VIDEOLOOP: LOOPED WORKING MEMORY AGAINST SEMANTIC THRASHING IN LONG-FORM VIDEO AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-form video understanding requires multimodal agents to iteratively gather evidence over many reasoning steps. However, most existing agentic methods suffer from *semantic thrashing*: as append-only working memory grows, attention to key evidence collapses, and the agent loses access to what it has already found. First, we provide a structural argument showing that append-only memory can incorporate newly observed target evidence, but cannot remove accumulated noise or prevent ordered context growth without a rewrite operator. Second, motivated by this analysis, we propose VideoLoop, a multimodal agent with two coupled loops. The outer loop reasons over the video and the inner loop, after each step, retrieves artifacts from an unbounded filesystem of past observations and intermediate analysis, and rewrites a bounded working memory. Extensive experiments demonstrate the effectiveness of VideoLoop, which improves four popular LVLM backbones in a plug-and-play manner, with an average gain of 4.2% points over baseline on VideoMME (*long*). Further analysis of working memory suggests that VideoLoop mitigates semantic thrashing: on the hardest quarter of VideoMME (*long*) questions, a blind judge that reads only the agent's context answers 81.1% correctly, versus 60.9% for the append-only agent. With Gemini 3.1 Pro, VideoLoop reaches 88.3% on VideoMME (*long*), 88.8% on VideoMMMU, and 80.9% on LongVideoBench (*long*).[1]

1 INTRODUCTION

Long-form video understanding (Lin et al., 2026; Luo et al., 2025; Wang et al., 2025b; Tang et al., 2025) requires reasoning over thousands of frames spanning minutes to hours, where the evidence relevant to a question is sparse and scattered across non-contiguous segments. Although recent studies attempt to scale representations to hour-level contexts (Lin et al., 2025; Shu et al., 2025), untrimmed videos contain massive natural redundancy (Yao et al., 2025; Li et al., 2025b). Therefore, numerous methods explore adaptive temporal search, pivot frame retrieval, and step-by-step reasoning (Ye et al., 2025; Gao et al., 2026; Li et al., 2025a; Bhatnagar et al., 2026; Li et al., 2026b). In this setting, general large vision-language models (LVLMs) (Gemini Team, Google, 2023; 2024; Google DeepMind, 2026; OpenAI, 2023; 2024) still struggle to ingest such massive contexts reliably. Agentic systems (Li et al., 2026a; Lin et al., 2026; Yan et al., 2026; Zhang et al., 2025; He et al., 2026) address this gap by reasoning iteratively: at each step, the agent observes a region of the video, updates a working memory of accumulated evidence, and decides where to look next. Iterative agents consistently outperform baselines on long-form video benchmarks.

However, most prior agentic methods (Zhang et al., 2025; Li et al., 2026a; Yan et al., 2026; He et al., 2026; Wang et al., 2024) share a restrictive design: a single reasoning loop with an *append-only* working memory. As the agent observes more, this memory accumulates within the LVLM's bounded context. Once it exceeds this bound, attention to key evidence of the user's query is severely diluted by the accumulation of observations, and the agent loses reliable access to what it has already found. As shown in Figure 1(a), we describe this systemic failure as *semantic thrashing*, in analogy to OS thrashing (Denning, 1968a): the agent expends increasing computational effort while

[1]The code will be released upon publication.

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.

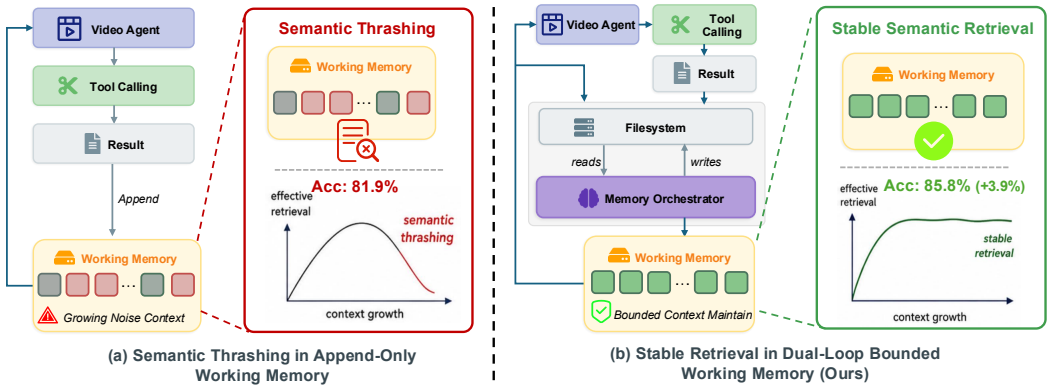


Figure 1: **Existing Append-only working memory leads to semantic thrashing, our VideoLoop sustains stable retrieval via a dual-loop bounded working memory.** **(a)** A single-loop video agent appends every tool result to a working memory bounded by the LVLM context. As context accumulates, attention to key evidence is diluted, and effective retrieval first peaks and then collapses as context continues to grow. The append-only agent achieves an 81.9% final accuracy on VideoMME (*long*). **(b)** VideoLoop adds an inner orchestration loop. Tool results are written to an unbounded filesystem, and a memory orchestrator retrieves from the filesystem to rewrite a bounded working memory after every outer step. Our method improves task accuracy by +3.9 points on VideoMME (*long*) to 85.8%. Both configurations use Gemini 3 Flash.

its grounding on prior findings degrades, as a thrashing OS spends most cycles swapping pages rather than running useful work. We further show that semantic thrashing is a structural property of append-only updates rather than an incidental tuning issue. Treating the working memory and the optimal evidence set as subsets of a common evidence universe, append-only can reduce missing target evidence when new relevant observations arrive, but cannot remove accumulated irrelevant evidence once it has entered memory. Closing this gap requires an operator that can *remove* evidence from the working memory, which append-only memory by definition forbids. Moreover, append-only memory grows increasingly costly as the ordered observation log expands. Therefore, append-only memory fails along two dimensions: it cannot discard irrelevant evidence, and it cannot bound the cost of retaining past observations.

These limitations motivate a decoupled design with two complementary properties: (i) an unbounded external store retaining all observations losslessly, and (ii) a bounded working memory rewritten at every step under a fixed budget. To this end, in Figure 1(b), we propose **VideoLoop**, a dual-loop architecture, which realizes this decoupled design. The *outer loop* is a multimodal agent that observes the video and writes its findings to a persistent filesystem of past observations and intermediate analysis. The *inner loop* is a memory orchestrator that, after each outer step, retrieves question-relevant artifacts from the filesystem and rewrites a concise working memory. Decoupling reasoning from memory management allows each loop to operate within a fixed context budget, while the orchestrator retains random-access read privileges on the full trajectory history during consolidation. Furthermore, we evaluate our VideoLoop on three popular benchmarks for long-form video understanding. To probe memory quality, a blind judge that reads only the agent's accumulated context shows that append-only retrievability drops from 86.7% on the easiest tasks (Q1) to 60.9% on the hardest (Q4), while our method drops only from 94.7% to 81.1%. With Gemini 3.1 Pro as the policy model, it reaches 88.3% on VideoMME (*long*), 88.8% on VideoMMMU, and 80.9% on LongVideoBench (*long*), with absolute gains of 3.2% to 4.5% over the native LVLM. The dual-loop architecture is also plug-and-play across LVLM backbones, yielding consistent improvements without retraining. Overall, our contributions are summarized as follows:

- **Structural motivation for semantic thrashing.** We formalize semantic thrashing as a structural failure mode of append-only video-agent memory: such updates fail to remove accumulated irrelevant evidence or prevent ordered context growth without a rewrite operator.
- **Dual-loop bounded working memory.** VideoLoop decouples reasoning from memory management through an unbounded filesystem and bounded working memory rewritten at every step.
- **Strong empirical performance.** Extensive experiments demonstrate that VideoLoop effectively mitigates memory degradation, improves accuracy on three long-form video benchmarks, and generalizes across diverse LVLM backbones without retraining.

2 Related Work

Agentic Video Understanding. Agentic video understanding pairs an LLM controller with a set of multimodal tools inside an iterative control loop (Yao et al., 2023; Zhang et al., 2025; Wang et al., 2024; Fan et al., 2024; Zhang et al., 2024a; Wang et al., 2025b; Pang & Wang, 2025; Lin et al., 2026; Li et al., 2026a; Zhang et al., 2024b; Yang et al., 2025b; Chen et al., 2025; Rege et al., 2026; Yan et al., 2026; Liu et al., 2025; Yu et al., 2026). At each step the controller reads the current state, picks a tool, and adds the result to what it knows about the video. The hard part is gathering and integrating evidence across temporal spans longer than any single context window. Most existing systems handle this with predefined pipelines or prebuilt indices (Zhang et al., 2024a; Pang & Wang, 2025; Chen et al., 2025; Yan et al., 2026; Wang et al., 2024; 2025b; Lin et al., 2026; Li et al., 2026a), which are predictable but rigid. ReAct-style observe–act–reason loops (Yao et al., 2023) are the most common design, applied to keyframe selection, caption chains, and tree-structured retrieval. DVD (Zhang et al., 2025), for example, builds a multi-granular video database that a ReAct-style agent queries. Differently, our VideoLoop runs inside a coding sandbox and writes code to direct its own exploration. With only a minimal toolset, it composes retrieval and analysis routines as it goes, without upfront video preprocessing or committing to a fixed schema.

Memory Mechanisms in Video Agents. Agentic memory has become a common way to handle long-context reasoning, with applications in long-form video understanding (Yin et al., 2026; Long et al., 2026; Yeo et al., 2026; Wang et al., 2025a; Hu et al., 2025b; Lin et al., 2023; Wang et al., 2023; Lv et al., 2026; Sanders et al., 2024) and persistent dialogue agents (Packer et al., 2023; Zhong et al., 2024; Chhikara et al., 2025; Zhou et al., 2023; Feng et al., 2026). One family compresses memory through sliding windows, KV-cache pruning, and backtracking (Xie et al., 2025; Yang et al., 2025a; Zuo et al., 2025) or learned retention and deletion in VideoMem's global memory buffer (Jin et al., 2025). VideoARM records observations and reasoning traces in hierarchical multimodal memory (Yin et al., 2026). WorldMM uses an LLM to consolidate semantic triplets into an evolving knowledge graph (Yeo et al., 2026), while MemGPT supports active context management for general-purpose agents (Packer et al., 2023). In contrast, VideoLoop decouples video exploration from working-memory maintenance through a dedicated inner-loop agent. After each outer step, this agent combines three capabilities: (i) active retrieval of prior evidence from a persistent filesystem, (ii) joint consolidation of evidence across video segments and memory sections, and (iii) section-level memory editing under a fixed token budget.

3 Methodology

3.1 Semantic Thrashing Problem

To formally analyze the failure modes of long-form video agents, we establish a structural analogy between the effective context limits of large vision-language models (LVLMs) (Gemini Team, Google, 2023; 2024; Google DeepMind, 2026; OpenAI, 2023; 2024) and the memory-scheduling limits of classical operating systems (OS).

Standard OS Thrashing (Denning, 1968a). In multiprogramming environments, a process's active memory demand can be characterized by its *working set* (Denning, 1968b). In particular, the working set $W_i(t, \delta)$ denotes the set of distinct memory pages referenced by process i within the recent time window $[t - \delta, t]$. Let M denote the total physical memory capacity, and let $\mathcal{I}(t)$ be the set of active processes at time t. *Thrashing* (Denning, 1968a) occurs when aggregate memory demand $\Lambda(t)$ exceeds physical capacity M:

$$\Lambda(t) \;=\; \sum_{i \in \mathcal{I}(t)} \big|\, W_i(t, \delta) \,\big| \;>\; M. \tag{1}$$

Once this persists, the OS spends most cycles swapping pages, and useful throughput collapses.

Semantic Thrashing in Video Agents. We formalize long-form video understanding as a sequential process of evidence gathering. Given a video $\mathcal{V}$ and query q, let $\mathcal{U}_q$ denote the universe of atomic evidence units relevant to query q, and let $\mathcal{K}_q \subseteq \mathcal{U}_q$ denote the latent set of critical evidence required to answer q. At reasoning step t, the agent ingests a new observation o_t and updates its working

memory $\mathcal{M}_t$. Most prior video agentic systems (He et al., 2026; Li et al., 2026a; Lin et al., 2026; Zhang et al., 2025) update the working memory $\mathcal{M}_t$ in an *append-only* manner:

$$\mathcal{M}_t \;=\; \mathcal{M}_{t-1} \cup \{o_t\}. \tag{2}$$

However, current LLMs/LVLMs exhibit a bounded effective context capacity C (An et al., 2025; Hsieh et al., 2024; Liu et al., 2024). As working memory $\mathcal{M}_t$ grows beyond this effective bound, $|\mathcal{M}_t| > C$, the agent can no longer reliably retrieve and integrate the evidence in $\mathcal{K}_q$. Consequently, additional observations can dilute relevant evidence and reduce the reliability of retrieval and reasoning. Analogous to OS thrashing, we term this failure mode *semantic thrashing*: the agent keeps accumulating evidence while its reasoning grows unstable and less grounded.

State Divergence as A Structural Indicator of Semantic Thrashing. We use the following as a conceptual diagnostic rather than a theorem-like reduction from OS thrashing. Treating $\mathcal{M}_t$ and $\mathcal{K}_q$ as subsets of a common evidence universe, we decompose their gap into *missing target evidence* $\Delta^{(t)}_{\text{target}} = \mathcal{K}_q \setminus \mathcal{M}_t$ and *redundant noise* $\Delta^{(t)}_{\text{pred}} = \mathcal{M}_t \setminus \mathcal{K}_q$, and define the *state divergence*:

$$\text{Dist}(\mathcal{M}_t, \mathcal{K}_q) \;:=\; \left|\Delta^{(t)}_{\text{target}}\right| + \left|\Delta^{(t)}_{\text{pred}}\right| \;=\; \left|\,\mathcal{M}_t \triangle \mathcal{K}_q\,\right|. \tag{3}$$

This symmetric-difference cardinality is a conceptual diagnostic for memory quality.

With $M_t = M_{t-1} \cup \{o_t\}$, one update changes the diagnostic by

$$\text{Dist}(\mathcal{M}_t, \mathcal{K}_q) - \text{Dist}(\mathcal{M}_{t-1}, \mathcal{K}_q) = -|\{o_t\} \cap (\mathcal{K}_q \setminus \mathcal{M}_{t-1})| + |\{o_t\} \setminus (\mathcal{K}_q \cup \mathcal{M}_{t-1})|. \tag{4}$$

Eq. 4 separates the effect of appending o_t into useful and noisy additions. The term $o_t \cap (\mathcal{K}_q \setminus \mathcal{M}_{t-1})$ captures newly observed target evidence that was missing from memory, thereby reducing divergence. In contrast, $o_t \setminus (\mathcal{K}_q \cup \mathcal{M}_{t-1})$ captures newly introduced content that is neither target evidence nor previously stored, thereby increasing divergence. Append-only memory cannot remove such noise once added, so noisy trajectories gradually consume the context budget, obscure target evidence, and lead to semantic thrashing.

Implications for Working Memory Design. Eq. 2 and Eq. 4 show the reason why append-only working memory is fragile in long-horizon or high-noise scenarios: critical evidence can enter $\mathcal{M}_t$, but non-target content in $\mathcal{M}_t \setminus \mathcal{K}_q$ remains unless later updates can remove or rewrite it. This motivates a *decoupled* memory architecture with two complementary properties: (i) an *unbounded external store* that retains all observations and intermediate artifacts losslessly, so that no evidence is lost prematurely, and (ii) a *bounded working memory rewritten after each step* under a fixed token budget, which can delete redundant content in $\mathcal{M}_t \setminus \mathcal{K}_q$ and re-import missing target evidence in $\mathcal{K}_q \setminus \mathcal{M}_t$ from (i).

3.2 Video Agent with Persistent Sandbox

Overview. Figure 2 illustrates the dual-loop architecture. Given a long-form video $\mathcal{V}$ and a query q, VideoLoop operates as a sandboxed multimodal agent with five core components: a policy model π instantiated by a multimodal LLM, a toolkit $\mathcal{T}$, a working memory $\mathcal{M}$, a memory orchestration π_m for working memory rewriting, and a persistent sandbox environment with a unbounded filesystem $\mathcal{F}$. VideoLoop follows a dual-loop workflow: the outer loop employs the policy model π to iteratively explore the video and save observations and artifacts to the filesystem $\mathcal{F}$, while the inner loop uses π_m to dynamically consolidate evidence units that can be recovered from those artifacts into an updated working memory. Appendix Algo. 1 gives the full procedure.

Basic Toolkit. The toolkit $\mathcal{T}$ of the outer loop comprises four primitives: ANALYZE($\hat{\mathcal{V}}$, $\hat{q}$) answers the generated question $\hat{q}$ based on the selected video frames $\hat{\mathcal{V}}$. TRANSCRIBE($\mathcal{V}$) returns the full timestamped transcript when requested by the policy and caches it after the first call. EXECUTE(P) runs arbitrary code P in a sandbox environment with image, video, and numerical libraries, persisting all artifacts (frames, captions, transcripts, a manifest of reasoning trace, etc.) on the sandbox filesystem $\mathcal{F}$. ANSWER(y) commits an answer y and terminates the agentic trajectory.

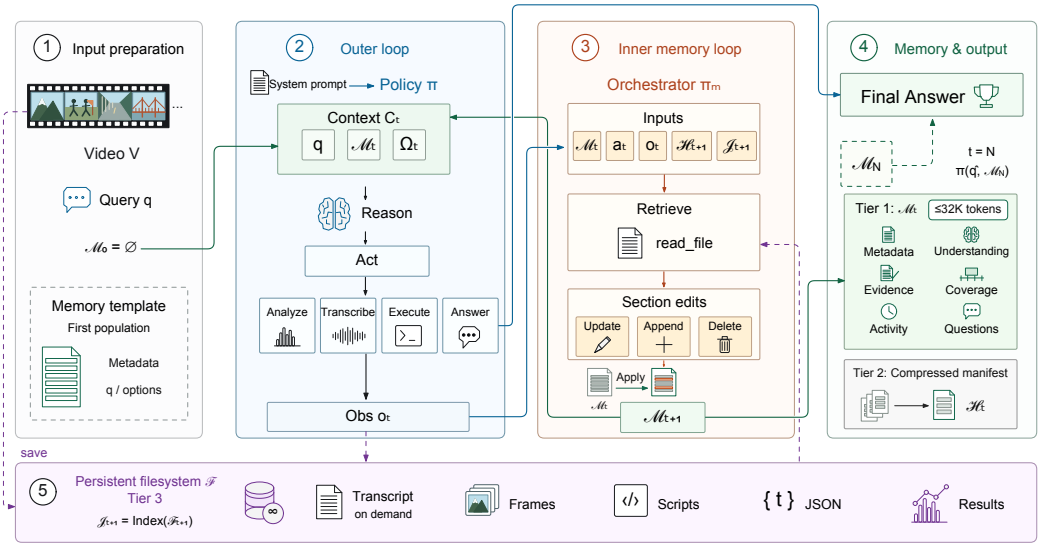


Figure 2: **Overview of VideoLoop's dual-loop architecture.** Working memory starts empty. The outer loop reasons about the task and explores the video with transcription and visual analysis tools as needed. After each outer step, the inner loop reads the question and the new observation, retrieves relevant evidence from the external store, and rewrites a compact working memory.

3.3 Dual-Loop Bounded Working Memory

Starting State. Working memory starts empty ($\mathcal{M}_0 = \emptyset$), so at $t = 0$ the outer policy sees only q. After the first step, π_m writes the six-section template from q and the first observation, including the video metadata, the question, and the options. The video is available in the sandbox, while the transcript is stored in the filesystem only after TRANSCRIBE$(\cdot)$ is called.

Outer Loop: Agentic Reasoning & Acting. At step t, π reads the bounded context:

$$C_t = \{q, \mathcal{M}_t, \Omega_t\}, \tag{5}$$

where the $\mathcal{M}_t$ denotes the current rewritten working memory, and Ω_t is a sliding window containing the thought-action-observation triplets from the nearest k most recent iterations. Conditioned on the context C_t, the policy model produces reasoning r_t and an action a_t, and executing a_t returns an observation o_t, respectively. The new observation o_t will be written into the filesystem before its useful evidence is selectively admitted into $\mathcal{M}_{t+1}$. The outer loop *never* ingests raw artifacts from prior iterations: it only accesses user query q, current work memory $\mathcal{M}_t$, and the nearest sliding window Ω_t. Each component of C_t is size-controlled: the working memory is bounded by a predefined token budget $B_{\mathcal{M}}$, and $|\Omega_t|$ is capped at the k most recent turns. Therefore, the total context size $|C_t|$ remains bounded by a constant independent of the iteration t.

Inner Loop: Orchestrator Memory Rewrite. After each outer step, a separate LLM π_m, the *memory orchestrator*, performs an active rewrite to produce a newly working memory $\mathcal{M}_{t+1}$ based on the previous working memory $\mathcal{M}_t$, the current action a_t and the corresponding observation o_t, a compressed manifest $\mathcal{H}_{t+1}$ of all prior actions, and $\mathcal{I}_{t+1} = \text{Index}(\mathcal{F}_{t+1})$ is an index over the current filesystem. By design, π_m guarantees the invariant $|\mathcal{M}_t| \leq B_{\mathcal{M}}$ for all t, where $B_{\mathcal{M}}$ is a token budget chosen below the effective context capacity C.

Termination and Fallback Generation. The trajectory terminates whenever $a_t = \text{ANSWER}(\cdot)$, returning the final response $\hat{y}$. If the agent has not explicitly committed by the maximum iteration N, a fallback answer is generated by querying the policy directly on the last consolidated memory, $\hat{y} = \pi(\hat{q}, \mathcal{M}_N)$, where $\hat{q}$ denotes the query q appended with an instruction that requires the model to produce an answer immediately, as specified in Algorithm 1.

3.4 FILESYSTEM-BASED MEMORY ORCHESTRATION

Following the taxonomy of prior agentic systems (Yin et al., 2026; Jin et al., 2025; Li et al., 2026a), we decompose π_m along three axes: *storage*, *retrieval*, and *consolidation*. Appendix G compares each axis with seven prior memory systems, and Appendix I shows a full rewrite.

Hierarchical Storage. VideoLoop maintains a three-tier memory hierarchy. **Tier 1.** The bounded working memory $\mathcal{M}_t$ visible to the outer loop is a typed document $\mathcal{M}_t = \sigma_1^{(t)} \oplus \cdots \oplus \sigma_6^{(t)}$ with six ordered sections: metadata, narrative understanding (updated as needed), timestamped evidence, temporal coverage, activity log, and open investigation targets (Appendix I). The working memory $\mathcal{M}_t$ is bounded by the token budget $B_{\mathcal{M}}$. **Tier 2.** The step manifest $\mathcal{H}_t$ is a compressed log of all prior actions with their action parameters, such as timestamps, generated queries, and executed code. Entries older than the most recent k are batch-summarized to govern which entries survive compression (Appendix H). **Tier 3.** The unbounded sandbox filesystem $\mathcal{F}_t$ stores all extracted frames, analysis outputs, and intermediate scripts losslessly across iterations, preserving the raw material from which evidence units can be recovered. $\mathcal{H}_t$ bridges the filesystem and working memory with a navigable, importance-weighted history.

Active Filesystem Retrieval. Before emitting its edits, π_m retrieves question-relevant text artifacts, such as prior frame analyses and intermediate scripts, using $\mathcal{T}_{\pi_m} = \{\texttt{read_file}\}$. The filesystem index in its context lists the available files and frames.

Working Memory Rewrite. At $t = 0$, π_m fills the empty memory with the six sections. At each later step, it emits edits $\mathcal{D}_t \subseteq \{\text{UPDATE}, \text{APPEND}, \text{DELETE}\}$ for $\mathcal{M}_t$, and sections without edits stay unchanged:

$$\mathcal{M}_{t+1} \;=\; \textsc{Apply}\big(\mathcal{D}_t,\ \mathcal{M}_t\big), \qquad \mathcal{D}_t \;=\; \pi_m(q, \mathcal{M}_t, a_t, o_t, \mathcal{H}_{t+1}, \mathcal{I}_{t+1}). \tag{6}$$

The edits are per-section in syntax but *cross-section in semantics*: the orchestrator π_m decides $\mathcal{D}_t$ over all sections jointly. For instance, the narrative understanding section $\sigma_2^{(t)}$ and the timestamped evidence section $\sigma_3^{(t)}$ are revised coherently against each other.

This closes the formal loop with Sec. 3.1: when π_m retrieves a missing target evidence item or discards redundant content, the rewrite is beneficial if removed noise plus imported target evidence outweighs deleted target evidence plus newly added noise. Let the following four quantities summarize one rewrite:

$$\begin{aligned} \rho_t &= |(\mathcal{M}_t \setminus \mathcal{K}_q) \setminus \mathcal{M}_{t+1}|, \quad \sigma_t = |(\mathcal{K}_q \setminus \mathcal{M}_t) \cap \mathcal{M}_{t+1}|, \\ \mu_t &= |(\mathcal{K}_q \cap \mathcal{M}_t) \setminus \mathcal{M}_{t+1}|, \quad \nu_t = |\mathcal{M}_{t+1} \setminus (\mathcal{K}_q \cup \mathcal{M}_t)|. \end{aligned} \tag{7}$$

Here ρ_t is removed non-target content, σ_t is imported missing target evidence, μ_t is target evidence accidentally removed, and ν_t is newly introduced non-target content. Then, we can derive

$$\text{Dist}(\mathcal{M}_{t+1}, \mathcal{K}_q) - \text{Dist}(\mathcal{M}_t, \mathcal{K}_q) = -\rho_t - \sigma_t + \mu_t + \nu_t. \tag{8}$$

Therefore, $\text{Dist}(\mathcal{M}_{t+1}, \mathcal{K}_q) \leq \text{Dist}(\mathcal{M}_t, \mathcal{K}_q)$ only when removed noise and imported target evidence are at least as large as lost target evidence and newly added noise, *i.e.*, $\rho_t + \sigma_t \geq \mu_t + \nu_t$, with strict improvement under strict inequality. This is a condition on rewrite quality, not an unconditional guarantee. Furthermore, active retrieval maintains a compact, query-relevant context across long-horizon trajectories, matching the design goal implied by Eq. 8.

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Evaluation Benchmarks. We evaluate our VideoLoop on three video understanding benchmarks. VideoMME (*long*) (Fu et al., 2025) contains 900 multiple-choice questions over 30–60 minute videos, such as lectures, sports, documentaries, and entertainment. VideoMMMU (Hu et al., 2025a) comprises 900 questions drawn from 300 educational videos and evaluates models across the Perception, Comprehension, and Adaptation tracks, with 300 questions per track. LongVideoBench

Method	VideoMME *(long)*	VideoMMMU				LongVideoBench *(long)*
		Per.	Comp.	Adapt.	Overall	
Video agentic systems						
VideoAgent (Wang et al., 2024)	49.0	–	–	–	–	–
LLoVi (Zhang et al., 2024a)	50.2	–	–	–	–	–
VideoTree (Wang et al., 2025b)	54.2	–	–	–	–	–
QuoTA (Luo et al., 2026)	55.7	–	–	–	–	–
OmAgent (Zhang et al., 2024b)	60.5	–	–	–	–	–
VCA (Yang et al., 2025b)	56.3	–	–	–	–	–
M3-Agent (Long et al., 2026)	61.8	–	–	–	–	–
DVD (Zhang et al., 2025)	67.3	–	–	–	–	68.6
MR. Video (Pang & Wang, 2025)	63.4	–	–	–	–	61.6
LensWalk (Li et al., 2026a)	71.4	81.3	81.3	72.3	78.3	70.6
VideoChat-A1 (Wang et al., 2026)	71.2	–	–	–	–	–
EGAgent (Rege et al., 2026)	74.1	–	–	–	–	–
VideoSeek (Lin et al., 2026)	81.2	–	–	–	–	73.5
VideoARM (Yin et al., 2026)	81.2	–	–	–	–	76.4
Native LVLMs and Ours						
GPT-4o (OpenAI, 2024)	72.1	66.0	62.0	55.7	61.2	60.9
Gemini 1.5 Pro (Gemini Team, Google, 2024)	77.4	59.0	53.3	49.3	53.9	58.6
OpenAI o3 (OpenAI, 2025)	84.9	79.3	75.7	71.3	75.4	–
Gemini 3 Flash* (Google DeepMind, 2025)	80.7	84.0	83.0	83.7	83.6	67.9
+ **VideoLoop**	85.8 (+5.1)	90.3	87.3	85.3	87.7 (+4.1)	73.8 (+5.9)
Gemini 3.1 Pro* (Google DeepMind, 2026)	83.8	85.7	83.7	84.3	84.6	77.7
+ **VideoLoop**	**88.3** (+4.5)	**90.7**	**87.7**	**88.0**	**88.8** (+4.2)	**80.9** (+3.2)

Table 1: Benchmark comparison (accuracy, %). Bold denotes column best; parentheses give overall gains over the preceding native backbone in percentage points. VideoMME and LongVideoBench use their long subsets with the official subtitles. VideoMMMU uses all 900 questions (300 per track): Perception (Per.), Comprehension (Comp.), Adaptation (Adapt.), and Overall. Results marked with * are reproduced by us under the same settings. Sources and policy models are detailed in Appendix M. Dashes denote unavailable results.

(*long*) (Wu et al., 2024) evaluates detailed retrieval and reasoning across diverse web videos lasting up to 1 hour, with subtitles. We report results for its long split, comprising 564 questions from 188 videos, each 900–3600 seconds long. Claude Opus 4.8 sorts VideoMME (*long*) questions by difficulty into Q1–Q4 (easiest to hardest, 225 each), followed by human review (Appendix D).

Baseline Methods. We compare VideoLoop with native large vision-language models (LVLMs) that answer from video context in a single inference pass, and with video-agentic systems that perform multi-step video reasoning through iterative exploration.

Implementation Details. We use Gemini 3.1 Pro and Gemini 3 Flash as the policy model for both the outer-loop agent and the inner-loop orchestrator. For the *outer-loop agent*, we set the maximum number of reasoning iterations N to 50 and the minimum to 6. The ANALYZE$(\cdot)$ tool uses the default policy model in its native multimodal mode to inspect selected video content. Working memory starts empty ($\mathcal{M}_0 = \emptyset$), with no pre-loop initialization. The TRANSCRIBE$(\cdot)$ tool returns the official subtitles for VideoMME and LongVideoBench. VideoMMMU provides no subtitles, so we transcribe its audio with Whisper-large (Radford et al., 2023). The tool is invoked only when selected by the outer policy. For the *inner-loop orchestrator*, we set the working memory budget $B_{\mathcal{M}}$ to 32K tokens and limited the number of selected key frames to a maximum of 6. In practice, we set the recent window to $k = 8$ message groups for the outer loop, and the orchestrator compresses the oldest $b = 10$ manifest records into one summary once $h = 30$ unsummarized records accumulate.

4.2 MAIN RESULTS

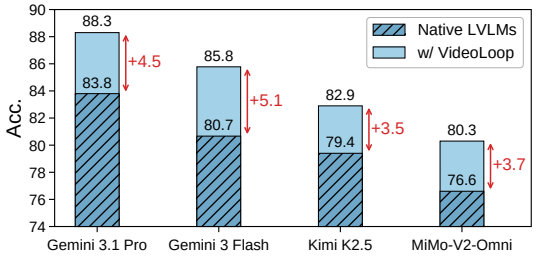
Overview. Table 1 compares VideoLoop with native LVLMs and prior video agentic systems across three long-video benchmarks. With Gemini 3.1 Pro as the policy model π, VideoLoop obtains 88.3% on VideoMME (*long*), 88.8% overall on VideoMMMU, and 80.9% on LongVideoBench (*long*).

Comparison with Base Models. Compared with its native base model, Gemini 3.1 Pro, VideoLoop improves by +4.5, +4.2, and +3.2 points on VideoMME (*long*), VideoMMMU, and

Table 2: Memory ablation on VideoMME (*long*) with Gemini 3 Flash (one run per configuration). Accuracies use 900 questions overall and 225 per fixed-task-difficulty quartile (Q1 easiest, Q4 hardest), shared across rows. Please see Appendix D for the grouping protocol. Gains use unrounded accuracies.

Memory Design		VideoMME (*long*)				
Dual-Loop	**Filesystem**	**All**	**Q1**	**Q2**	**Q3**	**Q4**
Native Single-Pass LVLMs		80.7	93.3	86.2	73.3	69.8
Append-only Agent		81.9	93.8	83.6	79.6	70.7
✓		83.3	**94.7**	86.7	79.6	72.4
✓	✓	**85.8**	**94.7**	**87.1**	**81.3**	**80.0**

Figure 3: Ablation study on policy model π.



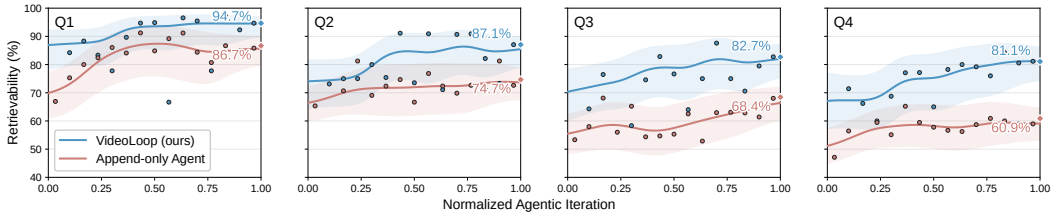


Figure 4: Illustration of semantic thrashing on VideoMME (*long*). Four panels present fixed task difficulty quartiles Q1–Q4 from left to right (easiest to hardest), with the same grouping used across memory designs. The curves show blind-judge answer accuracy on frozen context snapshots, used as a proxy for evidence retrievability. Diamonds and labels at the right end give final-snapshot accuracy. Semantic thrashing occurs when retrievability early plateaus or degrades despite continued agentic iterations.

LongVideoBench (*long*), respectively. Its margins over the strongest prior agentic methods are +7.1, +10.4, and +4.5 points on the three benchmarks. The VideoMMMU breakdown shows improvements across all three cognitive tracks for both backbones.

Effect of Policy Model π. Figure 3 evaluates different policy models as the reasoning engine. VideoLoop consistently improves over the corresponding native LVLMs, achieving gains of $+4.5$, $+5.1$, $+3.5$, and $+3.7$ points with Gemini 3.1 Pro, Gemini 3 Flash, Kimi K2.5, and MiMo-V2-Omni, respectively. These results suggest that VideoLoop is broadly compatible with different policy models and can effectively convert stronger reasoning engines into higher video understanding capability.

4.3 Analysis of VideoLoop Design Components

Table 2 separates agentic exploration from memory design. Append-only raises accuracy over native inference from 80.7 to 81.9 ($+1.2$). Dual-loop rewriting reaches 83.3 ($+1.4$ over append-only), with its largest gain on Q2 ($+3.1$) and no change on Q3. Adding filesystem access raises accuracy to 85.8 ($+2.4$ over dual-loop), gaining $+1.8$ and $+7.6$ on Q3 and Q4. Full VideoLoop exceeds append-only by $+3.9$ points (largest on Q4, $+9.3$) and native inference by $+5.1$ points.

4.4 Further Analysis

Evidence Retrievability Reveals Semantic Thrashing. Figure 4 evaluates evidence retrievability through a context-only answerability proxy. At sampled iterations, a separate blind judge (Gemini 3.1 Flash-Lite) receives the question, answer options, and a frozen context snapshot, without access to the video, filesystem, or tools. Its answer accuracy measures how well the current context supports answering the question. Appendix E details the evaluation protocol. VideoLoop maintains stable or gradually improving retrievability across quartiles, showing that additional iterations help accumulate and preserve usable evidence. In contrast, the append-only agent often plateaus early (Q2 and Q4) or even degrades (Q1), indicating that simply appending observations can dilute or obscure critical evidence rather than improve the effective working memory. As task difficulty increases from Q2 to Q4, the gap becomes substantially larger: In Q4, the final-snapshot judge accuracy is 81.1% for VideoLoop and 60.9% for the append-only agent.

Table 3: **Token cost vs. accuracy on VideoMME (*long*).** Per-question means, in thousands of tokens. Relative to append-only, full VideoLoop gains +3.9 pp in accuracy with additional cost of 0.5% tokens; dual-loop only gains +1.4 pp with additional cost of 5.2% tokens.

Memory Design	Tokens / Question (K)			Performance	
	In	Out	Total	Δ	Acc.
(a) Append-only	584.6	30.3	614.9	—	81.9
(b) Dual-loop only	590.0	57.0	647.0	+5.2%	83.3
(c) **VideoLoop** (*ours*)	559.0	59.2	**618.2**	**+0.5%**	**85.8**

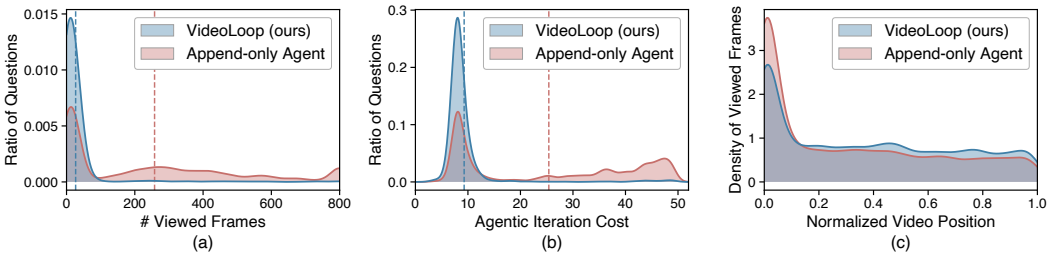


Figure 5: Agentic behavior analysis on VideoLoop and append-only agent on VideoMME (*long*). (a) Distribution of the number of viewed frames per question. (b) Distribution of agentic iteration cost per question. (c) KDE density of viewed frames over normalized video positions, where 0 and 1 indicate the beginning and end of a video. Dashed lines in (a) and (b) indicate the mean values.

Token Efficiency Analysis across Memory Designs. Table 3 compares token usage and accuracy across different memory designs on VideoMME (*long*) with Gemini 3 Flash. The append-only baseline uses 614.9K tokens per question and achieves 81.9% accuracy. Adding the dual-loop workflow improves accuracy to 83.3%, but increases token usage to 647.0K, a +5.2% overhead, mainly due to the additional output tokens introduced by memory rewriting (inner loop). In contrast, the full VideoLoop design with filesystem achieves the highest accuracy, 85.8%, while using only 618.2K tokens, nearly matching the append-only baseline with just +0.5% overhead. Although VideoLoop produces more output tokens, it reduces the number of input tokens from 584.6K to 559.0K, suggesting that the filesystem helps externalize and selectively reuse intermediate evidence rather than repeatedly carrying the full history in context.

Agentic Behavior Analysis. As shown in Figure 5(a), VideoLoop concentrates its viewed-frame distribution in the low-cost regime, while the append-only agent exhibits a substantially longer tail, indicating that append-only memory accumulation tends to trigger excessive visual inspection. A similar trend is observed in Figure 5(b): VideoLoop requires fewer agentic iterations per question, whereas the append-only agent often continues for many more reasoning steps. These results suggest that *VideoLoop's memory orchestration provides a more compact and effective working memory*, reducing redundant evidence collection and mitigating the iterative overhead caused by unstructured memory growth. Figure 5(c) further indicates that *VideoLoop produces a more temporally balanced distribution of viewed frames* across the normalized video timeline, rather than simply focusing on a narrow segment at the beginning of the videos.

5 Conclusion

In this paper, we present VideoLoop, a dual-loop agentic framework for long-form video understanding. Motivated by *semantic thrashing* in previous video agents, where append-only context dilutes early evidence as the trajectory grows, we pair an outer multimodal agent with an inner LLM that rewrites a bounded working memory after every outer step from a sandboxed artifact store. VideoLoop improves accuracy on three long-form video benchmarks, and the gain transfers across multiple LVLM backbones in a plug-and-play manner. Beyond video, we see a general principle for long-horizon agents: memory should be curated, not accumulated.

AI USE STATEMENT

Generative AI is not an incidental writing aid in this work: it is one of the instruments under study. VideoLoop retains model-generated observations and intermediate artifacts in a persistent filesystem, with a step manifest recording the agent's actions. *In the method:* pretrained vision-language models provide the outer-loop policy, video analysis, answer generation, and inner-loop memory rewriting described in Section 3. *In evaluation:* separate language models serve as blind judges for the context-based retrievability tests, with access restrictions and settings given in Appendices E and F. *In writing and engineering:* AI assistance supported manuscript revision, LaTeX preparation, figure and table formatting, numerical and protocol consistency checks, and result-table scripts. We take full responsibility for all text, claims, and artifacts in this work.

ETHICS STATEMENT

VideoLoop is a training-free framework evaluated on existing video-understanding benchmarks. *Data use and privacy:* benchmark videos and associated materials remain subject to their original terms of use. Video frames, speech transcripts, and intermediate artifacts may contain personal information. Applications to new data require appropriate permissions and access controls for both external model APIs and the persistent filesystem. *Code execution:* the agent generates and executes code through a sandbox. Such execution requires isolation from sensitive host files and credentials. *Model reliability:* video analysis and memory rewriting can propagate errors or biases from the underlying models. Evaluation on these benchmarks does not establish reliability for safety-critical applications. Appendix B discusses broader impacts.

REPRODUCIBILITY STATEMENT

Section 3 describes the architecture and tool interfaces, and Section 4 specifies the model backbones, iteration limits, memory budget, and key-frame cap. Appendix D details the datasets and evaluation splits. Appendices H, I, and J provide the algorithm, working-memory schema & example, and orchestrator prompt. Appendix E documents the blind-judge protocol, including snapshot sampling, the judge prompt, and aggregation. Appendices K and L illustrate memory evolution and agent trajectories. The code will be released upon publication.

REFERENCES

Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. Why does the effective context length of llms fall short? In *ICLR*, 2025.

Shubhang Bhatnagar, Renxiong Wang, Kapil Krishnakumar, Adel Ahmadyan, Zhaojiang Lin, Lambert Mathias, Xin Luna Dong, Babak Damavandi, Narendra Ahuja, and Seungwhan Moon. Videomind: Thinking in steps for long video understanding. In *ACL*, 2026.

Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. In *ICCV*, 2025.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. In *ECAI*, 2025.

Peter J. Denning. Thrashing: Its causes and prevention. In *Proceedings of the AFIPS Fall Joint Computer Conference*, 1968a.

Peter J. Denning. The working set model for program behavior. *Communications of the ACM*, 1968b.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *ECCV*, 2024.

Junyu Feng et al. M2a: Multimodal memory agent with dual-layer hybrid memory for long-term personalized interactions. *arXiv:2602.07624*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, 2025.

Hong Gao, Yiming Bao, Xuezhen Tu, Bin Zhong, Linan Yue, and Min-Ling Zhang. Apvr: Hour-level long video understanding with adaptive pivot visual information retrieval. In *AAAI*, 2026.

Gemini Team, Google. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

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

Google DeepMind. Gemini 3 Flash Model Card, 2025. URL https://deepmind.google/models/model-cards/gemini-3-flash/.

Google DeepMind. Gemini 3.1 Pro Model Card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. Framethinker: Learning to think with long videos via multi-turn frame spotlighting. In *ICLR*, 2026.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? In *COLM*, 2024.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv:2501.13826*, 2025a.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *ACL*, 2025b.

Hongbo Jin, Qingyuan Wang, Wenhao Zhang, Yang Liu, and Sijie Cheng. Videomem: Enhancing ultra-long video understanding via adaptive memory management. *arXiv:2512.04540*, 2025.

Jialuo Li, Bin Li, Jiahao Li, and Yan Lu. Divide, then ground: Adapting frame selection to query types for long-form video understanding. *arXiv:2512.04000*, 2025a.

Keliang Li, Yansong Li, Hongze Shen, Mengdi Liu, Hong Chang, and Shiguang Shan. Lenswalk: Agentic video understanding by planning how you see in videos. In *CVPR*, 2026a.

Zhi Li, Yanan Wang, Hao Niu, Julio Vizcarra, and Masato Taya. An empirical study for representations of videos in video question answering via mllms. *arXiv:2510.12299*, 2025b.

Zongyao Li, Kengo Ishida, Satoshi Yamazaki, Xiaotong Ji, and Jianquan Liu. Kfs-bench: Comprehensive evaluation of key frame sampling in long video understanding. In *WACV*, 2026b.

Jingyang Lin, Hang Hua, Ming Chen, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Jiebo Luo. Videoxum: Cross-modal visual and textural summarization of videos. *IEEE Transactions on Multimedia*, 2023.

Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, et al. Unleashing hour-scale video training for long video-language understanding. In *NeurIPS*, 2025.

Jingyang Lin, Jialian Wu, Jiang Liu, Ximeng Sun, Ze Wang, Xiaodong Yu, Jiebo Luo, Zicheng Liu, et al. Videoseek: Long-horizon video agent with tool-guided seeking. In *CVPR*, 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*, 2024.

Runtao Liu, Ziyi Liu, Jiaqi Tang, Yue Ma, Renjie Pi, Jipeng Zhang, and Qifeng Chen. Longvideoagent: Multi-agent reasoning with long videos. *arXiv:2512.20618*, 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. In *ICLR*, 2026.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, et al. Video-rag: Visually-aligned retrieval-augmented long video comprehension. In *NeurIPS*, 2025.

Yongdong Luo, Wang Chen, Weizhong Huang, Shukang Yin, Haojia Lin, Jinfa Huang, Chaoyou Fu, Jiayi Ji, Xiawu Zheng, and Jiebo Luo. Quota: Query-oriented token assignment via cot query decouple for long video comprehension. In *AAAI*, 2026.

Can Lv, Heng Chang, Yuchen Guo, Shengyu Tao, and Shiji Zhou. All-mem: Agentic lifelong memory via dynamic topology evolution. *arXiv:2603.19595*, 2026.

OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2023.

OpenAI. GPT-4o System Card, 2024. URL https://openai.com/index/gpt-4o-system-card/.

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

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *arXiv:2310.08560*, 2023.

Ziqi Pang and Yu-Xiong Wang. Mr. video: "mapreduce" as an effective principle for long video understanding. In *NeurIPS*, 2025.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *ICML*, 2023.

Aniket Rege, Arka Sadhu, Yuliang Li, Kejie Li, Ramya Korlakai Vinayak, Yuning Chai, Yong Jae Lee, and Hyo Jin Kim. Agentic very long video understanding. *arXiv:2601.18157*, 2026.

Kate Sanders, Reno Kriz, David Etter, Hannah Recknor, Alexander Martin, Cameron Carpenter, Jingyang Lin, and Benjamin Van Durme. Grounding partially-defined events in multimodal data. In *EMNLP*, 2024.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *CVPR*, 2025.

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

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *ECCV*, 2024.

Ying Wang, Yanlai Yang, and Mengye Ren. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos. *arXiv:2312.05269*, 2023.

Yun Wang, Long Zhang, Jingren Liu, Jiaqi Yan, Zhanjie Zhang, Jiahao Zheng, Ao Ma, Run Ling, Xun Yang, Dapeng Wu, Xiangyu Chen, and Xuelong Li. Video-em: Event-centric episodic memory for long-form video understanding. *arXiv:2508.09486*, 2025a.

Zikang Wang, Boyu Chen, Zhengrong Yue, Yi Wang, Yu Qiao, Limin Wang, and Yali Wang. Videochat-a1: Thinking with long videos by chain-of-shot reasoning. In *AAAI*, 2026.

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 *CVPR*, 2025b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *NeurIPS Datasets and Benchmarks Track*, 2024.

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

Haiyang Yan, Hongyun Zhou, Peng Xu, Xiaoxue Feng, and Mengyi Liu. Symphony: A cognitively-inspired multi-agent system for long-video understanding. In *CVPR*, 2026.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis L Brown II, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-s: Towards spatial supersensing in video. In *ICLR*, 2025a.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. Vca: Video curious agent for long video understanding. In *ICCV*, 2025b.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *ACM Multimedia*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *CVPR*, 2025.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *CVPR*, 2026.

Yufei Yin, Qianke Meng, Minghao Chen, Jiajun Ding, Zhenwei Shao, and Zhou Yu. Videoarm: Agentic reasoning over hierarchical memory for long-form video understanding. In *CVPR*, 2026.

Rongyi Yu, Chenyuan Duan, and Wentao Zhang. Longvidsearch: An agentic benchmark for multi-hop evidence retrieval planning in long videos. *arXiv:2603.14468*, 2026.

Ce Zhang, Taixi Lu, Md. Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple LLM framework for long-range video question-answering. In *EMNLP*, 2024a.

Lu Zhang, Tiancheng Zhao, Heting Ying, Yibo Ma, and Kyusong Lee. OmAgent: A multi-modal agent framework for complex video understanding with task divide-and-conquer. In *EMNLP*, 2024b.

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 *NeurIPS*, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *AAAI*, 2024.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Peng Cui, Tiannan Wang, Zhenxin Xiao, Yifan Hou, Ryan Cotterell, and Mrinmaya Sachan. RecurrentGPT: Interactive generation of (arbitrarily) long text. *arXiv:2305.13304*, 2023.

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 *NeurIPS*, 2025.