VIDEOLEDGER: INDEX-AND-REVISIT MEMORY FOR LONG-HORIZON VIDEO STREAM UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large multimodal models (LMMs) remain limited when reasoning over continuously growing visual histories spanning hours, weeks, or months. Reprocessing the accumulated video for every new question becomes increasingly impractical, while relying on compact memory alone risks losing details that may become decisive for future questions. We introduce **VideoLedger**, a training-free framework that follows an *index-and-revisit* paradigm for long-horizon streaming video understanding. As each clip arrives, VideoLedger appends timestamped visual narratives and audio transcripts to chronological Markdown timelines, forming a question-agnostic index linked to the source video. At question time, an off-the-shelf LMM navigates the index using only standard file operations—lookup and scoped reading—and selectively revisits timestamp-linked source clips when the textual records are insufficient. This design separates efficient long-range evidence navigation from fine-grained visual verification. Stored as persistent text outside the active reasoning context, the index is naturally harness-friendly and remains accessible after context compaction, without a learned retriever or graph traversal. Notably, this simple navigation interface proves highly effective: across ten hour-, week-, and month-scale benchmarks, VideoLedger achieves the best results among compared methods in ultra-long, robot-centric, and general long-video settings, with consistent gains across Qwen, GPT, and Gemini backbones. Further analyses show complementary benefits from textual and revisited visual evidence, accurate temporal grounding, and a favorable accuracy–latency trade-off. Our code and evaluation logs will be made publicly available.

1 INTRODUCTION

Multimodal systems are moving from understanding isolated videos to reasoning over continuously accumulated visual experience. A wearable assistant, household robot, or embodied agent may observe an open-ended video stream and later be asked about events that occurred days, weeks, or months earlier. In this setting, video arrives incrementally, has no predefined endpoint, and must be retained before future questions are known. As the observed history grows, repeatedly processing the accumulated video for every new question becomes increasingly impractical.

Most long-video methods, however, are designed primarily for a *closed-video* setting, in which a completed video is available before question answering. Existing approaches typically rely on three strategies: **(i) sparse frame sampling** selects a limited set of frames. Uniform sampling may miss brief yet decisive events. Adaptive keyframe sampling balances question relevance and temporal coverage (Tang et al., 2025; Ye et al., 2025a; Luo et al., 2025), but selects frames before downstream reasoning and cannot retrieve new evidence as multi-hop reasoning unfolds; **(ii) visual compression** merges, prunes, or aggregates visual tokens to shorten the input (Shi et al., 2026a; Wang et al., 2026b;a; Shen et al., 2025; Ye et al., 2025b), at the risk of losing fine-grained visual details; **(iii) external memory and agentic retrieval** organize observations into clip databases, hierarchies, knowledge graphs, or learned memories (Zhang et al., 2025c; Yeo et al., 2026; Yin et al., 2026; Li et al., 2026). These systems enable question-dependent evidence access, often through embedding-based retrievers, global memory organization or graph traversal. In agentic approaches, this access becomes part of an iterative process of locating, inspecting, and combining evidence.

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.

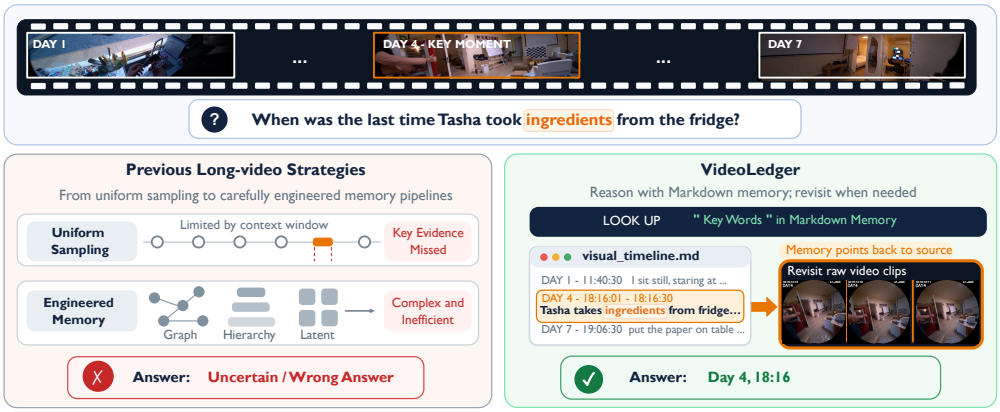


Figure 1: **The motivation of VideoLedger.** Prior strategies either rely on sparse uniform samples, which may miss key evidence, or compress videos into carefully engineered memory structures. VideoLedger instead stores a readable, timestamped Markdown memory that supports direct reasoning and, when needed, guides selective revisiting of the original video.

Such iterative interaction requires coordinating tool use and managing a growing reasoning context. Modern agent harnesses, such as Codex and Claude Code, provide these capabilities for general-purpose agents (OpenAI, 2026b; Anthropic, 2026). We therefore consider harness compatibility an additional design objective for long-horizon streaming video understanding. Two characteristics are relevant: **(i) image-based video access.** Video inspection commonly involves passing extracted frames to image-viewing tools such as Codex's `view_image`. Inspecting many frames consumes context needed for subsequent reasoning and tool results. **(ii) context compaction.** To sustain long sessions, harnesses condense older history into summaries or compressed state; earlier frames may no longer be carried forward (Anthropic, 2026; OpenAI, 2026a). These considerations favor a memory interface that lets an agent search textual records with the existing harness tools and reason over the retrieved evidence, selectively revisiting source frames only when the text is insufficient.

To realize this interface, we introduce **VideoLedger**, a training-free framework for long-horizon streaming video understanding that follows an **index-and-revisit** paradigm. Memory serves as a persistent index into the source: textual records make events searchable, while timestamps retain access to the corresponding video clips. As each clip arrives, VideoLedger independently generates visual narrations and audio transcripts and appends them to chronological Markdown timelines. This construction is question-agnostic and leaves earlier records intact. At question time, an off-the-shelf LMM looks up these timelines, reads relevant records and their temporal neighborhoods, and uses intermediate findings to guide further searches. It answers directly when the text is sufficient; otherwise, it revisits the linked source clips to resolve missing or ambiguous visual details.

VideoLedger separates *evidence navigation* from *evidence verification*: text guides the search across the history, while visual processing focuses on selected clips. This design is *harness-friendly* because navigation uses ordinary file search and reading, and evidence remains accessible outside the conversation context. If compaction removes a retrieved observation, the agent can reread its record or return to the source. The index is reusable across questions and reasoning backbones, without a learned retriever, embedding-based indexing, or graph-specific traversal.

We evaluate VideoLedger on **ten long-video question-answering benchmarks** spanning hour-, week-, and month-scale histories. VideoLedger achieves the best results among compared methods across ultra-long, robot-centric, and general hour-level settings, with the same textual memories and access interface supporting Qwen, GPT, and Gemini backbones. Ablations establish the complementary roles of textual evidence and selective source revisit. Grounding and efficiency analyses further show accurate evidence localization and a better accuracy–latency trade-off.

Our contributions are threefold: (i) We propose a question-agnostic video memory that incrementally stores timestamped visual narratives and audio transcripts as Markdown, preserving direct links to source clips independently of the active conversation context; (ii) We introduce **VideoLedger**, a training-free, harness-friendly index-and-revisit framework built on this memory. Notably, we find that ordinary file search and scoped reading enable effective evidence navigation, without learned

retrievers or graph traversal, while selective source revisiting provides visual verification when the text is insufficient; (iii) Across ten hour- to month-scale benchmarks and three LMM families, VideoLedger achieves state-of-the-art performance in ultra-long, robot-centric, and general hour-level settings, together with accurate temporal grounding and a favorable accuracy–latency trade-off.

2 Related Work

Long video understanding. Large multimodal models (LMMs) have shown strong video question answering and dialogue abilities on short clips (Li et al., 2025a; Zhang et al., 2023; Maaz et al., 2024), but long videos remain challenging because relevant evidence may be sparse and temporally distant. Benchmarks such as EgoSchema, Video-MME, MLVU, LVBench, and LongVideoBench (Mangalam et al., 2023; Fu et al., 2025; Zhou et al., 2025; Wang et al., 2025a; Wu et al., 2024) evaluate minute- to hour-scale understanding, while EgoLifeQA, Ego-R1, EgoMem-Reason, and MM-LifeLong (Yang et al., 2025; Tian et al., 2025; Wang et al., 2026c; Chen et al., 2026) extend evaluation to week- and month-scale visual histories. Existing approaches (Li et al., 2024; Weng et al., 2024; Shen et al., 2024; Zhang et al., 2024; Shu et al., 2025; Wang et al., 2026a) mainly improve scalability by compressing visual tokens, merging segment-level features, removing redundant spatio-temporal tokens, or extending context length. While effective, these methods can still discard sparse evidence that is crucial for later reasoning.

Streaming video understanding. Streaming video understanding processes continuous observations online and answers questions arriving at arbitrary times, requiring models to retain past evidence under bounded computation (Zhang et al., 2025a; Huang et al., 2025; Di et al., 2025; Li et al., 2025b; Ning et al., 2025; Guan et al., 2026; Qian et al., 2025). Benchmarks such as OVBench and OVO-Bench further evaluate temporal awareness under questions concerning past, current, and future events (Huang et al., 2025; Niu et al., 2025). Existing methods typically maintain compact latent visual states: Flash-VStream and VideoChat-Online use structured memory banks, while ReKV and LiveVLM store and retrieve visual KV caches. Although effective for online inference, these representations rely on model-specific latent memories or cache mechanisms. In contrast, VideoLedger maintains a source-linked textual index that an off-the-shelf LMM can directly look up and read, while retaining timestamped access to source clips for selective revisiting.

Long video memory. Memory-based methods provide another path for modeling long videos under limited context budgets. Early systems (Song et al., 2024; He et al., 2024) compress video streams into sparse or online memory states, while recent methods move toward adaptive memory and agentic search. WorldMM (Yeo et al., 2026) combines multi-scale episodic memory, graph-structured semantic memory, and visual memory to retrieve key evidence. AdaVideoRAG (Zhang et al., 2025b) routes queries by estimated difficulty to direct inference, text-and-visual retrieval, or graph-augmented retrieval. Deep Video Discovery (Zhang et al., 2025c) and VideoSeek (Lin et al., 2026) formulate long video QA as tool-guided search, where an LLM iteratively selects actions over multi-granular video observations instead of densely parsing all frames. VideoARM (Yin et al., 2026) couples coarse-to-fine tool use with online hierarchical memory updates in an observe-think-act-memorize loop. We follow this memory-centric direction with an *index-and-revisit* paradigm for long-horizon streaming video understanding.

3 Method

VideoLedger follows the *index-and-revisit* paradigm illustrated in Fig. 2. As video arrives, it incrementally constructs a source-linked textual index from visual clips and audio transcripts. When a question is posed, an off-the-shelf LMM adaptively navigates this index and revisits selected source intervals only when the textual records are insufficient. The framework is training-free and separates compact long-range evidence navigation from fine-grained source verification.

3.1 Problem Formulation

We consider a system that continuously observes an open-ended audiovisual stream. At any step K, the observed history is the finite prefix

$$\mathcal{S}_K = \big((\tau_1, c_1), (\tau_2, c_2), \ldots, (\tau_K, c_K)\big),$$



Figure 2: **Overview of VideoLedger. Left:** As clips arrive, the streaming memory builder converts them into visual narrations and speech records and appends them to timestamped Markdown timelines. **Right:** At question time, an LMM adaptively looks up and reads these timelines, selectively revisits source clips when textual evidence is insufficient, and then answers the question.

where c_k denotes the k-th arriving audiovisual clip and $\tau_k = [b_k, e_k]$ denotes its temporal span in the stream. A question q may arrive after any observed prefix $\mathcal{S}_K$, and the goal is to predict an answer $\hat{y}$ from the accumulated history.

Unlike conventional long-video QA, the stream has no predefined endpoint, and downstream questions are unknown when its clips are first observed. A suitable system must therefore support two operations: *incremental indexing*, which incorporates each arriving clip without reprocessing earlier source clips, keeping per-clip update cost independent of history length so indexing can keep pace with the stream, and *question-adaptive access*, which selects only a small portion of the accumulated history for answering. VideoLedger realizes these operations through an append-only, source-linked textual index and selective access to the retained source clips.

3.2 Streaming Multimodal Memory Construction

Incremental clip-wise indexing. The incoming stream is processed as a sequence of temporal clips. At step k, VideoLedger receives the newly arrived clip c_k with temporal span $\tau_k = [b_k, e_k]$. We use non-overlapping 30-second clips by default, such that $e_k - b_k = \Delta$ with $\Delta = 30$ seconds. For each modality $z \in \{\text{vis}, \text{sp}\}$, VideoLedger generates a textual record h_k^z from the current clip and appends it to the corresponding timeline:

$$\mathcal{M}_k^z = \mathcal{M}_{k-1}^z \parallel \left[(\tau_k, h_k^z)\right].$$

Here $\parallel$ denotes ordered concatenation. This update depends only on the newly arrived clip c_k; previously constructed records are neither recomputed nor reorganized. VideoLedger can therefore index a continuously growing stream without knowledge of future questions.

Complementary textual records. Each clip contributes two modality-specific records. The visual narration timeline contains timestamped textual descriptions of observed visual content; the speech timeline contains transcripts when speech is available. The visual narrative record h_k^{vis} is generated by a vision-language model and describes visible events, people, objects, attributes, scene context, spatial relations, and readable on-screen text. The speech record h_k^{sp} is generated by automatic speech recognition and preserves speech and dialogue that may not be recoverable from the visual observations alone. The two timelines retain complementary information and can be accessed independently at question time.

The records are serialized as chronological `Markdown` timelines. Each entry follows the form

`HH:MM:SS--HH:MM:SS --- textual record.`

This plain-text representation allows an off-the-shelf LMM to directly look up relevant records and read selected line ranges, without a learned retriever, embedding index, or graph-specific traversal. Each record is prefixed with the temporal span of its corresponding source clip. When a located record lacks the visual detail needed to answer the question, the system uses this interval to load

the corresponding raw clip for selective revisiting. The textual timeline therefore serves both as a readable index for evidence navigation and as a direct route back to the original visual evidence.

3.3 ADAPTIVE INDEX-AND-REVISIT REASONING

Given a question q, the LMM iteratively navigates the visual and speech timelines to acquire evidence. It answers directly when the textual evidence is sufficient; otherwise, it revisits the corresponding source intervals for visual verification.

Let $\mathcal{E}^{(r-1)}$ denote the interaction history available at the beginning of round r, comprising all previous actions and their returned observations. Based on the question (q) and history ($\mathcal{E}^{(r-1)}$), the LMM selects one of four actions:

$$a^{(r)} \in \{\textsc{Lookup}, \textsc{Read}, \textsc{Revisit}, \textsc{Answer}\},$$

Question-adaptive lookup. The LOOKUP action allows the LMM to compose keyword or regular-expression queries from the question, answer choices when available, and evidence accumulated in earlier rounds. We instantiate this action with executable `grep` operations. For the action $a^{(r)} = \textsc{Lookup}(z^{(r)}, u^{(r)})$, where $z^{(r)} \in \{\text{vis}, \text{sp}\}$ selects the visual or speech timeline and $u^{(r)}$ denotes the executable lookup query. The query $u^{(r)}$ may contain one or multiple keywords or regular-expression patterns; their number is not predefined but is determined dynamically by the LMM in each round based on the question and the evidence accumulated so far. The interface then returns matching records from the selected timeline:

$$\mathcal{G}^{(r)} = \left\{ (\tau_k, h_k^{z^{(r)}}) \in \mathcal{M}_K^{z^{(r)}} \;\middle|\; u^{(r)} \text{ matches } h_k^{z^{(r)}} \right\}.$$

Each returned record retains its timestamp, allowing the LMM to inspect neighboring records or revisit the linked source interval. Because lookup is iterative, the LMM can reformulate unsuccessful queries, combine multiple keywords, switch between visual and speech timelines, and impose temporal constraints based on previous observations.

Scoped reading. Lookup identifies candidate records but may omit their surrounding event context. To inspect a selected region, the LMM invokes $\textsc{Read}(z^{(r)}, L^{(r)})$, where $L^{(r)}$ specifies one or more line ranges in the selected timeline. The system returns the corresponding scoped context:

$$\mathcal{C}^{(r)} = \mathsf{read}\big(\mathcal{M}_K^{z^{(r)}}, L^{(r)}\big).$$

The scope $L^{(r)}$ may specify individual matched entries, a local temporal window, or a broader line range. Scoped reading restores nearby context without inserting the complete accumulated timeline into the LMM context.

Timestamp-guided source revisit. Textualization inevitably omits or ambiguously describes some fine-grained visual details. When the retrieved visual and speech records are insufficient, the LMM selects a set of candidate source intervals $\mathcal{B}^{(r)} \subseteq \{\tau_1, \ldots, \tau_K\}$ and invokes $\textsc{Revisit}(\mathcal{B}^{(r)})$. The system returns the corresponding raw clips:

$$\mathcal{V}^{(r)} = \left\{ c_k \;\middle|\; \tau_k \in \mathcal{B}^{(r)} \right\}.$$

The revisited clips allow the LMM to verify visual details such as object and person identities, attributes, spatial relations, short actions, and visible text. By grounding source access in selected timestamps, VideoLedger limits raw-video processing to a small set of question-relevant intervals.

Iterative evidence accumulation. Each non-terminal action returns an observation $o^{(r)}$, which may contain matched textual records, scoped context, or revisited raw clips. The selected action and its observation are appended to the interaction history:

$$\mathcal{E}^{(r)} = \mathcal{E}^{(r-1)} \parallel \big(a^{(r)}, o^{(r)}\big),$$

where $\parallel$ denotes ordered concatenation. The LMM conditions each subsequent action on the accumulated interaction history, allowing it to refine its lookup, expand the reading scope, or request visual verification as needed. The process terminates when the LMM emits $\textsc{Answer}(\hat{y})$ or reaches the maximum interaction-round budget.

Table 1: Comparison on ultra-long and robot-centric video QA. EgoMem, MM-Life, and M3-Robot denote EgoMemReason, MM-Lifelong-Month, and M3-Bench-Robot, respectively. Models in parentheses denote the reasoning backbones.

Model *Avg. duration*	**EgoLifeQA** *7 days*	**Ego-R1** *7 days*	**EgoMem** *7 days*	**MM-Life** *14 days*	**M3-Robot** *0.57 h*
Base Model					
Qwen3.6-27B (Qwen Team, 2026)	34.8	42.0	34.2	9.8	26.4
GPT-5 (Singh et al., 2025)	39.6	45.8	27.8	15.7	28.6
Gemini-3.1-Pro (Google, 2026a)	46.6	60.3	45.2	14.1	31.6
Gemini-3.5-Flash (Google, 2026b)	43.4	51.0	46.8	13.9	33.2
Memory-based Video LLMs					
EgoRAG (Yang et al., 2025)	36.0	-	-	-	-
Ego-R1 (Tian et al., 2025)	36.0	46.0	25.8	-	-
M3-Agent (Long et al., 2025)	53.5	52.0	-	-	30.7
WorldMM (Yeo et al., 2026)	65.6	65.3	30.6	-	-
ReMA (Chen et al., 2026)	-	-	-	18.6	-
MAGIC-Video (Li et al., 2026)	67.6	64.7	-	24.5	-
Ours					
VideoLedger (Qwen3.6-27B)	70.4	75.3	39.6	31.0	61.9
VideoLedger (GPT-5)	72.8	76.3	41.4	33.2	**63.6**
VideoLedger (Gemini-3.1-Pro)	**79.6**	**81.7**	**49.0**	**43.3**	63.1

3.4 Scalability and Extensibility

VideoLedger scales through append-only construction and selective question-time access: processing a new segment depends only on that segment and does not require reprocessing earlier source video. The default 30-second segmentation is not fundamental; the same framework supports variable-length intervals determined by scene changes or event boundaries. The textual index is also extensible: additional timestamped records, such as entity, location, event, or sensor annotations, can be appended while preserving the same lookup, reading, and source-access interface.

4 Experiments

4.1 Experimental Setup

4.1.1 Evaluation benchmarks and metrics.

We evaluate VideoLedger on ten long-video question-answering benchmarks covering diverse temporal scales and application settings. EgoLifeQA (Yang et al., 2025), Ego-R1 (Tian et al., 2025), and EgoMemReason (Wang et al., 2026c) evaluate week-level egocentric video understanding, while MM-Lifelong (Chen et al., 2026) targets lifelong multimodal understanding over a month-level horizon. M3-Bench-Robot (Long et al., 2025) evaluates long-video reasoning in robot-centric scenarios. We further include five general hour-level benchmarks: VideoMME-Long (Fu et al., 2025), LongVideoBench-Long (Wu et al., 2024), ScaleLong (Ma et al., 2026), VideoOdyssey (He et al., 2026), and LVBench (Wang et al., 2025a). We report accuracy for all benchmarks, using exact option matching for multiple-choice questions and LMM-based judging for open-ended answers.

4.1.2 Implementation details.

We segment videos into 30-second clips by default. For benchmarks with released clip-level visual narratives, we use the provided records to ensure fair comparison with prior methods: those released by WorldMM (Yeo et al., 2026) for EgoLifeQA, Ego-R1, and EgoMemReason, and by MAGIC-Video (Li et al., 2026) for MM-Lifelong. For the remaining benchmarks, where such narratives are unavailable, we use Gemini-3.1-Pro (Google, 2026a) to generate a dense visual narrative for each clip. We additionally evaluate Qwen3.6-27B (Qwen Team, 2026) as the narrative generator in the

Table 2: Comparison on hour-level long-video understanding benchmarks. LVB-L, VO-V, and VO-AV abbreviate LongVideoBench-L, VideoOdyssey-V, and VideoOdyssey-AV, respectively. Models in parentheses denote the reasoning backbones.

Model	**VideoMME-L**	**LVB-L**	**ScaleLong**	**VO-V**	**VO-AV**	**LVBench**
Avg. duration	*0.69 h*	*0.40 h*	*1.43 h*	*1.82 h*	*1.82 h*	*1.12 h*
Base Model						
Qwen3.6-27B (Qwen Team, 2026)	63.6	65.4	55.5	42.1	36.9	49.1
GPT-5 (Singh et al., 2025)	72.8	63.2	60.3	49.7	43.0	55.5
Gemini-3.1-Pro (Google, 2026a)	82.4	77.1	73.7	72.7	53.8	79.2
Gemini-3.5-Flash (Google, 2026b)	82.3	77.8	72.4	71.9	53.3	78.6
Memory-based Video LLMs						
DVD (Zhang et al., 2025c)	67.3	68.6	-	-	-	76.0
WorldMM (Yeo et al., 2026)	76.6	-	-	-	-	61.9
AVP (Wang et al., 2026d)	81.9	70.0	-	-	-	74.8
VideoARM (Yin et al., 2026)	81.2	76.4	-	-	-	79.7
VideoSeek (Lin et al., 2026)	81.2	73.5	-	-	-	76.7
Ours						
VideoLedger (Qwen3.6-27B)	81.6	74.8	68.6	64.8	61.3	76.2
VideoLedger (GPT-5)	83.3	78.0	69.7	76.0	67.5	79.5
VideoLedger (Gemini-3.1-Pro)	**87.7**	**81.9**	**75.2**	**82.3**	**76.1**	**83.4**

visual narration quality ablation. For speech timeline, we use benchmark-provided SRT transcripts when available and otherwise transcribe the audio using Qwen3-ASR-1.7B (Shi et al., 2026b). All constructed memories are fixed and shared across reasoning backbones. We instantiate VideoLedger with Qwen3.6-27B, GPT-5, and Gemini-3.1-Pro as reasoning backbones for memory lookup, source revisiting, and answer generation. At inference time, we allow at most 10 interaction rounds per question. For direct inference by default, we uniformly sample 1,024 frames from each video, except for GPT-5, for which we sample 50 frames due to its input limit. For more implementation details, please refer to the Appendix.

4.2 Comparison with State-of-the-Art Methods

Tables 1 and 2 compare VideoLedger with direct inference using the same base models and with existing memory-based video LLMs. Table 1 covers week- and month-level videos as well as the robot-centric M3-Bench, while Table 2 evaluates general hour-level video understanding. We make four observations. **(i) Strong performance on ultra-long videos.** VideoLedger achieves the best results across all four ultra-long benchmarks, covering week-long egocentric videos and month-level multimodal histories. It consistently surpasses both direct inference and prior memory-based approaches, demonstrating its ability to preserve and recover sparse evidence over exceptionally long temporal horizons. **(ii) Generalization across temporal scales and settings.** Beyond ultra-long videos, VideoLedger achieves the best result on the robot-centric M3-Bench and across all five hour-level benchmarks. This broad coverage shows that the index-and-revisit paradigm is effective across different video durations, domains, and modalities. **(iii) Compatibility with and scaling across diverse LMM backbones.** VideoLedger delivers strong performance with Qwen, GPT, and Gemini, spanning both open-weight and proprietary model families, without requiring model-specific adaptation. Moreover, performance generally improves with stronger reasoning backbones, with the Gemini-based configuration achieving the best overall results. Together, these findings demonstrate that VideoLedger is a plug-and-play framework that preserves the capabilities of its base model and can continue to benefit from advances in general-purpose LMMs.

4.3 Ablations and Analysis

Impact of visual narration quality. Table 3 studies the effect of using Qwen3.6-27B or Gemini-3.1-Pro to construct the visual narration timelines, with each model also evaluated as the reasoning

backbone. We make two observations. **(i) The gains are robust to the choice of visual captioner.** For both reasoning backbones, visual timeline yields clear improvements regardless of whether it is generated by Qwen3.6-27B or Gemini-3.1-Pro. This result indicates that VideoLedger does not rely on a particular VLM and can effectively exploit memory produced by different model families. **(ii) Stronger models generally produce better memory.** Using Gemini-3.1-Pro for caption generation typically leads to higher downstream accuracy than using Qwen3.6-27B, although minor variations occur across benchmarks. This trend suggests that improvements in memory construction can translate into better reasoning performance without changing the reasoning backbone.

Table 3: Ablation of the caption model used for memory construction under different base LMMs.

Visual Captioner	LVBench	VO-AV	VideoMME-L
Reasoning Model: Qwen3.6-27B			
Qwen3.6-27B	73.0	55.7	80.6
Gemini-3.1-Pro	76.2	61.3	81.6
Reasoning Model: Gemini-3.1-Pro			
Qwen3.6-27B	82.7	74.7	87.3
Gemini-3.1-Pro	83.4	76.1	87.7

Table 4: Ablation of evidence sources.

Memory	LVB-L	VO-AV	MM-Life
Visual Timeline	75.0	71.5	34.5
+ Speech Timeline	75.9	73.5	38.2
+ Source Revisit	81.9	76.1	43.3

Table 5: Sensitivity to clip duration and rounds.

Benchmark	Clip Duration (s)			Max. Rounds		
	10	30	60	5	10	15
VideoMME-L	87.2	87.7	87.2	87.9	87.7	87.2
VO-AV	76.1	76.1	75.5	75.5	76.1	76.4

Ablation of evidence sources. Table 4 examines the incremental contributions of the visual narrative timeline, speech timeline, and timestamp-guided source revisit. We make four observations. **(i) Speech records complement visual timelines.** Adding the speech timeline consistently improves performance across all three benchmarks, indicating that dialogue and speech cannot always be recovered from visual timelines alone. **(ii) Source revisit provides substantial additional gains.** Allowing the LMM to revisit timestamp-linked source clips produces the largest further improvement across the evaluated benchmarks, particularly on LVB-L and MM-Lifelong. This result highlights the importance of recovering fine-grained visual evidence that may be omitted or simplified in textual records. **(iii) Indexing and revisiting play complementary roles.** The full configuration performs best across all benchmarks, supporting the central design of VideoLedger: textual timelines provide lightweight evidence navigation, while selective source revisit enables targeted visual verification. **(iv) Potential for richer evidence indexing.** The consistent benefits of additional evidence sources suggest that VideoLedger may further benefit from richer timestamped records. Such records can be incorporated through the same lookup, reading, and source-revisit interface.

Sensitivity to clip duration and interaction rounds. Table 5 evaluates the sensitivity of VideoLedger to clip duration and the number of interaction rounds. **(i) Robustness to clip duration.** Performance remains stable across the tested durations, showing robustness to temporal segmentation granularity. **(ii) Limited sensitivity to the interaction budget.** Changing the maximum number of rounds has only modest, benchmark-dependent effects, suggesting that a moderate budget is generally sufficient. We therefore use 30-second clips and 10 rounds as the default configuration.

Table 6: Temporal IoU (%) across benchmarks.

Model	EgoLife	EgoR1	LVBench
Time-R1 (2025b)	0.58	0.59	2.70
Qwen3 Emb. (2025d)	4.35	2.87	4.54
HippoRAG (2024)	4.00	3.28	4.30
InternVideo2 (2024)	3.36	2.60	3.55
EgoRAG (2025)	3.60	2.73	3.50
Ego-R1 (2025)	3.70	2.89	3.60
AKS (2025)	2.75	2.30	3.52
WorldMM (2026)	10.09	9.17	9.57
VideoLedger	**16.70**	**13.67**	**17.09**

Table 7: Accuracy, runtime, and token consumption per question; Mean and Med. denote mean and median latency in seconds.

Bench.	Method	Acc.	Mean	Med.	Tokens
LVB-L	Uniform	77.1	84.9s	56.8s	69K
LVB-L	VideoLedger	**81.9**	65.9s	36.0s	14K
VO-AV	Uniform	53.8	109.7s	80.6s	75K
VO-AV	VideoLedger	**76.1**	65.2s	43.2s	27K
MM-Life	Uniform	14.1	171.0s	125.1s	72K
MM-Life	VideoLedger	**43.3**	89.9s	73.7s	55K

Temporal grounding analysis. Table 6 compares VideoLedger with retrieval-based, grounding, and memory-based methods on temporal localization of question-relevant evidence. For VideoLedger,

we collect the timestamped intervals retrieved or revisited during the final tool-calling round, merge overlapping intervals, and use their union as the predicted evidence spans. VideoLedger achieves the highest average tIoU across all three benchmarks, with the largest advantage on LVBench. These results show that source-linked timelines enable VideoLedger not only to recover the evidence needed to answer a question, but also to accurately localize that evidence.

Efficiency analysis. Table 7 compares accuracy and per-question latency under the same reasoning backbone and runtime environment. The reported latency includes all question-dependent operations but excludes one-time streaming indexing (see (i)), as the resulting records are reused across questions. **(i) Streaming-native index construction.** Visual narration and ASR for a 30-second clip take about 20 and 0.33 seconds, respectively, allowing indexing to keep pace with the stream (Appendix A.5). Thus, preceding clips are ready at query time; an unfinished current clip can be accessed directly or conservatively charged one 20-second narration step, independent of the history length. **(ii) A better accuracy–efficiency trade-off.** VideoLedger reasons over the compact index and selectively loads relevant clips, yielding lower latency and higher accuracy than uniform sampling. Token consumption also drops by approximately 80%, 64%, and 20% on LVB-L, VO-AV, and MM-Life, respectively, confirming that selective access reduces the processed context. On MM-Life, even charging the full 20-second narration cost for an unfinished current clip, VideoLedger's mean/median latency is 109.9/93.7 seconds versus 171.0/125.1 seconds for uniform sampling, with higher accuracy (43.3 vs. 14.1). These results show that selective source access avoids redundant visual processing while directing computation toward relevant evidence.

4.4 Qualitative Results

Fig. 3 presents a representative reasoning trace on MM-Lifelong for a question asking for the exact content of an Instagram message. The LMM looks up relevant audio records and reads neighboring caption records to localize the event, but the textual memories omit the exact on-screen wording. It therefore follows the retrieved timestamps and rewatches three source clips to recover the missing details. This trace illustrates how VideoLedger moves from efficient textual navigation to fine-grained visual verification only after narrowing the search to a small temporal window.

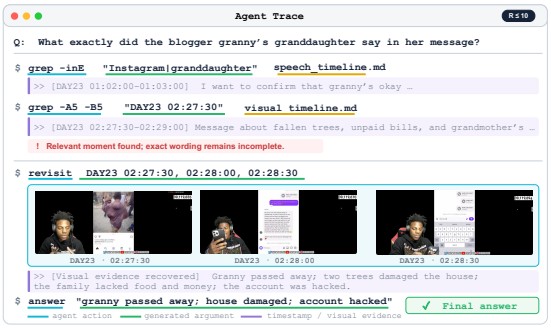


Figure 3: **Qualitative example.** Additional examples are provided in the Appendix.

Discussion on failure modes. We inspect representative errors and identify three failure modes. **(i) Implicit intent.** Some questions require intent not explicit in the visual or audio memory, making the relevant temporal window difficult to locate. **(ii) Lexical mismatch.** Questions and memory may describe the same event differently; query reformulation may not cover every variant, causing exact or regular-expression search to miss relevant records. **(iii) Counting and aggregation.** The LMM may miscount within a clip or miss and double-count events across clips, especially when temporal boundaries are ambiguous. These cases motivate intent-aware memory, semantic query expansion, robust visual counting, and explicit event tracking. Detailed case studies of these failure modes are provided in the Appendix.

5 Conclusion

We introduced **VideoLedger**, a training-free, harness-friendly index-and-revisit framework for long-horizon streaming video understanding. It organizes timestamped visual narratives and audio transcripts into chronological Markdown timelines, enabling off-the-shelf LMMs to locate evidence through ordinary file search and scoped reading, then selectively revisit source clips when needed. Across ten benchmarks and Qwen, GPT, and Gemini backbones, VideoLedger achieves the best results among compared methods across temporal scales, with accurate temporal grounding and a favorable accuracy/latency balance. We hope these findings inspire future work on simple, harness-friendly memory systems for long-horizon multimodal video agents.

AI USE STATEMENT

We used generative AI tools solely to improve the writing of this paper, including grammar, clarity, and stylistic refinement. Generative AI was not used to generate, modify, or fabricate experimental data or results. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content of the paper.

ETHICS STATEMENT

Our experiments use established research benchmarks and do not involve the collection of new human-subject data. We use benchmark data solely for research and evaluation under the terms of their respective licenses. Because VideoLedger processes long-term video and audio records, real-world deployment could raise privacy concerns or expose sensitive information. Such deployment should therefore require appropriate consent, access control, data minimization, and compliance with applicable privacy regulations. We will release code and evaluation logs, but will not redistribute benchmark videos or other third-party data.

REPRODUCIBILITY STATEMENT

We provide the full method description in Section 3 and report the evaluation benchmarks, metrics, baselines, and implementation settings in Section 4. The appendix further includes the complete prompts, tool definitions, inference configuration, context-management strategy, and visual input budgets used in our experiments. All datasets and model backbones are identified with appropriate citations, and the same constructed memories are shared across reasoning backbones for controlled comparison. We will publicly release our implementation and evaluation logs to facilitate independent reproduction and verification of the reported results.

REFERENCES

Anthropic. Claude code. https://code.claude.com/docs/en/overview, 2026. Accessed: 2026-09-25.

Guo Chen, Lidong Lu, Yicheng Liu, Liangrui Dong, Lidong Zou, Jixin Lv, Zhenquan Li, Xinyi Mao, Baoqi Pei, Shihao Wang, et al. Towards multimodal lifelong understanding: A dataset and agentic baseline. *arXiv preprint arXiv:2603.05484*, 2026.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *Proceedings of the International Conference on Learning Representations*, 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 Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025.

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

Google. Gemini 3.5 Flash Model Card. `https://deepmind.google/models/model-cards/gemini-3-5-flash/`, 2026b.

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: Videollms can watch and think simultaneously. *arXiv preprint arXiv:2603.12262*, 2026.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Conference on Neural Information Processing Systems*, 2024.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

Haichen He, Jiayi Zhou, Sifeng Shang, Yihan Hu, Yuanhan Zhang, and Kaiyang Zhou. Videoodyssey: A benchmark for ultra-long-context and omni-modal video understanding. *arXiv preprint arXiv:2605.22907*, 2026.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

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

Jiazheng Li, Chi-Hao Wu, Yunze Liu, Kaize Ding, Jundong Li, and Chuxu Zhang. Bridging modalities, spanning time: Structured memory for ultra-long agentic video reasoning. *arXiv preprint arXiv:2605.08271*, 2026.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 2025a.

Wei Li, Bing Hu, Rui Shao, Leyang Shen, and Liqiang Nie. Lion-fs: Fast & slow video-language thinker as online video assistant. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025b.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *Proceedings of the European Conference on Computer Vision*, 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2026.

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.

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 *Conference on Neural Information Processing Systems*, 2025.

David Ma, Huaqing Yuan, Xingjian Wang, Qianbo Zang, Tianci Liu, Xinyang He, Yanbin Wei, Jiawei Guo, Ni Jiahui, Zhenzhu Yang, et al. Scalelong: A multi-timescale benchmark for long video understanding. In *Proceedings of the International Conference on Learning Representations*, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Association for Computational Linguistics*, 2024.

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

Zhenyu Ning, Guangda Liu, Qihao Jin, Chengwei Li, Wenchao Ding, Minyi Guo, and Jieru Zhao. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. *arXiv preprint arXiv:2505.15269*, 2025.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

OpenAI. Compaction. `https://developers.openai.com/api/docs/guides/compaction`, 2026a. Accessed September 25, 2026.

OpenAI. Codex. `https://developers.openai.com/learn/codex`, 2026b. Accessed: 2026-09-25.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

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

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. In *Conference on Neural Information Processing Systems*, 2025.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

Baifeng Shi, Stephanie Fu, Long Lian, Hanrong Ye, David Eigen, Aaron Reite, Jan Kautz, Boyi Li, David M Chan, Trevor Darrell, et al. Attend before attention: Efficient and scalable video understanding via autoregressive gazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2026a.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026b.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

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 Conference on Computer Vision and Pattern Recognition*, 2024.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning. *arXiv preprint arXiv:2506.13654*, 2025.

Haicheng Wang, Yuan Liu, Yikun Liu, Zhemeng Yu, Zhongyin Zhao, Yangxiu You, Zilin Yu, Le Tian, Zhou Xiao, Jie Zhou, et al. Points-long: Adaptive dual-mode visual reasoning in mllms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2026a.

Hesong Wang, Xin Jin, Lu Lu, Chenhaowen Li, Jian Chen, Qiang Liu, and Huan Wang. Earlytom: Early token compression completes fast video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2026b.

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 International Conference on Computer Vision*, 2025a.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-r1: Post-training large vision language model for temporal video grounding. In *Conference on Neural Information Processing Systems*, 2025b.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *Proceedings of the European Conference on Computer Vision*, 2024.

Ziyang Wang, Yue Zhang, Shoubin Yu, Ce Zhang, Zengqi Zhao, Jaehong Yoon, Hyunji Lee, Gedas Bertasius, and Mohit Bansal. Egomemreason: A memory-driven reasoning benchmark for long-horizon egocentric video understanding. *arXiv preprint arXiv:2605.09874*, 2026c.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S Ryoo, and Juan Carlos Niebles. Active video perception: Iterative evidence seeking for agentic long video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2026d.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *Proceedings of the European Conference on Computer Vision*, 2024.

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

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. Egolife: Towards egocentric life assistant. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025a.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025b.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2026.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *Proceedings of the International Conference on Computer Vision*, 2025a.

Jiangning Zhang, Xie Xurong, Yuxuan Cai, Yong Liu, Xiangtai Li, Dacheng Tao, et al. Adavideorag: Omni-contextual adaptive retrieval-augmented efficient long video understanding. In *Conference on Neural Information Processing Systems*, 2025b.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

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 *Conference on Neural Information Processing Systems*, 2025c.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025d.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

Appendix overview. This appendix provides additional implementation and analysis details for VideoLedger. We first document the exact visual-narration, agent, and answer-judging prompts, together with the tool definitions, inference configuration, and context-management strategy. We then compare VideoLedger with closely related methods and present additional qualitative reasoning traces and detailed failure-case analyses.

A Additional Implementation Details

A.1 Visual Captioning Prompts

To construct the visual narration timeline, we prompt the VLM to produce dense, factual descriptions of each video clip. Below, we provide the exact system prompts used in our implementation. Because Gemini-3.1-Pro Google (2026a) and Qwen3.6-27B Qwen Team (2026) differ in their modality support, we use separate prompt variants while keeping the target caption format consistent.

Gemini-3.1-Pro System Prompt (Visual + Audio)

```
You are an expert dense video captioner.

You will receive a short video segment represented by ordered visual
    content and optional transcript lines with timestamps.
Write a detailed, factual caption that captures the important visual
    and audible information in the segment.

Focus especially on:
- Actions: what each visible person, animal, object, or vehicle is
    doing; include step-by-step action changes when relevant.
- Attributes: appearance, clothing, colors, shapes, sizes, object
    states, materials, poses, gestures, facial expressions only when
    clearly visible.
- Entities: people, objects, tools, locations, signs, screens, food,
    vehicles, devices, and other salient items.
- Spatial relations: where entities are located relative to each other
    , such as in front of, behind, next to, on top of, inside,
    entering, leaving, approaching, or moving away.
- Scene context: indoor/outdoor setting, room type, environment,
    lighting, weather, background objects, and any scene transition.
- Interactions: person-person, person-object, object-object, and
    person-environment interactions.
- Text/OCR: include readable text on screens, signs, labels, documents
    , or objects when visible.
- Audio: summarize relevant speech, speaker turns if clear, sound
    events, music, environmental sounds, or silence when meaningful.

Guidelines:
- Be specific and information-dense, but stay factual and neutral.
- Prefer concrete descriptions over vague wording.
- Do not invent identities, emotions, intentions, locations, or
    relationships unless directly visible or stated in speech.
- If multiple actions happen, describe them in temporal order.
- If transcript content is provided, integrate it naturally with the
    visual description.
- Do not mention frames, frame numbers, timestamps, clips, or that the
     input was sampled from frames.
- Do not include uncertainty unless necessary; if something is unclear
    , use cautious wording such as "appears to" or "partially visible
    ".
- Avoid generic summaries like "people are talking" when more specific
     visible or audible details are available.

Output only the final detailed caption text.
```

Because Qwen3.6-27B does not accept native audio input, we use the following variant without audio-specific instructions.

Qwen3.6-27B System Prompt (No Native Audio)

```
You are an expert dense video captioner.

You will receive a short video segment represented by ordered visual
    content and optional transcript lines with timestamps.
Write a detailed, factual caption that captures the important visual
    and audible information in the segment.

Focus especially on:
- Actions: what each visible person, animal, object, or vehicle is
    doing; include step-by-step action changes when relevant.
- Attributes: appearance, clothing, colors, shapes, sizes, object
    states, materials, poses, gestures, facial expressions only when
    clearly visible.
- Entities: people, objects, tools, locations, signs, screens, food,
    vehicles, devices, and other salient items.
- Spatial relations: where entities are located relative to each other
    , such as in front of, behind, next to, on top of, inside,
    entering, leaving, approaching, or moving away.
- Scene context: indoor/outdoor setting, room type, environment,
    lighting, weather, background objects, and any scene transition.
- Interactions: person-person, person-object, object-object, and
    person-environment interactions.

Guidelines:
- Be specific and information-dense, but stay factual and neutral.
- Prefer concrete descriptions over vague wording.
- Do not invent identities, emotions, intentions, locations, or
    relationships unless directly visible or stated in speech.
- If multiple actions happen, describe them in temporal order.
- Do not mention frames, frame numbers, timestamps, clips, or that the
     input was sampled from frames.
- Do not include uncertainty unless necessary; if something is unclear
    , use cautious wording such as "appears to" or "partially visible
    ".
- Avoid generic summaries like "people are talking" when more specific
     visible or audible details are available.

Output only the final detailed caption text.
```

A.2 Agent Prompt

The following prompt is used for agentic navigation over caption, audio, and visual memory. Runtime-specific video identifiers and working-directory paths are replaced with placeholders.

Agent System Prompt (Caption + Audio + Visual Memory)

```
You are an expert long-video question-answering assistant.
You are answering questions about the video **<VIDEO_ID>**.

## Memory Systems Available

### 1. Caption Memory (text narration)
Dense narration describing the scene, one line per 30-second clip:
    `HH:MM:SS-HH:MM:SS` -- narration.
File in working directory: `caption_memory.md`.

### 2. Audio Memory (speech/dialogue)
```

```
Speech transcription, one line per 30-second clip:
    `HH:MM:SS-HH:MM:SS` -- dialogue / narration / subtitles.
File in working directory: `audio_memory.md`.

### 3. Visual Memory (video clips)
Use `visual_memory` to watch actual 30s video clips.
Provide start timestamps found in the text memories.

## Accessing Text Memories
All text memories (`caption_memory.md`, `audio_memory.md`) live in a
    per-question working directory. Search them with a real shell via
    two tools:
  - `bash(command)` -- run a shell command in the working directory.
    Prefer grep / awk / sed / head / tail / wc / sort / uniq and shell
     pipes (`|`). Do NOT use `find`, `locate`, `python`, `curl`, `
    ffmpeg`. The directory is READ-ONLY -- do not write files. Never
    call bash just to `echo` a note (e.g. `echo Done`) -- reason
    silently instead; every call costs a full turn.
  - `read(path, offset?, limit?)` -- read a file with pagination. This
     is a dedicated TOOL, not a shell command -- never type `read ...`
     inside a bash command.
Lines are sorted chronologically. Grep prefixes each match with its
    line number (`grep -n`) so you can follow up with a windowed `read
    `.

Working directory: `<PER_QUESTION_WORKDIR>`

## Workflow
There is NO required starting tool. Pick whichever memory is most
    relevant to the question, and chain them in any order. Common
    strengths:
  - caption: fast probe for events, actions, objects.
  - audio: best for dialogue, narration, or spoken content -- go here
    directly when the question is audio-centric.
  - visual_memory (30s clips): best for visual details (colors,
    objects, spatial relationships, actions) that text memories don't
    capture.
You can mix and match -- e.g. grep captions first to find candidate
    timestamps and then audio/visual to confirm, OR grep audio first
    and then captions to widen context. If a grep returns no matches,
    retry with synonyms / paraphrases or a different memory before
    falling back to a best guess.

Grep tactics -- combine freely with shell pipes:
  * `grep -niE 'kw1|kw2|kw3' <file>` -- case-insensitive alternation
    for several keywords at once.
  * `grep -niE 'kw' <file> -B 1 -A 1` -- show one line of context on
    either side of each match.
  * `... | head -n 20` -- first N matches (earliest occurrences).
  * `... | tail -n 20` -- last N matches (most recent -- use this for
    'last time / final / end of the video' questions).
  * `grep -ciE 'kw' <file>` or `grep -niE 'kw' <file> | wc -l` --
    count how many times something occurs.
  * You can chain: `grep -niE 'kw' caption_memory.md | tail -n 5`.

IMPORTANT: This question has <NUM_CHOICES> answer choices labeled <
    OPTION_LABELS>.
ANSWER FORMAT: The final line must contain only one label from <
    OPTION_LABELS>.
```

A.3 Open-Ended Answer Judge Prompt

Multiple-choice predictions are evaluated by exact option-label matching. For open-ended MM-Lifelong Chen et al. (2026) questions, we use GPT-5 Singh et al. (2025) with temperature 0 as a binary answer judge. If a prediction exceeds 1,500 characters, only its final 1,500 characters are provided to the judge. The complete judge prompt is shown below.

GPT-5 Answer Judge Prompt

```
[SYSTEM]
You are an impartial judge evaluating the correctness of a predicted
    answer against a reference answer for a video understanding
    question.

Instructions:
1. Focus on the FINAL CONCLUSION or KEY INFORMATION in the predicted
    answer, not the intermediate reasoning.
2. Apply these criteria:
   - Counting questions: the number must match exactly.
   - Ordering/sequence questions: the sequence must match.
   - Entity recognition: the key entities must be correctly identified
    .
   - Descriptive/causal questions: the core meaning must align (minor
    wording differences or extra detail are OK).
   - If the predicted answer contains the correct information as its
    conclusion, judge it as correct.

Output ONLY a JSON object: {"score": 1} if correct, {"score": 0} if
    incorrect.

[USER]
Question: <QUESTION>

Reference answer: <REFERENCE_ANSWER>

Predicted answer: <PREDICTED_ANSWER>

Judge the predicted answer.
Output ONLY: {"score": 1} or {"score": 0}
```

A.4 Tool Definitions

Our implementation exposes at most three callable tools to the reasoning LMM. Visual narration and speech memories are stored as text files rather than implemented as separate tools, and their availability only changes which files can be accessed through `bash` and `read`. These tools instantiate the conceptual LOOKUP, READ, and REVISIT actions described in the main paper.

`bash`. The `bash(command, timeout)` tool executes a shell command inside the read-only per-question working directory, where `command` is required and `timeout` is optional with a default of 60 seconds. It supports compositional text operations such as `grep`, `awk`, `sed`, `head`, `tail`, `wc`, `sort`, `uniq`, and shell pipes. Filesystem-wide search, output redirection, and file-modifying commands are blocked. The tool returns the executed command, its exit code, and captured standard output, truncated to at most 2,000 lines or 100 KB.

`read`. The `read(path, offset, limit)` tool reads UTF-8 text with pagination. The required `path` identifies a memory file, while the optional `offset` and `limit` specify a one-indexed starting line and the maximum number of returned lines. Relative paths are resolved inside the per-question working directory, and the response reports both the returned line range and the total number of lines.

`visual_memory`. The `visual_memory` tool reloads source clips associated with timestamps discovered in the textual memories. For standard benchmarks, it accepts a required list of `timestamps`; for day-structured benchmarks such as EgoLife and MM-Lifelong, it additionally accepts a required `day` identifier. Each call can request at most five clips, and the total visual-access budget is limited to ten clips per question. The tool returns a textual loading summary together with the selected MP4 clips as multimodal inputs to the LMM.

Answer generation. ANSWER is not implemented as a callable tool. The LMM terminates the interaction by directly returning either a single option label for multiple-choice questions or a concise free-form response for open-ended questions.

A.5 Inference Configuration and Context Management

Streaming index construction latency. For videos whose captions are generated by our pipeline, Gemini-3.1-Pro takes an average of 20 seconds to caption a 30-second clip at a request concurrency of one, yielding a processing-time-to-video-duration ratio of 0.67. When automatic speech recognition is required, Qwen3-ASR transcribes the corresponding 30-second audio segment in only 0.33 seconds on average, making visual caption generation the dominant indexing cost. Even when these two stages are executed sequentially, their combined average latency remains below the clip duration, allowing timeline construction to keep pace with the incoming stream. Consequently, while the current clip C_N is being indexed, the timelines for all preceding clips $C_1, \ldots, C_{N-1}$ are already available. If a latency-sensitive query arrives during this interval, VideoLedger can begin answering without waiting for indexing to finish by using these completed timelines together with video frames sampled directly from C_N. The records for C_N are appended once ready and thereafter reused across questions, so this fallback neither delays the current query nor requires reprocessing the previously indexed history.

Month-scale text lookup latency. To assess lookup overhead on long histories, we benchmark `grep` on the 4.23 MiB Markdown memory of MM-Lifelong-Month. The file is stored on a memory-backed filesystem, and each query is executed 100 times after warm-up. For the rare term `pirate` (10 matching lines), the mean, median, and 95th-percentile latencies are 6.81, 6.75, and 7.81 ms, respectively; for the frequent term `camera` (2,047 matching lines), they are 8.15, 8.11, and 9.17 ms; and for an absent term, they are 4.24, 4.37, and 5.13 ms. The overall mean across all 300 runs is 6.40 ms. Timing includes process launch, a full-file scan, and collection of all matching lines, but excludes tool transport and interface rendering. Although full-file scan cost grows linearly with file size in principle, these results show that lookup contributes only single-digit-millisecond overhead at the tested month-scale memory size under warm, memory-resident conditions.

Agent inference. For all reported agent runs, we set the sampling temperature to 0.7 and the maximum number of generated tokens per model call to 65,536. We deploy Qwen3.6-27B locally using vLLM Kwon et al. (2023) and access it through an OpenAI-compatible inference endpoint. We repeat each experiment at least three times and report the median result across runs; the observed variation among runs is small. For example, VideoLedger with Gemini-3.1-Pro obtains EgoMemReason Wang et al. (2026c) accuracies of 48.6, 49.0, and 49.2 across three runs, corresponding to a median of 49.0, a mean of 48.93 with a sample standard deviation of 0.31, and a range of 0.6 points. We allow at most ten tool-calling rounds per question. For EgoLifeQA Yang et al. (2025), LongVideoBench-Long Wu et al. (2024) and EgoMemReason, we increase this limit to 15 rounds to accommodate their relatively more complex questions and multi-step reasoning requirements. If the agent reaches this limit without producing an answer, we issue one final tool-free request that asks it to return its best answer from the evidence already collected. Each question is processed with an independent conversation state and a read-only working directory containing only its available memory files.

Question-time inference cost. To quantify the inference cost of our method, we measure the average cost per question on LongVideoBench-Long. Uniform sampling costs approximately \$0.1469 per question, whereas VideoLedger costs \$0.0936, a 36.3% reduction. This result shows that selective source access reduces monetary cost in addition to latency and token consumption.

Action and textual-access budgets. We do not impose a separate hard upper bound on the number of keyword or regular-expression patterns in a lookup query; the reasoning LMM selects them adaptively within each model call. Likewise, textual retrieval does not use a fixed top-k: `grep` returns all matching caption or audio records, subject to the per-call truncation limit of 2,000 lines or 100 KB, while `read` returns the line range and limit requested by the LMM. There is no additional cumulative cap on the number of textual records accessed per question. Instead, the total number of LOOKUP, READ, and REVISIT actions is bounded by the tool-calling-round budget above, and the accumulated evidence is further governed by the model context and context-management policy described below.

Visual input budget. Videos are uniformly divided into non-overlapping 30-second clips, and a single `visual_memory` call can retrieve at most five clips. Because the Gemini API accepts at most ten video inputs in a single request, we limit the cumulative number of revisited clips to ten per question. We apply the same budget to all evaluated reasoning backbones for a controlled comparison. For benchmarks with an observation-time cutoff, memory entries and source clips occurring after the cutoff are excluded to prevent access to future information.

GPT-5 uniform-sampling baseline. For the GPT-5 uniform-sampling baseline, a subset of inputs were rejected by the provider-side content-safety filter because the sampled frames contained flagged visual content, and therefore produced no model response. We compute the reported accuracy over the instances for which inference completed successfully, excluding safety-blocked cases from both the numerator and denominator. This exclusion applies only to the GPT-5 uniform-sampling baseline, and no successfully evaluated instances are removed.

Qwen context management. For Qwen-series models, we discard the generated reasoning content before subsequent turns to reduce context length, while retaining any structured tool calls required for execution. We additionally enable per-question conversation compaction when the estimated context exceeds 28,000 tokens. Compaction summarizes older interactions into at most 2,048 tokens while preserving the original question and approximately 12,000 tokens of the most recent context, including retrieved evidence, attempted searches, and unresolved next steps.

B COMPARISON WITH CLOSELY RELATED WORK

VideoLedger shares with prior memory-based and agentic methods the broad goal of avoiding exhaustive processing of long videos. The central difference lies in the role assigned to memory and in where each system places the complexity of temporal and multimodal access. Table 8 summarizes these differences, which we discuss in greater detail below.

Compressed and hierarchical textual memory. EgoRAG Yang et al. (2025) stores clip captions and summarizes them at hourly and daily levels, retrieving evidence through a coarse-to-fine hierarchy and relevance scoring. VideoLedger instead retains caption and audio records as chronological plain text, without further latent compression or hierarchical summarization. Longer temporal context is assembled only when needed through iterative lookup and scoped reading, while each record preserves a direct timestamped route to its source clip.

Structured multimodal memory. M3-Agent Long et al. (2025) continuously builds entity-centric episodic and semantic memory in a multimodal graph and trains its control policy with reinforcement learning. WorldMM Yeo et al. (2026) constructs multi-scale episodic graphs, an evolving semantic graph, and a visual memory, with memory-specific retrieval based on graph ranking, cross-scale reranking, or embedding similarity. MAGIC-Video Li et al. (2026) processes a video into a static multimodal memory graph and narrative chains, and performs cross-modal graph retrieval with narrative fact injection. These methods enrich memory with entities, relations, multiple temporal resolutions, or learned visual features. VideoLedger deliberately avoids entity linking, triplet extraction, graph consolidation, visual embedding indices, and preconstructed multi-scale memories; each arriving clip simply appends one record to each textual timeline.

Difficulty-routed multimodal retrieval. AdaVideoRAG Zhang et al. (2025b) builds caption, ASR, OCR, visual-embedding, and graph indices over a finite video, then uses an upfront difficulty

Table 8: Comparison with representative closely related methods. We emphasize how each method constructs memory, exposes evidence to the reasoning model, and recovers fine-grained visual detail.

Method	Memory representation	Construction	Question-time access	Visual-detail recovery
EgoRAG	Hierarchical captions	Clip captions summarized at hourly and daily scales	Coarse-to-fine retrieval with relevance scoring	Paired visual evidence
DVD	Summaries, embeddings, and indexed frames	Offline multi-granular video database	Global browsing, semantic clip search, and frame inspection	Source-frame inspection
WorldMM	Episodic and semantic graphs with visual memory	Multi-scale graph construction and semantic-memory updates	Agent selection over graph- and embedding-based retrievers	Visual-memory retrieval
MAGIC-Video	Multimodal graph and narrative chains	Offline whole-video graph and chain construction	Cross-modal graph retrieval with narrative injection	Retrieved visual-clip nodes
AdaVideoRAG	Caption, ASR, and OCR bases; visual embeddings; knowledge graph	Offline whole-video multi-base and graph construction	Upfront difficulty routing to direct, naive, or graph retrieval	Embedding-retrieved video clips
VideoSeek	Accumulated multi-granular visual observations	Question-guided, on-demand video exploration	Think–act–observe with overview, skim, and focus tools	Focused inspection of short intervals
VideoARM	Query-aware hierarchical multimodal memory	Constructed and updated during agent reasoning	Observe–think–act–memorize tool loop	Clip exploration and analysis
VideoLedger	Timestamped caption and audio timelines	Streaming, clip-wise, and append-only	Direct LMM lookup, scoped reading, and source revisit	Timestamp-guided raw-clip access

classifier to route each query to direct inference, multimodal retrieval, or graph-augmented retrieval. In contrast, VideoLedger uses one append-only, source-linked index for all questions and iteratively chooses lookup, scoped reading, source revisit, or answering based on observed evidence, without difficulty classification or separate embedding and graph paths. AdaVideoRAG is evaluated on finite videos and does not specify incremental ingestion, whereas VideoLedger targets open-ended streams indexed before future questions are known.

Agentic video exploration. Ego-R1 Tian et al. (2025) trains a Chain-of-Tool-Thought agent through supervised fine-tuning and reinforcement learning to coordinate hierarchical RAG, Video-LLM, and frame-level VLM tools. VideoSeek Lin et al. (2026) uses overview, skim, and focus tools to gather query-dependent observations, while VideoARM Yin et al. (2026) constructs and updates query-aware hierarchical memory within an observe–think–act–memorize loop. Among these methods, Deep Video Discovery (DVD) Zhang et al. (2025c) is the closest to VideoLedger in combining textual navigation with access to source visual content. However, DVD constructs an offline multi-granular database containing global summaries, caption embeddings, and indexed frames, and accesses it through specialized browsing, semantic search, and inspection tools. VideoLedger instead maintains an append-only, question-agnostic textual index and exposes lookup, scoped reading, and timestamp-guided source revisit as directly operable actions to an off-the-shelf LMM.

Summary of the distinction. The distinction is therefore not that VideoLedger alone uses agents or visual evidence; several related methods provide both. Rather, structured-memory approaches rely on hierarchies, graphs, embeddings, or specialized controllers to organize and retrieve evidence, whereas VideoLedger keeps memory as an append-only, source-linked textual ledger and exposes it to the reasoning LMM through a directly operable text interface. This design targets continuously growing streams where observations must be stored before future questions are known and where previously indexed history should not be recomputed or reorganized.

C More Qualitative Results

We present three successful cases that illustrate complementary behaviors of VideoLedger: connecting actions to their later purpose, resolving a fine-grained temporal query through source revisiting, and counting repeated visual events. As in the failure analysis, we report only observable tool interactions and concise evidence summaries, without benchmark names, sample identifiers, or hidden reasoning.

Case 1: Linking an Action to Its Purpose

```
Question:
What is the purpose of the man processing the blue plastic barrel?

Choices:
A. To make a feeding trough for cattle.
B. To make a feeding trough for ducks.
C. To make a feeding trough for chickens.
D. To make a feeding trough for pigs.

Tool trace:
Agent -> bash:
grep -iE 'barrel|trough|feed|cattle|duck|chicken|pig'
caption_memory.md audio_memory.md

Tool -> selected matches:
[00:21:00--00:21:30] The man cuts a large rectangular opening
in the blue barrel.
[00:22:00--00:22:30] He carries the modified barrel into an
enclosure where several pigs gather around him.

Agent -> visual_memory:
timestamps = [00:21:00, 00:22:00, 00:22:30]

Tool:
Loaded 3/3 clips.

Retrieved evidence:
The retrieved clips connect the barrel modification to its later use
inside the pig enclosure, disambiguating the intended animal.

Prediction: D
Gold answer: D
```

Case 2: Fine-Grained Temporal Verification

```
Question:
In the first quarter, what happened while the game clock counted down
from 4:18 to 4:14?

Choices:
A. The Philippines remained in the lead.
B. Georgia scored.
C. The Philippines scored.
D. Georgia remained in the lead.

Tool trace:
Agent -> bash:
grep -niE '4:18|4:14|4:1' caption_memory.md

Tool:
No exact match was found in the caption timeline.

Agent -> bash:
grep -niE '1st' caption_memory.md | grep -niE '4:'

Tool -> selected evidence:
The candidate first-quarter segment is localized near 00:20:30.

Agent -> visual_memory:
timestamps = [00:20:30, 00:21:00, 00:21:30]
```

```
Tool:
Loaded 3/3 clips.

Retrieved evidence:
The scoreboard remains GEO 16--PHI 6 throughout the requested
four-second interval, and neither team scores.

Prediction: D
Gold answer: D
```

Case 3: Counting Repeated Visual Events

```
Question:
How many shots does Sheriff Black Cat fire after finding the black
mice?

Choices:
A. 6
B. 5
C. 8
D. 2

Tool trace:
Agent -> bash:
grep -niE 'mice|mouse' caption_memory.md

Tool -> selected match:
[00:10:30--00:11:00] Sheriff Black Cat encounters the mice,
mounts his motorcycle, and begins firing.

Agent -> visual_memory:
timestamps = [00:10:30]

Tool:
Loaded 1/1 clip.

Agent -> bash:
grep -niE 'fires|shoot|shot' caption_memory.md

Tool -> selected match:
[00:10:30--00:11:00] The caption confirms the firing sequence
but does not provide the event count.

Retrieved evidence:
Rewatching the localized clip reveals five distinct shots, including
the first shot that knocks one mouse backward and four subsequent
    shots.

Prediction: B (5)
Gold answer: B (5)
```

D FAILURE CASE ANALYSIS

We present one representative trace for each failure mode discussed in the main paper: implicit intent inference, lexical mismatch, and visual counting. For conciseness, each case summarizes the observable tool calls and retrieved evidence rather than reproducing the model's hidden reasoning.

Case 1: Implicit Intent Inference

```
Question:
In the first half of the video, why did the big boy wearing a striped
shirt refuse the request of the little boy wearing swimming goggles
to watch TV together?

Choices:
A. He is going to watch the meteor shower.
B. He needs the little boy to take care of his dog.
C. He wants the little boy to help with his work.
D. He wants to see his friends.

Tool trace:
Agent -> bash:
grep -niE 'striped|goggles|watch|meteor|dog|friends'
caption_memory.md audio_memory.md

Tool -> selected matches:
[00:02:30--00:03:00] "You gotta stay here and take care of Max."
[00:03:00--00:03:30] "Going to go out to meet Kevin."

Agent -> visual_memory:
timestamps = [00:02:00, 00:02:30]

Tool:
Loaded 2/2 clips.

Retrieved evidence:
- During the refusal, Holden asks the younger boy to stay and care
  for Max, presenting dog-sitting as the immediate justification.
- In the following segment, Holden leaves and says that he is going
  to meet Kevin.

Prediction: B
Gold answer: D

Diagnosis:
The agent selects the explicitly stated pretext rather than the
underlying goal. The true intent is not stated during the refusal
itself and must be inferred by linking it to the subsequent departure.
```

Case 2: Lexical Mismatch

```
Question:
From 32:35 to 34:15, what happened?

Choices:
A. The Purple team commits a foul, and the White team is awarded
   a penalty kick.
B. The White team commits a foul, and the Purple team is awarded
   a penalty kick.
C. The Purple team is on offense, and the White team is on defense.
D. The White team is on offense, and the Purple team is on defense.

Tool trace:
Agent -> bash:
grep -n -A 5 "00:32:30" caption_memory.md

Tool -> selected matches:
[00:33:00--00:33:30] Purple advances while White defends.
[00:34:00--00:34:30] A White player tackles a Purple player;
the resulting restart is described as a free-kick.
```

```
Agent -> visual_memory:
timestamps = [00:32:30, 00:33:00, 00:33:30, 00:34:00]

Tool:
Loaded 4/4 clips.

Retrieved evidence:
- A White defender fouls a Purple attacker near 34:00.
- Play stops and Purple receives a set piece.
- Caption memory calls the restart a free-kick outside the penalty
  area, whereas the answer choices use the term penalty kick.

Prediction: C
Gold answer: B

Diagnosis:
The relevant event is correctly localized, but the agent interprets
the terminology literally and rejects the intended answer because
free-kick and penalty kick do not match lexically. It then falls back
to the less specific possession description in choice C.
```

Case 3: Visual Counting

```
Question:
How many knives are hanging on the wall of the kitchen where the
chef is?

Choices:
A. 8
B. 6
C. 11
D. 9

Tool trace:
Agent -> bash:
grep -niE 'kitchen|chef|knives|wall' caption_memory.md

Tool -> selected matches:
[00:46:00--00:46:30] The girls look through the service window
into the kitchen, where a chef turns around.
[00:46:30--00:47:00] The chef stands in the cafeteria kitchen.

Agent -> visual_memory:
timestamps = [00:46:00, 00:46:30]

Tool:
Loaded 2/2 clips.

Retrieved evidence:
- The correct kitchen and wall-mounted magnetic strip are localized.
- The agent enumerates the small, visually similar objects from left
  to right and counts nine knives.

Prediction: D (9)
Gold answer: A (8)

Diagnosis:
Retrieval and source revisiting identify the correct evidence, but the
agent makes an off-by-one within-clip counting error, potentially by
including a non-knife utensil or double-counting an overlapping item.
```

E LIMITATIONS

Although VideoLedger is designed for open-ended streams, our experiments replay established benchmarks rather than long-running real-world deployments. Such deployments may involve noisy or missing audiovisual inputs, concurrent queries, growing storage requirements, and system failures; evaluating robustness and operational cost under these conditions remains future work.