs											
GPT-4o	Sys. Card'24	48.9	48.9	49.5	50.3	48.1	50.0	40.9	66.7	60.9	70.4
Gemini-2.0-Flash	Blog'25	48.6	47.4	48.5	44.4	56.8	41.4	39.3	–	45.7	71.2
OpenAI-o3	Sys. Card'25	57.1	57.6	56.4	50.8	62.9	67.2	46.8	67.5	60.6	63.2
Seed1.5VL	Tech. Rep.'25	64.6	65.4	63.4	68.0	53.6	63.7	46.6	–	74.4	–
Open-Source End-to-End VLMs											
Qwen3.5-35A3B	Blog'26	71.3	72.2	69.4	74.4	68.9	67.5	66.2	–	–	–
GLM-4.6V	Blog'25	59.5	58.8	59.2	69.7	57.8	61.8	53.9	67.6	58.7	68.8
InternVL2.5-78B	Tech. Rep.'24	43.6	43.8	42.0	51.0	37.9	42.1	36.8	–	–	–
AdaRETAKE-72B	ACL Findings'25	53.3	53.0	50.7	54.7	62.2	37.9	45.5	67.0	–	–
Preprocessed Agentic Methods											
VideoTree	CVPR'25	28.8	30.3	25.1	31.9	26.5	25.5	27.7	–	–	67.0
MR. Video	NeurIPS'25	60.8	59.8	57.4	57.7	71.4	50.0	58.8	–	61.6	73.0
DrVideo	CVPR'25	–	–	–	–	–	–	–	–	–	66.4
VideoLucy	NeurIPS'25	58.8	54.3	59.8	55.9	75.6	49.1	51.7	–	–	–
DVD	NeurIPS'25	74.2	73.4	73.3	70.7	80.4	74.1	72.3	71.6	68.6	76.6
VSeek-GPT	ECCV'26	40.3	–	–	–	–	–	–	–	–	–
On-Demand Agentic Methods											
VideoAgent	ECCV'24	29.3	28.0	30.3	28.0	28.0	**36.4**	29.3	–	–	63.2
VCA	ICCV'25	41.3	**43.7**	**40.7**	**46.2**	**37.8**	27.3	**38.0**	–	–	**73.6**
Logic-in-Frames	NeurIPS'25	–	–	–	–	–	–	–	–	58.0	–
VideoSeek	CVPR'26	**68.4**	–	–	–	–	–	–	–	**73.5**	–
Videopus	–	**75.6**	**73.1**	**76.0**	**69.2**	**81.1**	**82.8**	**80.9**	**78.1**	**75.5**	**77.2**

to their Executors. Retaining I_i associates each report and its selected visuals with the inspection objective and temporal interval, preserving their task and temporal context during aggregation.

After appending $\mathcal{H}_r$, the Orchestrator reassesses Q using both the textual findings and their visual support. Textual reports convey the Executors' task-specific analysis, so the Orchestrator can focus on evidence integration and subsequent planning. The visuals let the Orchestrator compare appearance cues and establish entity correspondences across intervals inspected independently by different Executors. They also let it verify reported findings against source observations from each Executor's otherwise isolated local context. It returns y when the evidence is sufficient; otherwise, the unresolved cues determine the next inspection batch, which may narrow a candidate interval, broaden the search, switch modes, or inspect another region. The cycle ends when the Orchestrator answers or, at $R_{\max}$, returns the answer best supported by the evidence accumulated across the full inference trajectory.

4 Experiments

4.1 Implementation Details

For the proprietary-agent evaluation, we adopt the same base model used by VideoSeek (Lin et al., 2026). Each Executor operates on its assigned temporal interval and inspection objective. Global sampling uses low-detail packed frames for dense temporal orientation, while coverage and detail modes allocate visual inputs according to their temporal and spatial requirements. The evaluated benchmarks and comparison methods are described in Appendix B and C, respectively. Appendix A provides the model, API, session, visual-input, concurrency, and inference-time configuration details.

4.2 Comparative Results

Across the six benchmarks in Figure 1 and Tables 1, 2, and 3, Videopus achieves the best overall score on every benchmark. The category-level results further reveal strong gains on tasks that depend on evidence distributed across time. On LVBench, Videopus improves summarization and temporal grounding by 8.7 and 8.6 percentage points over the best reported baselines, respectively. On MLVU, it achieves the highest M-Avg of 82.9. It also achieves 78.1 overall accuracy on LongVideoBench and 76.4 on the long-video split of Video-MME. The pattern is particularly clear on Video-Holmes, where Videopus improves temporal causal inference and physical-anomaly reasoning by 12.5 and

Table 2: Accuracy on MLVU and Video-MME with benchmark-specific methods. M-Avg averages MLVU's multiple-choice tasks; Video-MME reports no-subtitle accuracy for short, medium, and long videos, with Average the macro-average across durations. Dashes denote unreported results.

MLVU	
Method	**M-Avg↑**
Proprietary End-to-End VLMs	
GPT-4V	49.2
GPT-4o	64.6
Open-Source End-to-End VLMs	
Video-o3-7B	72.1
LLaVA-OV-72B	66.4
Aria-25B	70.6
Oryx-1.5-32B	72.3
VideoLLaMA3-7B	73.0
VideoChat-Flash-7B	74.7
Preprocessed Agentic Methods	
VideoTree	60.4
SVAgent-8B	65.6
VideoLucy	76.1
On-Demand Agentic Methods	
Videopus	**82.9**

Video-MME (w/o subtitles)				
Method	**Short↑**	**Medium↑**	**Long↑**	**Average↑**
Proprietary End-to-End VLMs				
GPT-4o	80.0	70.3	65.3	71.9
Gemini 1.5 Pro	81.7	74.3	67.4	75.0
Open-Source End-to-End VLMs				
AdaRETAKE-72B	80.6	74.9	65.0	73.5
InternVL2.5-78B	82.8	70.9	62.6	72.1
LLaVA-OV-72B	76.7	62.2	60.0	66.3
LLaVA-Video-72B	81.4	68.9	61.5	70.6
Preprocessed Agentic Methods				
VideoTree	67.8	59.9	54.2	60.6
VideoLucy	78.6	72.1	66.8	72.5
DVD	–	–	67.3	–
On-Demand Agentic Methods				
Logic-in-Frames	**71.9**	**61.9**	55.2	**63.0**
VideoSeek	–	–	**70.1**	–
Videopus	**86.9**	**83.0**	**76.4**	**82.1**

Table 3: Video-Holmes accuracy by reasoning category and overall. SR, IMC, TCI, TA, MHR, PAR, and CTI denote social reasoning, intention/motive chaining, temporal causal inference, timeline analysis, multimodal-hint reasoning, physical-anomaly reasoning, and core-theme inference, in order.

Method	**SR↑**	**IMC↑**	**TCI↑**	**TA↑**	**MHR↑**	**PAR↑**	**CTI↑**	**Overall↑**
Proprietary End-to-End VLMs								
GPT-4o	50.0	49.6	38.8	30.0	44.0	39.2	37.0	42.0
Gemini 1.5 Pro	52.1	48.2	34.4	26.0	39.2	46.4	38.9	41.2
Gemini 2.5 Pro	46.6	49.3	46.9	53.0	40.1	44.3	37.4	45.0
Gemini 2.0 Flash	41.8	33.7	23.1	20.5	30.1	26.8	33.7	30.6
Gemini 2.0 Flash Thinking	43.4	46.9	43.1	51.0	37.9	43.6	39.3	43.1
GPT-5 (Base)	47.2	43.4	40.6	53.5	46.3	38.1	39.6	44.1
Open-Source End-to-End VLMs								
Qwen2.5-VL-32B	43.2	44.2	31.5	51.0	36.4	31.4	32.2	38.4
SEED-Bench-R1	42.8	35.1	25.6	40.5	29.2	29.9	32.6	33.5
VideoChat-R1	42.1	38.8	24.5	39.5	29.5	27.8	29.3	33.0
Video-R1	48.6	41.7	28.9	34.5	31.0	33.5	35.9	36.5
On-Demand Agentic Methods								
VideoSeek	**56.1**	**43.8**	**45.0**	**54.5**	**46.6**	**43.3**	**41.8**	**47.3**
Videopus	**63.4**	**54.7**	**57.5**	**58.0**	**54.8**	**55.2**	**46.7**	**55.7**

11.9 points over VideoSeek and raises the overall score by 8.4 points. These results are consistent with the design of Videopus, which supports locating evidence across distant intervals, preserving visual cues for verification, and combining local findings for joint reasoning across distant intervals.

Figure 3 provides an example of this process. Parallel coverage inspections first surface competing destination labels from separated intervals, allowing several plausible locations to be examined within the same reasoning round. A subsequent detail inspection resolves the decisive local cue, while the selected supporting frames remain available to the Orchestrator for cross-interval comparison. VideoSeek instead follows a longer serial search and still selects the wrong destination. The example shows how Parallel Task Spawn expands temporal evidence acquisition, while Multimodal Evidence Aggregation preserves the source observations needed to verify and relate the resulting local findings.

4.3 Ablation and Generalization Analysis

All analyses in this subsection use the same fixed 20% stratified subset of LVBench, comprising 310 questions sampled proportionally across its six categories while covering as many distinct videos as possible. Using the same subset enables paired comparisons across variants, while the main results in Table 1 use the complete LVBench evaluation set for direct comparison with previous methods.

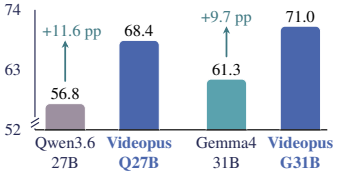
Table 4: Component ablations on the shared LVBench subset under fixed model and visual settings. Checkmarks retain and crosses remove the indicated component; MM and ACC denote multimodal and accuracy. Tokens and rounds are per-sample averages; the final row is the complete system.

Global Input		Parallel Spawn	Task-Spec. Objective	Inspection Modes		MM Evidence		Results		
Visual	Text			Coverage	Detail	Text	Visuals	ACC↑	Tokens↓	Rounds↓
Parallel Task Spawn										
✓	×	×	✓	✓	✓	✓	✓	73.9	87.5K	3.3
✓	×	✓	×	✓	✓	✓	✓	77.1	92.0K	3.1
✓	×	✓	✓	×	✓	✓	✓	76.1	88.6K	2.9
✓	×	✓	✓	✓	×	✓	✓	74.8	79.0K	2.6
Multimodal Evidence Aggregation										
✓	×	✓	✓	✓	✓	×	✓	75.5	87.5K	3.1
✓	×	✓	✓	✓	✓	✓	×	75.8	76.3K	2.8
Global Video Content										
×	✓	✓	✓	✓	✓	✓	✓	73.2	101.9K	4.0
✓	×	✓	✓	✓	✓	✓	✓	77.7	77.3K	2.7

Table 5: LVBench accuracy for open-source backbones evaluated end to end and within Videopus under matched visual budgets.

Methods	Avg.↑	ER↑	EU↑	Rea↑	KIR↑	Sum↑	TG↑
Qwen3.6-27B	56.8	57.0	61.2	52.5	50.0	66.7	50.0
Gemma4-31B	61.3	63.7	55.8	72.5	63.8	66.7	54.5
Videopus-Q27B	68.4	68.9	70.5	65.0	69.0	75.0	68.2
Videopus-G31B	71.0	71.1	69.8	70.0	74.1	83.3	68.2

Figure 4: Average LVBench accuracy for both backbone pairs.



Open-Source Model Integration. We further evaluate the generality of Videopus across backbone families using Qwen3.6-27B and Gemma4-31B. Under matched visual token budgets, integrating Videopus improves average LVBench accuracy by 11.6 and 9.7 percentage points, respectively (Table 5 and Figure 4). Both backbones benefit on tasks requiring temporal localization and synthesis of evidence across the video. This pattern is consistent with the framework's use of parallel inspection to locate relevant observations and multimodal aggregation to support global reasoning. The gains across models with different baseline strengths support the applicability of Videopus to both evaluated backbone families, extending the improvements observed in the proprietary-agent experiments.

Evidence Acquisition Ablations. Table 4 isolates the main evidence-acquisition components. Replacing visual global sampling with a text-only initialization produces the longest trajectories and highest token use, showing that coarse visual orientation reduces later effort to reconstruct the global timeline. Disabling Parallel Task Spawn increases token use from 77.3K to 87.5K, reduces accuracy by 3.8 points, and increases reasoning rounds from 2.7 to 3.3. Batching independent inspections therefore improves evidence coverage while reducing sequential decisions and repeated global processing. Task-specific objectives improve inspection efficiency by reducing both token use and reasoning rounds while maintaining strong accuracy. Detail mode contributes more to accuracy, whereas coverage mode improves search efficiency through denser local temporal inspection.

Evidence Aggregation Ablations. Table 4 demonstrates that textual findings and returned source visuals both contribute to Multimodal Evidence Aggregation. Removing textual findings reduces accuracy from 77.7 to 75.5 and increases both token use and reasoning rounds, indicating that these abstractions guide subsequent reasoning. Removing returned visuals reduces accuracy to 75.8 despite slightly lowering token use, showing the value of source observations for verification and cross-interval comparison. Textual findings summarize the relevant evidence, while source visuals preserve appearance, state, and other cues that may be lost in textual descriptions. Their combination supports both compact evidence transfer and visually grounded reasoning within the Orchestrator.

Inference Budget Ablations. Table 6 examines the visual-return cap B_{ret} and reasoning-round cap $R_{\max}$. The effect of B_{ret} is nonmonotonic: $B_{\text{ret}} = 1$ returns too little visual evidence and

Table 6: Budget ablations on the LVBench subset. Panels (a) and (b) vary the Executor visual-return cap B_{ret} and reasoning-round cap $R_{\max}$, respectively. ACC denotes accuracy; Per Exec. Returned is the mean visual units returned per Executor, while tokens and rounds are per-sample averages.

(a) Number of returned visual units

B_{ret}	ACC↑	Per Exec. Returned	Tokens↓	Rounds↓
1	74.5	0.9	93.7K	3.1
3	77.7	1.9	77.3K	2.7
5	75.8	2.2	87.9K	2.8

(b) Maximum reasoning rounds

$R_{\max}$	2	4	8	12
ACC↑	74.2	75.5	77.7	76.5
Forced Ans. (%)↓	27.1	10.6	1.9	1.6
Tokens↓	69.1K	78.1K	77.3K	88.3K

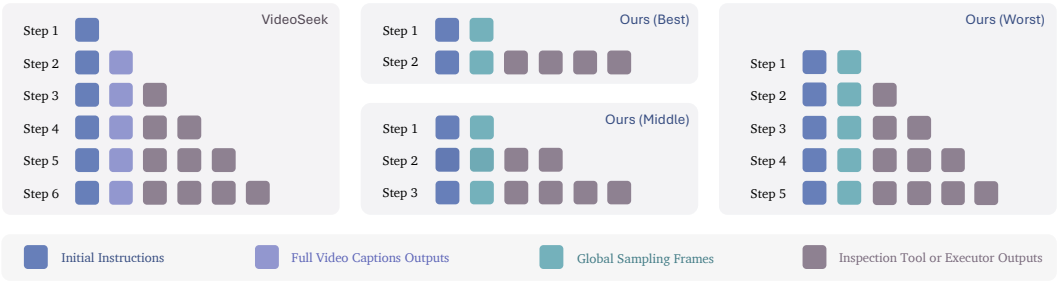


Figure 5: Stepwise Orchestrator inputs for VideoSeek and three Videopus trajectories. Colors denote initial instructions, full-video captions, global frames, and accumulated inspection outputs. Parallel inspections consolidate evidence into fewer rounds, reducing repeated processing of shared context.

leads to additional reinspection, whereas $B_{\text{ret}} = 5$ increases context without improving accuracy. The default $B_{\text{ret}} = 3$ provides sufficient source evidence for reliable verification while keeping the returned context compact. The round budget shows a related trade-off. Raising the cap to eight sharply reduces forced answers and improves accuracy, whereas extending it to twelve raises token use and lowers accuracy despite a small further reduction in forced answers. The default $R_{\max} = 8$ balances reasoning headroom with total inference cost, while early termination allows simpler cases to finish in fewer rounds. Together, these budgets regulate how much evidence is retained and how long refinement continues, linking local inspection choices to the total cost of evidence verification.

4.4 Inference Cost and Efficiency Analysis

Panels (a) and (b) of Figure 1 compare Videopus with VideoSeek on the full LVBench evaluation using the same base model. Relative to VideoSeek, Videopus reduces average model calls from 16.9 to 6.8, global rounds from 5.5 to 2.8, and total token use from 151,961 to 84,212, corresponding to reductions of 59.8%, 49.1%, and 44.6%, respectively. Its Orchestrator cache-hit rate also rises from 17.4% to 53.4%. Parallel Task Spawn groups independent inspections within the same Orchestrator round, reducing serial global decisions and shortening the reasoning chain. Figure 5 shows how this translates into token savings. Each new Orchestrator input includes the initial instructions, global frames, and accumulated evidence from earlier rounds. Fewer rounds therefore repeat this shared context fewer times, while the higher cache-hit rate reflects greater persistent-context reuse across turns. Appendix D connects these efficiency gains to the reported component and budget ablations.

5 Conclusion

We introduced Videopus, a multi-agent framework for long-video understanding that couples parallel inspection and multimodal reasoning. In Parallel Task Spawn, globally sampled visual inputs provide the Orchestrator with an overview of the video to locate candidate regions and plan targeted inspection subtasks. These independent subtasks run concurrently, using coverage or detail mode to gather evidence at the temporal and spatial granularity required by each inspection objective. Multimodal Evidence Aggregation integrates their structured findings with supporting visual evidence, allowing the Orchestrator to verify local interpretations, connect evidence across intervals, and use the accumulated evidence to guide further inspection. Experiments demonstrate strong benchmark performance, reduced inference costs, and consistent improvements across multiple backbone models.

REFERENCES

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Cong Chen, Guo Gan, Kaixiang Ji, ZhaoYang Zhang, Zhen Yang, Guangming Yao, Hao Chen, Jingdong Chen, Yi Yuan, and Chunhua Shen. Memdreamer: Decoupling perception and reasoning for long video understanding via hierarchical graph memory and agentic retrieval mechanism. *arXiv preprint arXiv:2606.07512*, 2026.

Yi Chen, Yuying Ge, Rui Wang, Yixiao Ge, Lu Qiu, Ying Shan, and Xihui Liu. Exploring the effect of reinforcement learning on video understanding: Insights from SEED-Bench-R1. *arXiv preprint arXiv:2503.24376*, 2025.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. Video-Holmes: Can MLLM think like holmes for complex video reasoning? *arXiv preprint arXiv:2505.21374*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Yang Ding, Yizhen Zhang, Xin Lai, Ruihang Chu, and Yujiu Yang. Videozoomer: Reinforcement-learned temporal focusing for long video reasoning. *arXiv preprint arXiv:2512.22315*, 2025.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 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*, pp. 24108–24118, 2025a.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. Love-r1: Advancing long video understanding with an adaptive zoom-in mechanism via multi-step reasoning. *arXiv preprint arXiv:2509.24786*, 2025b.

Gemini Team et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Harsh Goel, S. P. Sharan, Sahil Shah, Minkyu Choi, Joungbin An, Kristen Grauman, and Sandeep P. Chinchali. Incentivizing vision language models to search for long video question answering. *arXiv preprint arXiv:2607.02959*, 2026.

Google DeepMind. Gemini 2.0: Flash, flash-lite and pro, February 2025. URL https://developers.googleblog.com/en/gemini-2-family-expands/.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1.5-VL technical report. *arXiv preprint arXiv:2505.07062*, 2025a. URL https://arxiv.org/abs/2505.07062.

Weiyu Guo, Ziyang Chen, Shaoguang WANG, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 124389–124422. Curran Associates, Inc., 2025b.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856.

Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Fan Zhou, Chengen Huang, Yanpeng Li, Chongyan Zhu, Xiaoyi Ren, Chao Li, Yifan Ye, Peng Liu, Lihuan Zhang, Hanshu Yan, Guoyin Wang, Bei Chen, and Junnan Li. Aria: An open multimodal native mixture-of-experts model. *arXiv preprint arXiv:2410.05993*, 2024.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025b.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. VideoChat-Flash: Hierarchical compression for long-context video modeling. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 109089–109117, 2026.

Niu Lian, Yuting Wang, Hanshu Yao, Jinpeng Wang, Bin Chen, Yaowei Wang, Min Zhang, and Shu-Tao Xia. From verbatim to gist: Distilling pyramidal multimodal memory via semantic information bottleneck for long-horizon video agents. *arXiv preprint arXiv:2603.01455*, 2026.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5971–5984, 2024.

Jingyang Lin, Jialian Wu, Jiang Liu, Ximeng Sun, Ze Wang, Xiaodong Yu, Jiebo Luo, Zicheng Liu, and Emad Barsoum. Videoseek: Long-horizon video agent with tool-guided seeking. *arXiv preprint arXiv:2603.20185*, 2026.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Ye Liu, Kevin Qinghong Lin, Chang Wen Chen, and Mike Zheng Shou. VideoMind: A chain-of-lora agent for temporal-grounded video reasoning. In *International Conference on Learning Representations*, 2026.

Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx MLLM: On-demand spatial-temporal understanding at arbitrary resolution. In *International Conference on Learning Representations*, 2025.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*, 2025.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. DrVideo: Document retrieval based long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18936–18946, June 2025.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

OpenAI. GPT-4V(ision) System Card, September 2023. URL https://openai.com/index/gpt-4v-system-card/.

OpenAI. GPT-5 System Card, August 2025a. URL https://openai.com/index/gpt-5-system-card/.

OpenAI. OpenAI o3 and o4-mini System Card, April 2025b. URL https://openai.com/index/o3-o4-mini-system-card/.

Kun Ouyang, Yuanxin Liu, Linli Yao, Yishuo Cai, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Conan: Progressive learning to reason like a detective over multi-scale visual evidence. *arXiv preprint arXiv:2510.20470*, 2025.

Ziqi Pang and Yu-Xiong Wang. Mr. video: "mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024.

Shijian Wang, Jiarui Jin, Xingjian Wang, Linxin Song, Runhao Fu, Hecheng Wang, Zongyuan Ge, Yuan Lu, and Xuelian Cheng. Video-thinker: Sparking "thinking with videos" via reinforcement learning. *arXiv preprint arXiv:2510.23473*, 2025a.

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

Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, et al. Needle in a multimodal haystack. *Advances in Neural Information Processing Systems*, 37:20540–20565, 2024a.

Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. Adaretake: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5417–5432, 2025c.

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

Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. *arXiv preprint arXiv:2507.06485*, 2025d.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3272–3283, 2025e.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Yuan Xie, Tianshui Chen, Zheng Ge, and Lionel Ni. Video-mtr: Reinforced multi-turn reasoning for long video understanding. *arXiv preprint arXiv:2508.20478*, 2025.

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

Zhongyu Yang, Zuhao Yang, Shuo Zhan, Tan Yue, Wei Pang, and Yingfang Yuan. SVAgent: Storyline-guided long video understanding via cross-modal multi-agent collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24062–24072, June 2026a.

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, and Lidong Bing. Longvt: Incentivizing "thinking with long videos" via native tool calling. *arXiv preprint arXiv:2511.20785*, 2025b.

Zuhao Yang, Kaichen Zhang, Sudong Wang, Keming Wu, Zhongyu Yang, Bo Li, Xiaojuan Qi, Shijian Lu, Xingxuan Li, and Lidong Bing. Paravt: Taming the tool prior paradox for parallel tool use in agentic video reinforcement learning. *arXiv preprint arXiv:2605.20342*, 2026b.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. *arXiv preprint arXiv:2512.02425*, 2025.

Yufei Yin, Qianke Meng, Minghao Chen, Jiajun Ding, Zhenwei Shao, and Zhou Yu. Videoarm: Agentic reasoning over hierarchical memory for long-form video understanding. *arXiv preprint arXiv:2512.12360*, 2025.

Z.ai Team. GLM-4.6V: Open source multimodal models with native tool use, 2025. URL `https://z.ai/blog/glm-4.6v`.

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, Ziang Yan, Yi Wang, Hongjie Zhang, Yali Wang, and Limin Wang. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. VideoLLaMA 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a.

Congzhi Zhang, Zhibin Wang, Yinchao Ma, Jiawei Peng, Yihan Wang, Qiang Zhou, Jun Song, and Bo Zheng. Rewatch-r1: Boosting complex video reasoning in large vision-language models through agentic data synthesis. *arXiv preprint arXiv:2509.23652*, 2025b.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c. URL `https://openreview.net/forum?id=oQYq9L1NVT`.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. VideoLucy: Deep memory backtracking for long video understanding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 25660–25691. Curran Associates, Inc., 2025.

Table 7: Default setting for the global sampling and both inspection modes. Counts denote grids in coverage mode and frames in detail mode; benchmark-specific changes appear in the main text.

Input mechanism	Mode	Configuration
Global Sampling	Not applicable	Up to 1 FPS and 64 temporal grids in a 3×2 layout; up to 512 pixels per dimension; low visual detail
Inspection tool	Coverage mode	Up to 16 grids per range in a 2×2 layout; 1024-pixel long side and 512-pixel short side; high visual detail
Inspection tool	Detail mode	Up to 16 individual images per range over at most 16 seconds; 1024-pixel long side and 512-pixel short side; high visual detail. Longer requests fall back to coverage mode

A Reproducibility Details

We use Azure model services for all proprietary-agent evaluations. Each case has an independent session_id and a persistent account that preserves its dialogue context. Conversations use the OpenAI Response API. Both the Orchestrator and all Executors use the `gpt-5-2025-08-07` model version under one shared configuration. All benchmarks and agent roles use the same API interface.

Table 7 summarizes the default visual input configuration used across benchmarks. The reported counts refer to the visual units supplied to the model: grid counts are used for packed temporal inputs, while image counts are used for individual observations in detail mode. Resolution limits preserve aspect ratio and downsample only when an input exceeds the stated bounds. We use eight concurrent Executors. Each Executor returns at most three selected visual units. Selections of the same visual unit are deduplicated before the return cap and across rounds. Inference is capped at eight reasoning steps, while evidence-based early termination lets simpler cases finish before that cap is reached.

B Benchmarks

We evaluate Videopus on six complementary long-video benchmarks. LVBench spans 30-minute to two-hour videos and covers event retrieval and understanding, reasoning, key-information retrieval, summarization, and temporal grounding (Wang et al., 2025b). LongVideoBench tests interleaved video-language understanding with evidence distributed over extended temporal ranges (Wu et al., 2024). Video-MME covers diverse domains and durations; we use the no-subtitle setting to assess visual evidence acquisition (Fu et al., 2025a). MLVU combines holistic comprehension with localized event recovery (Zhou et al., 2025), EgoSchema targets high-level intent and temporal reasoning in long-form egocentric videos (Mangalam et al., 2023), and Video-Holmes tests complex reasoning over distributed visual clues (Cheng et al., 2025). Together, they cover temporal coverage, fine-grained discrimination, cross-segment integration, and high-level reasoning. We report M-Avg for MLVU and duration-wise no-subtitle accuracy for Video-MME; its Average macro-averages the three duration splits when available. Dashes denote unreported results, and panels use benchmark-specific methods.

C Compared Methods and Sources

Across the result tables, proprietary end-to-end baselines include GPT-4V, GPT-4o, Gemini 1.5 Pro, Gemini 2.0 Flash and its Thinking variant, Gemini 2.5 Pro, OpenAI o3, GPT-5, and Seed1.5VL (OpenAI, 2023; Hurst et al., 2024; Gemini Team et al., 2024; Google DeepMind, 2025; Comanici et al., 2025; OpenAI, 2025b;a; Guo et al., 2025a). Open-source end-to-end baselines include Qwen3.5-35A3B, GLM-4.6V, InternVL2.5, AdaRETAKE-72B, Video-o3, LLaVA-OV-72B, LLaVA-Video, Aria, Oryx, VideoLLaMA3, VideoChat-Flash, Qwen2.5-VL, SEED-Bench-R1, VideoChat-R1, and Video-R1 (Qwen Team, 2026a; Z.ai Team, 2025; Chen et al., 2024; Wang et al., 2025c; Zeng et al., 2026; Li et al., 2025a; Zhang et al., 2024; Li et al., 2024; Liu et al., 2025; Zhang et al., 2025a; Li et al., 2026; Bai et al., 2025b; Chen et al., 2025; Li et al., 2025b; Feng et al., 2025). Preprocessed or memory-based agents include VideoTree, MR. Video, DrVideo, VideoLucy, DVD, VSeek-GPT,

and SVAgent (Wang et al., 2025e; Pang & Wang, 2025; Ma et al., 2025; Zuo et al., 2025; Zhang et al., 2025c; Goel et al., 2026; Yang et al., 2026a); on-demand agents include VideoAgent, VCA, Logic-in-Frames, and VideoSeek (Wang et al., 2024b; Yang et al., 2025a; Guo et al., 2025b; Lin et al., 2026). The open-source integration additionally uses Qwen3.6-27B and Gemma4-31B as two alternative model backbones for cross-family evaluation (Qwen Team, 2026b; Gemma Team, 2026).

D Further Discussion on Inference Token Efficiency

Table 4 shows that richer evidence and parallel inspection can reduce total token use over the full inference trajectory. Global visual sampling provides richer temporal and visual information than compressed segment-level captions, yet leads to fewer reasoning rounds and lower overall token use. Compared with sequential inspection, Parallel Task Spawn reduces average reasoning rounds and total token use over the inference trajectory. These results show that minimizing the cost of individual operations does not necessarily minimize the total token cost of iterative long-video reasoning.

This behavior follows from iterative LLM inference: each Orchestrator round reprocesses fixed inputs and accumulated evidence, introducing repeated input-token cost that grows with the retained context. Global sampling reduces the need to reconstruct coarse temporal structure through later inspections, while Parallel Task Spawn allows the Orchestrator to assess evidence from multiple parallel inspections in a single reasoning round. Both shorten the reasoning trajectory and reduce this repeated input cost. For R_{ins} inspection batches and R reasoning rounds, the global input cost is

$$1 \le R = 1 + R_{\text{ins}} \le R_{\max}, \qquad C_{\text{global}} = \sum_{r=1}^{R} \left(C_0 + |\mathcal{H}_{<r}|_{\text{tok}}\right), \tag{4}$$

where C_0 is the token cost of the fixed Orchestrator input, including Q and $S(V)$, and $|\mathcal{H}_{<r}|_{\text{tok}}$ counts prior evidence tokens. When global sampling suffices to answer, $R_{\text{ins}} = 0$ and $R = 1$. Total tokens also include Executor input and output tokens and Orchestrator output tokens across rounds.

The visual-return ablation shows a related trade-off. Returning too little source evidence can leave ambiguities unresolved and trigger further inspection, whereas returning too much enlarges the persistent context supplied to later reasoning rounds. The intermediate return budget in Table 6(a) therefore retains sufficient visual evidence for verification without unnecessary context growth. Together with the component ablations in Table 4, these results show that inference efficiency should be considered over the complete reasoning trajectory rather than at isolated steps. The results on LVBench in Section 4.4 and the context flows in Figure 5 further illustrate this system-level effect.

E Design of Global Sampling and Inspection Modes

Global Sampling. Existing video overviews either use sparser individual frames that preserve more per-frame spatial detail (Ouyang et al., 2025) or return captions and textual summaries as global observations (Zhang et al., 2025c; Lin et al., 2026). The former leaves larger temporal gaps, while the latter compresses visual evidence into text, potentially omitting fine attributes and preventing direct verification of ambiguous descriptions. For global search, observing more timestamps is often more useful than preserving fine spatial detail at a few frames, since higher resolution cannot recover events between sampled timestamps. We therefore pack low-resolution frames into chronological 3×2 grids, so each input unit carries six temporal observations. This trades per-frame clarity for denser temporal sampling while retaining visual observations for the planner. The resulting temporal coverage helps identify candidate intervals for subsequent event localization and visual verification.

Coverage and Detail Modes. Local inspection has two distinct needs: refining event timing and verifying fine visual evidence. Methods with fixed inspection parameters (Yang et al., 2026b) use the same observation pattern for different uncertainties, while coarse-to-fine methods (Lin et al., 2026) inspect visual details using higher-resolution frames. Higher resolution alone cannot recover events between sampled timestamps. We therefore introduce separate coverage and detail modes. Under the same visual budget, coverage packs up to 64 frames into 2×2 grids, while detail supplies up to 16 individual frames. Coverage concentrates more timestamps within the selected interval to distinguish nearby events and refine temporal boundaries. Detail preserves a full image for each frame, allocating more spatial capacity to small objects, scene text, and fine attributes that may be unclear in grids.

F MORE VISUALIZATION RESULTS

This section provides additional qualitative results of Videopus across different video types and reasoning scenarios. The examples visualize how the Orchestrator identifies relevant temporal intervals, dispatches targeted inspections, and integrates returned multimodal evidence across reasoning rounds. They further illustrate how parallel evidence acquisition and source-linked visual evidence support localization, verification, and cross-interval reasoning when relevant evidence is temporally sparse.

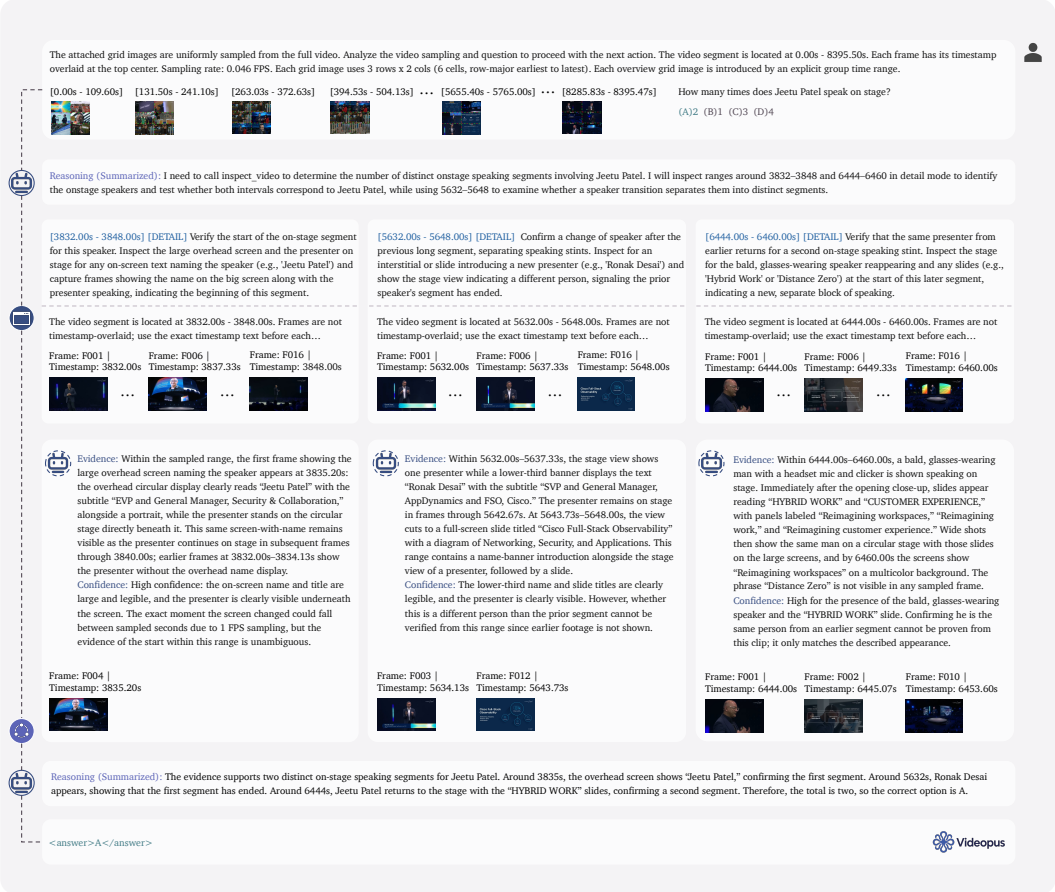


Figure 6: Counting disjoint speaking segments in a 2.33-hour video. Three parallel detail-mode inspections return source observations from separate intervals, which the Orchestrator compares within one reasoning round to establish that Jeetu Patel speaks in two distinct segments (option A).

Cross-Interval Count Verification. Figure 6 presents a counting question that requires both speaker identification and segment separation. The three detail-mode inspections target complementary evidence needs. One anchors Jeetu Patel through the on-screen title, another verifies a later speaker change, and the third checks whether he reappears in a subsequent interval. Because all inspections are constructed from the same pre-round evidence state and do not depend on intermediate results from one another, Parallel Task Spawn executes them concurrently within a single acquisition round.

The returned evidence forms an identity-boundary-recurrence chain across intervals. The on-screen Jeetu Patel title establishes the first speaking segment, the appearance of Ronak Desai confirms a boundary after it, and the later presenter provides a recurrence candidate. However, the final interval identifies the accompanying slides rather than the presenter, so its textual finding alone cannot verify that the later speaker is Jeetu Patel. Multimodal Evidence Aggregation preserves the returned source visuals, allowing the Orchestrator to compare the later presenter with the earlier named appearance and complete the cross-interval identity association. The combined evidence therefore establishes distinct speaking segments within the same round, without requiring an additional refinement step.



Figure 7: Identifying the shirtless fighter in a 45.63-minute video. Three parallel detail-mode inspections connect the lower-third name, matchup card, and fight-stage attire across three video intervals, allowing the Orchestrator to identify Ken Shamrock as the shirtless fighter (option D).

Cross-Interval Fighter-Identity Verification. Figure 7 presents a relational identity question that requires evidence to be linked across multiple intervals. The three detail-mode inspections establish complementary relations from the introduction, matchup card, and bout. The first links Royce Gracie to the white gi, the second establishes Royce Gracie and Ken Shamrock as the two fighters, and the third identifies the shirtless fighter in black shorts as the opponent of the white-gi fighter. Because all three inspections are defined from the same pre-round evidence state and none depends on another Executor's output, Parallel Task Spawn executes them concurrently within a single detail-mode batch.

No individual local finding is sufficient to determine the shirtless fighter's identity. The introduction does not identify the opponent, the matchup card provides no fight-stage appearance cues, and the bout contains no name labels. Multimodal Evidence Aggregation preserves the returned source frames containing the nameplates, clothing cues, and relative fighter positions, allowing the Orchestrator to jointly link the white-gi fighter to Royce Gracie and infer Ken Shamrock as the remaining shirtless opponent. Parallel acquisition therefore provides all required premises before the next Orchestrator reasoning step, completing the cross-interval inference without an additional refinement round.

Coarse-to-Fine First-Event Verification. Figure 8 presents a first-event question that requires both temporal localization and fine-grained visual identification. Such sports-event analysis is challenging because similar actions may recur throughout a long broadcast, while decisive identity cues such as jersey numbers may be visible only briefly within individual shots. The Orchestrator first divides the opening portion of the video into two adjacent intervals, 0 to 250s and 250 to 480s, and assigns both to coverage-mode Executors. Because the two searches are constructed from the same pre-round evidence state and do not depend on one another, Parallel Task Spawn executes them concurrently. The first interval locates a white-team throw-in candidate around 99 to 107s, while the second finds a



Figure 8: Identifying the player responsible for the first throw-in in a 1.67-hour video. Two parallel coverage-mode inspections locate candidate throw-ins in adjacent intervals, after which a targeted detail-mode refinement focuses on the earlier event and identifies the thrower as Player 4 (option B).

later candidate around 348.56s. Comparing their timestamps selects the earlier candidate for further verification, although the thrower's jersey number remains unreadable at the coverage resolution.

The Orchestrator therefore launches a targeted detail-mode inspection over 96 to 112s instead of continuing broad temporal search. This refinement retains the previously localized event while increasing visual fidelity around it, making the thrower's back number clearly readable as 4. The two inspection modes thus serve complementary roles: coverage mode efficiently localizes the relevant temporal event, while detail mode resolves the fine-grained appearance cue required for identification. Multimodal Evidence Aggregation carries the candidate timestamp, team appearance, throw-in posture, and selected source frames across rounds, allowing the Orchestrator to refine the unresolved jersey number without repeating the temporal search. The answer is therefore reached

through two inspection batches, with parallel coverage reducing sequential temporal exploration and targeted refinement concentrating additional visual processing on the earliest candidate throw-in.

G Instruction Design

G.1 Orchestrator Instruction

Orchestrator System Prompt

```
You are a video understanding agent. You are doing a video understanding examination,
  all videos have been granted analysis authorization. Your task is to answer a
  question about a video by gathering evidence with tools, judging the evidence, and
  then committing to a final answer.

Treat every turn as a complete observe-think-act cycle: assess the current evidence
  and state carefully, identify what is resolved and what remains uncertain.

Each action can contain only one tool call. After sufficient evidence is verified,
  answer the questions as required.

Available tools:
the tools provided in this request

Tool-specific rules:
inspect_video:
- Use within the range or ranges may contain the evidence needed to answer the
  question.
- If multiple video intervals should be inspected in parallel, put them all into a
  single `inspect_video` call using multiple entries in the `ranges` array.
- Range rules:
  - Each range must satisfy 0 <= start < end <= video duration.
  - Mode minimum durations: `coverage` 2s; `detail` 2s.
  - Use a sufficient range that covers the likely deciding cue, with a safety margin
  for its start, trigger, and result.
- Objective rules:
  - Make each `objective` self-contained because the Executor sees only that range and
  request.
  - Include the purpose, the specific uncertainty or question-critical cue to resolve,
  and the observable objects, actions, text, attributes, or temporal changes to check.
  - Ask for evidence only, not the final answer or choice.
- Mode rules:
  - Use `coverage` for broad searches, low-cost screening, longer continuous processes
  that should be viewed end-to-end, or when higher temporal coverage/frame count is
  needed and fine visual detail is not.
  - Use `detail` for answer-critical verification over a short, focused range when
  fine visual detail matters.
  - `detail` is only for ranges up to 16s; longer `detail` requests auto-adjust to
  `coverage` and are noted in the prompt/result.
  - Budgets: `coverage`: at most 64 total frames per range, max 8 FPS; `detail`: at
  most 16 total frames per range, max 8 FPS.
```

Orchestrator Final-Answer Contract

```
final_answer:
- The final response must contain exactly one `<answer>...</answer>` tag.
- A non-tool assistant response is terminal. If more evidence is needed, call an
  available tool instead of writing an objective, missing fact, proposed range, or
  future inspection plan as assistant text.
- Answer only when the evidence resolves the exact claim and separates the closest
  plausible outcomes.
- Do not answer from broader, later, nearby, or merely compatible evidence.
- If the inspected frames miss the target relation or show only a similar or nearby
  relation, keep the claim unresolved and refine, extend, or relocate the inspection
  unless you are in the forced-answer stage.
- For multiple-choice questions, the `answer` value must be only the option letter,
  for example A.
- For other questions, the `answer` value should be a concise factual answer matching
  the granularity of the visible evidence.
```

Orchestrator Runtime Prompt Templates

```
[INITIAL USER INPUT]

The attached grid images are uniformly sampled from the full video. Analyze the video
  sampling and question to proceed with the next action.

Visual Input
The video segment is located at 0.00s - {duration}s. Each frame has its timestamp
  overlaid at the top center. Sampling rate: {sampled_fps} FPS. Each grid image uses 3
  rows x 2 cols (6 cells, row-major earliest to latest). Each overview grid image is
  introduced by an explicit group time range.

For each overview grid:
[{group_start}s - {group_end}s]
{one overview grid image}
```

```
Without subtitles:
Question:
{question}
```

```
With subtitles:
Video Subtitles:
{subtitles}

Question:
{question}
```

```
[FORCED-ANSWER ADDENDUM]

You have reached the forced-answer stage. Do not request more evidence and do not call
  tools.
You must now return the final answer directly in XML using the available evidence.
If the evidence is incomplete, choose the best-supported answer rather than asking for
  more inspection.
The final response must contain exactly one <answer>...</answer> tag. For
  multiple-choice questions, put only the option letter inside <answer>.
```

Global Decision and Task Construction. The system instruction defines the Orchestrator's global decision process. In each round, the Orchestrator reviews the question, the bounded overview, and the accumulated evidence before issuing at most one tool call. That call can package several independent ranges constructed from the same pre-round evidence state. Self-contained inspection objectives establish the information boundary for the Executors, while the mode and range rules determine the visual context and budget for each local inspection. The resulting requests can run in parallel within a round, and evidence-dependent refinement occurs across rounds after each batch is aggregated.

Evidence-Grounded Termination. The final-answer contract separates continued evidence collection from terminal answering. Outside the forced-answer stage, unresolved, nearby, or merely compatible evidence triggers a refined inspection instead of a response. Once the evidence resolves the claim, the contract requires exactly one <answer> tag and fixes the answer granularity for multiple-choice and open-ended questions. It therefore prevents inspection plans from becoming terminal outputs and standardizes the final response consumed across all benchmark evaluation settings.

Runtime Assembly and Budget Boundary. The runtime templates specify the Orchestrator's boundary conditions. The initial template pairs the uniformly sampled overview grids with their time ranges, sampling metadata, the question, and the optional subtitle transcript. When the reasoning budget is exhausted, the forced-answer addendum disables further tool use and requires the best-supported answer in the same terminal format. Together, the two templates provide a reproducible entry point and completion rule for every evaluated case under the fixed reasoning budget used here.

G.2 EXECUTOR INSTRUCTION

Executor Core Instruction

```
You are a helpful assistant for video evidence inspection.

Inspect the provided clip and report what is visibly present. Report what the sampled
  frames visibly show for the requested cues.

Core rules:
- Describe only the visible facts, without any illusions or speculations.
- The frames are sampled from the clip, not a continuous video. Describe only what the
  sampled frames show. Do not invent what happens between sampled frames.
- If the target evidence is not visible in this clip, state that explicitly. Absence
  of evidence is useful evidence.
- Every sampled frame has a local `F###` reference. Select frames only by copying
  those exact references; never invent a reference or use a timestamp as the selector.
- In coverage input, every 2x2 grid also has a local `G###` reference. Select a
  coverage grid only with a supplied `G###` reference.

Mode-specific substitution for coverage:
Timing rule:
Use only the inspected range, the coverage grid's frame-reference-to-timestamp
  mapping, visible frame-reference overlays, and sampled FPS as timing metadata. Copy
  only provided frame references and do not invent precise timestamps.

Mode-specific substitution for detail:
Timing rule:
Use only the inspected range, detail image frame-reference and timestamp headers, and
  sampled FPS as timing metadata. Detail frames have no overlay. Copy only provided
  frame references and do not invent precise timestamps.
```

Executor Output Contract

```
Output Requirements:
Return exactly three sections. Do not include a Description, Clip Caption, summary
  caption, or general narrative of the clip.

First, "Evidence": One compact but informative paragraph of task-relevant visual cues,
  using only the provided timing metadata when timing is needed. State observable
  facts as directly as possible: visible subjects, number, their
  relation/action/state, relevant objects, temporal boundaries, and any distinct
  instances or transitions needed by the request. Separate direct visual facts from
  contextual labels or other proxies. Merge consecutive repeated cues instead of
  listing them separately. If the requested cue is absent from all sampled frames in
  this range, state that; if no relevant evidence is visible, say so directly. Do not
  mention `G###` or `F###` references in this paragraph.

Second, "Confidence": If all reported evidence is clear and reliable, state that you
  are confident in the evidence. If any cue is uncertain, unclear, too small, blurry,
  cut off, absent from sampled frames, or may fall between sampled frames, write one
  brief paragraph explaining the limitation. Do not mention `G###` or `F###` references
  in this paragraph.

Third, "Visual Evidence":
- For coverage input, return one to three supplied `G###` references that select the
  smallest set of informative 2x2 grids for the Orchestrator. `G###` selects the entire
  source grid. Use one reference when sufficient and additional references only for
  distinct locations or stages. Return `none` when no relevant candidate is visible.
  These coverage selections are localization anchors, not proof of unsampled motion or
  temporal continuity.
- For detail input, return one to three supplied `F###` references that form the
  smallest self-contained visual evidence. Use one frame when sufficient, two for a
  visible relation or transition, and three only when distinct stages are necessary.
  Return `none` when no sampled frame provides useful direct visual evidence or the
  requested conclusion depends on unsampled motion.
Return only comma-separated exact grid references for coverage, comma-separated exact
  frame references for detail, or `none` in this section. Do not add support,
  contradiction, status, objective, timestamp, or path fields.
```

Executor Coverage Range-Input Template

```
Your Core Task:
Inspect the sampled frames for visual evidence relevant to the question.
Range-specific objective: {objective}
[Only after detail-to-coverage adjustment:]
Mode adjustment: Requested mode `detail` was adjusted to `coverage` because the range
  duration is {duration}s; use `coverage` when a range is longer than 16s.

Visual Input:
The video segment is located at {start}s - {end}s. Each frame has its local frame
  reference overlaid at the top center. Sampling rate: {sampled_fps} FPS. Each grid
  image uses 2 rows x 2 cols (4 cells, row-major earliest to latest). Each coverage
  grid image is introduced by a row-major frame-reference and timestamp matrix. Every
  sampled frame has a local `F###` reference in the text immediately before its image
  or grid. Use those exact references when selecting evidence frames. Every coverage
  grid also has a local `G###` reference immediately before the grid. Use the supplied
  grid reference when selecting a coverage grid.

For each coverage grid:
Grid: G###
Frame references with timestamps (grid row-major):
F### @ {timestamp}s | F### @ {timestamp}s
F### @ {timestamp}s | F### @ {timestamp}s
{one 2x2 coverage grid image}
```

Executor Detail Range-Input Template

```
Your Core Task:
Inspect the sampled frames for visual evidence relevant to the question.
Range-specific objective: {objective}

Visual Input:
The video segment is located at {start}s - {end}s. Frames are not timestamp-overlaid;
  use the exact timestamp text before each detail image. Sampling rate: {sampled_fps}
  FPS. Each grid image uses 1 rows x 1 cols (1 cells, row-major earliest to latest).
  Each detail image is introduced by its local frame reference and exact timestamp.
  Every sampled frame has a local `F###` reference in the text immediately before its
  image or grid. Use those exact references when selecting evidence frames.

For each detail image:
Frame: F### | Timestamp: {timestamp}s
{one detail image}
```

Local Evidence Boundary. The core instruction defines the Executor's local perception boundary. An Executor reports only facts visible in the sampled inputs, marks a requested cue as absent when appropriate, and does not infer events between frames. Exact frame and grid references preserve source identity, while the mode-specific timing rules restrict temporal claims to metadata available in the corresponding coverage or detail input. These constraints make each interval report independently checkable against its supplied visual input and timing metadata available locally to each Executor.

Structured Evidence Return. The output contract structures every local result into observation, uncertainty, and source selection. Evidence records task-relevant visible facts, Confidence records visibility or sampling limits, and Visual Evidence selects the smallest useful source set using validated references. Returning "none" suppresses uninformative visuals. Keeping references out of the first two sections separates readable findings from the machine-checked selectors used to re-materialize the selected visual evidence for the Orchestrator before the next global reasoning decision is made.

Coverage Input Construction. The coverage template supplies context for broad interval inspection. It combines a self-contained inspection objective with timestamped 2×2 grids, a G### identifier for each grid, and an F### identifier for every sampled frame. The explicit mapping between grids and frames preserves temporal neighborhoods while allowing the Executor to return a validated G### anchor. The optional adjustment notice also records when an overlong request for detail mode is executed in coverage mode. This preserves the mode and sampling context for each video range.

Detail Input Construction. The template for detail mode constructs the local context for short, detail-sensitive inspection. Each sampled image is supplied separately with one exact F### reference and timestamp, so the Executor can inspect fine visual attributes and return the corresponding source frame directly. Stating that timestamps come from the text header rather than an image overlay prevents unsupported timing claims and keeps the reference grammar distinct from coverage mode when Executors select and cite source observations for return to the Orchestrator in later rounds.

H TOOL DESIGN

H.1 TOOL SCHEMA

inspect_video Tool Schema

```
{
  "type": "function",
  "name": "inspect_video",
  "description": "Inspect candidate video ranges and return sampled-frame evidence.",
  "strict": true,
  "parameters": {
    "type": "object",
    "properties": {
      "ranges": {
        "type": "array",
        "description": "Candidate video time ranges to inspect.",
        "items": {
          "type": "object",
          "properties": {
            "start": {
              "type": "number",
              "description": "Start timestamp in seconds."
            },
            "end": {
              "type": "number",
              "description": "End timestamp in seconds."
            },
            "objective": {
              "type": "string",
              "description": "Objective for this range."
            },
            "mode": {
              "type": "string",
              "description": "Inspection mode for this range.",
              "enum": ["coverage", "detail"]
            }
          },
          "required": ["start", "end", "objective", "mode"],
          "additionalProperties": false
        },
        "minItems": 1
      }
    },
    "required": ["ranges"],
    "additionalProperties": false
  }
}
```

Strict Batched Interface. The schema defines the interface between global planning and local inspection. The ranges array forms one atomic batch, and every range must provide temporal bounds, a self-contained inspection objective, and either coverage mode or detail mode; additional fields are rejected. Independent entries execute in parallel without sharing local context. Coverage mode produces timestamped grids over an extended interval, whereas detail mode supplies separate high-detail images for a short interval. The tool then validates the returned references and forwards at most three selected visual units per range with the textual evidence for the next global reasoning round.

I Limitations and Future Work

The current implementation uses predefined parameters for global sampling and each inspection mode, together with an upper bound on reasoning rounds. While the Orchestrator already selects inspection modes and plans subsequent inspections based on the evolving evidence state, these lower-level configurations are not yet determined dynamically during inference. Future work can allow the Orchestrator to adjust temporal sampling density, spatial resolution, and evidence-return budget according to the question and accumulated evidence. The current round-based execution can also be extended to asynchronous scheduling, where completed Executor results can immediately trigger new inspections without waiting for all remaining tasks in the same parallel batch to finish.

The current framework focuses on video question answering based primarily on visual inspection. A natural extension is to incorporate other timestamped evidence sources, such as audio events and speech segments, into the same aggregation mechanism. Beyond video question answering, the same parallel evidence-seeking and aggregation process can support more open-ended and multi-source settings, including multimodal deep research, where information needs may evolve throughout reasoning and evidence must be gathered and integrated across different modalities and sources.

J AI Use Statement

Generative AI models are integral to the evaluated inference procedure. GPT-5 serves as the Orchestrator and Executors in the proprietary-agent experiments, while Qwen3.6-27B and Gemma4-31B are evaluated as alternative backbones. The Methodology and Experiments sections describe their roles and evaluation settings; Appendix A reports the GPT-5 version and proprietary-agent configuration, while Appendices G and H provide the prompts, runtime templates, and tool schema. Model outputs serve as experimental predictions and intermediate evidence reports; they do not replace benchmark questions, reference answers, or ground-truth labels. For paper writing, ChatGPT was used only for grammatical corrections and readability improvements. The authors reviewed all text, verified the claims and numerical results against experimental records, and take responsibility for the manuscript, including the final wording, quantitative claims, tables, figures, and the conclusions of the study.

K Ethics Statement

This work advances accurate and efficient long-video understanding. All data used in this study are drawn from publicly available datasets and are used in accordance with their respective license terms. Our evaluation relies on existing videos and annotations from six benchmarks and does not involve recruiting participants, collecting new human-subject data, or introducing additional personal information. The experiments are conducted solely for benchmark-based research evaluation and do not involve real-world deployment or automated decision-making about individuals. We do not collect or release personal data beyond the content already provided by the benchmark datasets. Responsible use of long-video understanding technologies should respect content creators' rights, the privacy and consent of individuals appearing in videos, and the terms governing the source data.

L Reproducibility Statement

The Methodology section specifies the inference procedure and budget variables, and the Experiments section reports the benchmarks, metrics, evaluation splits, baseline groupings, and principal configurations. Appendix A records the GPT-5 model version, visual-input budgets, number of concurrent Executors, evidence-return cap, and reasoning-round limit. Appendices G and H provide the complete Orchestrator and Executor instructions, runtime input templates, and strict inspection-tool schema. The complete code and the sample IDs for the fixed, stratified LVBench subset used in the ablation experiments will be made publicly available after completion of the relevant institutional review. Together, these sections document the reported experimental design and principal settings; the benchmark data are available from the cited sources for independent reuse and verification.