WRITE ONCE, QUERY MANY: EVOLVING COMPACT REUSABLE MEMORY FOR LONG VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long videos often exceed the context capacity of multimodal large language models (MLLMs), while agentic search repeats costly video processing and grounding for each query. Reusable memory can amortize this cost, but must preserve information for questions unknown during construction. We propose Video-EMem, an evolvable video memory harness built on time-anchored perceptual observations. This representation combines compact free-form text with sparse images to preserve semantic content and complementary visual details across temporal scales, supporting temporal navigation without a fixed semantic schema. A Writer encodes each video once by consolidating overlapping multiscale observations, and a Reader retrieves and organizes evidence solely from the resulting memory. With the underlying models frozen, we evolve the operators within their writing and reading workflows using distinct feedback: query-independent source fidelity and coverage for the Writer, and QA failures in evidence localization and utilization for the Reader. Experiments support this separation for generalization beyond development queries, with our evolution strategy improving average accuracy by **7.2%** over the unevolved harness. While reducing storage by about **99.86%** on average relative to raw videos, Video-EMem improves Qwen3-VL-30B by an average of 2.6% across three long-video QA benchmarks over direct input of 2,048 frames. Using the same o3 answer model as the state-of-the-art agentic method, it improves average accuracy by 2.5% with $\mathbf{19}\times$ lower per-query latency.

1 INTRODUCTION

Long videos often exceed the practical context capacity of multimodal large language models (MLLMs). Search-based and agentic methods address this limitation by iteratively locating relevant segments (Ye et al., 2025; Fan et al., 2024; Zhang et al., 2026b), but may repeat costly video processing and grounding, as shown in Figure 1 (a). Inspired by external memory in text-based agents (Park et al., 2023; Packer et al., 2023; Wang et al., 2023), an appealing alternative is to encode the video once into a compact representation reusable across queries. This raises a key question: *how can we construct memory under a limited storage budget that preserves video information for unknown future questions and enables efficient evidence retrieval and organization?*

Existing video representations offer different trade-offs. Retaining all visual tokens is costly, while subsampling can discard details and disrupt temporal continuity (Song et al., 2024; He et al., 2024; Zhang et al., 2024b; Yang et al., 2025; Wang et al., 2025b; Qian et al., 2024). Textual memories are compact, but verbalization can omit fine-grained visual evidence (Kahatapitiya et al., 2025; Ma et al., 2025). Predefined schemas or relations for multimodal memory (Fan et al., 2024; Yin et al., 2026; Yeo et al., 2026) may constrain what is retained, and independently processing temporal segments (Zhang et al., 2024a; Pang & Wang, 2025) can overlook broader context. We therefore represent a video using a lightweight memory composed of *time-anchored perceptual observations*. Rather than exhaustively storing all frames or intermediate observations, it selectively retains semantic and perceptual evidence across multiple temporal scales through free-form text and sparse images, without imposing a predefined semantic schema.

To construct and access such memory, we introduce Video-EMem, an evolvable harness with frozen underlying MLLMs. Development queries may cover only a subset of video content, so query-guided optimization (Xu et al., 2026b; Wang et al., 2025c; Diko et al., 2025) risks overlooking

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.

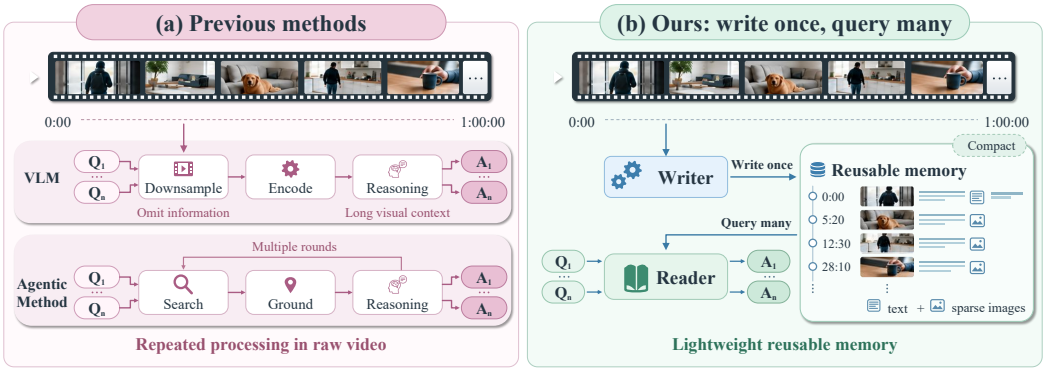

Figure 1: **The need for compact, reusable video memory.** (a) Answering multiple questions about the same video incurs repeated processing costs: VLMs repeatedly process long visual contexts and also often omit details due to context limits, while agentic methods revisit the raw video through multiple retrieval and reasoning rounds. (b) To reduce query and storage costs, we adopt a *write-once, query-many* paradigm, encoding the video once into light-weight memory that preserves information for unknown future questions and supports efficient access across queries.

evidence needed for unseen questions. Our harness therefore separates query-independent writing from query-specific reading. Although prior work adopts this separation (Qian et al., 2024; Kahatapitiya et al., 2025), improving the two procedures under distinct objectives remains less explored. Our *Writer* processes each video once, consolidating observations from overlapping, multiscale windows bottom-up. It compacts text within the token budget, calibrates source-supported information lost in compression, and supplements images while limiting redundancy. Our *Reader* operates solely on the resulting memory: it retrieves text, accesses images for unresolved details, and organizes the evidence for a frozen answer VLM in a single retrieval-and-answer pass.

We leverage LLM-driven agent optimization (Xu et al., 2026c; Zhang et al., 2026a; Ye et al., 2026; Khattab et al., 2024; Opsahl-Ong et al., 2024; Zhang et al., 2025; Agrawal et al., 2026) to evolve the harness with separate feedback for each agent. Optimizing both agents solely for Question-Answering (QA) accuracy risks biasing memory toward development questions. The Writer therefore evolves using only source-video evidence, with feedback that rewards coverage and penalizes redundancy in textual and visual observations. With the resulting memory fixed, the Reader evolves from QA failures attributed to evidence localization or utilization. In both stages, an LLM makes targeted updates to the responsible operator.

We evaluate Video-EMem on hour-long video QA benchmarks with a memory budget of at most 20K text tokens and 160 images per video, reducing storage by about 99.86% on average relative to raw videos. For a one-hour video, this corresponds to ~ 5.6 tokens and ~ 0.04 images per second. Compared with direct video input, Video-EMem improves Qwen3-VL-30B by 1.2%, 4.8%, and 1.9% on LVBench (Wang et al., 2025a), Video-MME Long (Fu et al., 2025), and LongVideoBench Long (Wu et al., 2024), while the baseline using much more context (2048 images). The Writer evolves without QA supervision, while the Reader uses QA feedback only from a small subset of LVBench; both transfer to the other benchmarks without further adaptation. Our evolution strategy improves average accuracy by 7.2% over the unevolved harness. In addition, compared with the state-of-the-art agentic method DVD (Zhang et al., 2026b), Video-EMem improves average accuracy by 2.5 percentage points (72.3% vs. 69.8%) with $\sim 19\times$ lower per-query latency (9.69 s vs. 180 s), using the same OpenAI o3 answer model. Our contributions are as follows:

- A compact, reusable video memory representation based on time-anchored perceptual observations that preserves source information independently of queries and supports efficient evidence retrieval and organization, and reduces storage by $\sim 99.86\%$ relative to raw hour-long videos.
- An evolvable memory harness with separate Writer and Reader agents, optimized independently for source fidelity and QA performance to support generalization to unseen queries, improving average accuracy by 7.2% over the unevolved baseline.
- Experiments demonstrating our improved long-video QA performance and query efficiency ($19\times$ faster than agentic baseline).

2 Related Work

Structured Video and Multimodal Memory. Memory has become a central mechanism for long-video understanding, enabling models to retain long-range evidence beyond their limited context windows and substantially improving performance. Early methods preserve video history through feature banks, recurrent states, or compressed visual tokens (Wu et al., 2022; Song et al., 2024; He et al., 2024; Zhang et al., 2024b; Yang et al., 2025). Subsequent work externalizes video content into language repositories or searchable documents, making long-range evidence easier to retrieve and reason over (Qian et al., 2024; Kahatapitiya et al., 2025; Ma et al., 2025). More recent systems further organize text, images, objects, and events into structured multimodal memories that agents can query or update across temporal scales (Fan et al., 2024; Diko et al., 2025; Yin et al., 2026; Yeo et al., 2026). Despite this progress, memory construction typically follows a fixed architecture with predefined writing, compression, or update rules. Moreover, many repository- and agent-based memories accumulate observations with video duration or reasoning trajectories, while bounded approaches often focus on compressing latent or KV states rather than preserving broadly useful semantic and perceptual information. In contrast, Video-EMem evolves its Writer from source-preservation feedback instead of manually specifying a fixed construction procedure. Under explicit text-token and image-count budgets, the Writer learns to preserve as much source information as possible, producing a lightweight, bounded, and reusable multimodal memory for future queries.

Agentic Long-Video Understanding. Recent long-video understanding systems increasingly employ agentic workflows to locate and organize evidence beyond the context capacity of multimodal language models. Search-based methods progressively narrow the relevant temporal or spatial region: T* Ye et al. (2025) performs temporal-spatial zooming, while DrVideo Ma et al. (2025) iteratively retrieves and augments a textual video document. VideoAgent Fan et al. (2024) equips a language-model agent with localization, memory-querying, and visual-perception tools, and DVD Zhang et al. (2026b) uses agentic reasoning to repeatedly search long videos for question-relevant evidence. Related methods construct query-adaptive representations or retrieve relevant video segments, including R-VLM, VideoTree, and ReWind (Xu et al., 2026b; Wang et al., 2025c; Diko et al., 2025). Although these methods achieve superior performance, they largely overlook the substantial redundant computation involved. Instead of invoking the agent over the raw video for every question, Video-EMem processes the video once into a lightweight, reusable memory, after which a Reader performs query-conditioned evidence access solely over the frozen memory. This write-once, query-many design substantially improves efficiency.

LM-Based Memory and Workflow Evolution. Fixed memory operations and hand-designed LM workflows limit how agents improve from experience. Memory systems such as A-MEM Xu et al. (2026c) reorganize stored content and links, while MemSkill Zhang et al. (2026a) learns and refines reusable memory operations. CoEvo-Mem Ye et al. (2026) further alternates updates to a retrieval policy and memory bank using task outcomes. A related line optimizes prompts, modular LM programs, or code-represented workflows from execution feedback, including DSPy, MIPRO, AFlow, and GEPA Khattab et al. (2024); Opsahl-Ong et al. (2024); Zhang et al. (2025); Agrawal et al. (2026). In this paper, to address the dual challenges of constructing a broadly informative video memory before future questions are known and efficiently accessing relevant evidence from it for each query, we adopt a two-stage evolution process. First, we evolve a query-independent Writer using source-preservation feedback and materialize the resulting video memories. We then freeze these memories and evolve a query-conditioned Reader from attributed downstream QA failures.

3 Method

3.1 Problem Setting

Given a long video V and a set of queries Q, conventional video understanding methods predict an answer $a = A(q, V)$ for each query $q \in Q$, incurring substantial computational costs from repeated visual encoding and long-context agentic reasoning. We instead study a *write-once, query-many* setting: the video is processed once into a compact and lightweight reusable memory M, from which all subsequent queries are answered without revisiting the raw video. To realize this setting, we introduce Video-EMem, consisting of two agents: a *Writer* constructs M from V, while a *Reader* accesses the frozen memory for each downstream query, as illustrated in Figure 2 (a).

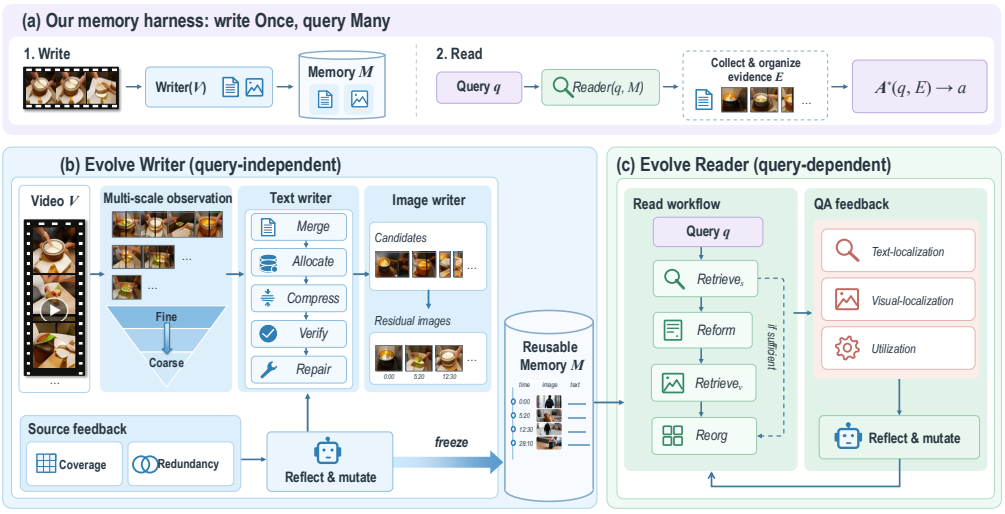

Figure 2: **Method overview.** (a) We present Video-EMem, an evolvable memory harness with a Writer that constructs compact, query-independent video memory and a Reader that retrieves and organizes evidence for question answering. Our memory consists of time-anchored observations combining multiscale textual descriptions with sparse complementary images. We first evolve the Writer (b) using source-preservation feedback without queries, then evolve the Reader (c) from QA failures attributed to evidence localization and utilization.

The key insight behind it is that memory construction and memory access operate under different conditions and therefore need the different optimization goals.

Writer objective. The Writer aims to construct a query-independent memory M under a budget B before observing any downstream queries. Because future information needs are unknown, its objective is to preserve a broad range of source information to support diverse queries:

$$\max_{\text{Writer},\, M} \ \text{Info}(V, M) \qquad \text{subject to} \qquad M = \text{Writer}(V; B), \quad |M| \le B, \tag{1}$$

where $|M|$ denotes memory usage under the prescribed accounting scheme, and $\text{Info}(V, M)$ quantifies the source information retained in M. The construction and evolution procedures introduced below realize this objective through source-preservation diagnostics. Once constructed, M is fixed and reused across all downstream queries.

Reader objective. For each query q, the Reader selects and organizes relevant evidence from the frozen memory M for an answer model $A^\star$. Unlike the Writer, which constructs memory independently of downstream questions, the Reader is optimized directly from question-answer supervision:

$$\min_{\text{Reader}} \ \mathbb{E}_{(M,q,\hat{a})\sim\mathcal{D}_{\text{QA}}} \left[\mathcal{L}_{\text{QA}}(A^\star(q, \text{Reader}(q, M)), \hat{a})\right], \tag{2}$$

where $\mathcal{D}_{\text{QA}}$ denotes the training set of memory-query-answer tuples and $\mathcal{L}_{\text{QA}}$ measures the prediction error with respect to the ground-truth answer $\hat{a}$. During Reader optimization, both the memory M and the answer model $A^\star$ remain fixed.

3.2 Reusable Video Memory Representation

Reusable video memory must balance information fidelity, storage cost, and evidence accessibility. Dense visual tokens are costly to retain for long videos, while compact textual descriptions may omit visual details that are difficult to verbalize.

We therefore adopt a *hybrid multimodal memory representation*: a temporally ordered sequence of time-anchored observations, each comprising a compact textual record and, where needed, sparse images that preserve complementary visual observations within a temporal interval. This representation provides a shared interface for the Writer to encode video content and the Reader to retrieve relevant evidence.

Time-anchored multimodal observations. We represent memory M as a sequence of N observations ordered by their interval start times:

$$M = [o_1, o_2, \ldots, o_N], o_i = (\Delta_i, s_i, \mathcal{I}_i), \quad \text{subject to} \quad \sum_{i=1}^{N} |s_i|_{\text{tok}} \leq B_s, \quad \sum_{i=1}^{N} |\mathcal{I}_i| \leq B_v, \quad (3)$$

Here, $\Delta_i = [t_i^{\text{start}}, t_i^{\text{end}}]$ denotes the video interval, s_i its textual record, and $\mathcal{I}_i$ a sparse or possibly empty set of timestamped images from that interval. The memory budget $B = (B_s, B_v)$ separately limits text tokens and image counts to account for their different encoding costs, with $|s_i|_{\text{tok}}$ denoting the token count of s_i.

The textual record s_i forms the primary memory scaffold, integrating information from multiple temporal scales: fine-grained observations capture local actions and state changes, while coarser observations capture event progression and longer-range context. The visual component $\mathcal{I}_i$ supplements this record with sparse evidence of appearance, spatial layout, and subtle or transient details that are difficult to verbalize. Image timestamps align this evidence with the text, and no images are retained when the textual record is sufficient.

3.3 Writer: Constructing Query-Independent Memory

The Writer first constructs textual observations from a multi-scale source corpus, then supplements them with sparse images that provide complementary visual observations. The writing workflow and evolution are illustrated in Figure 2 (b).

3.3.1 Writing Workflow

In the writing workflow, we first construct textual observations from multi-scale source corpus and then supplement the textual with very sparse images to provide visual observations.

Multi-scale source corpus. To capture fine-grained details and long-range context, we construct a multi-scale caption corpus for the Writer. At each temporal scale $\ell \in \{1, \ldots, L\}$, ordered from coarsest to finest, we divide the video into intervals $\mathcal{T}^\ell$ and independently generate a caption c_{Δ^ℓ} for each interval $\Delta^\ell \in \mathcal{T}^\ell$. These captions provide complementary descriptions at different temporal granularities and serve as the Writer's source corpus.

Compact. The Writer consolidates textual observations from fine to coarse scales. For each interval Δ^ℓ, to produce the final observation z_{Δ^ℓ}, we merges the source caption c_{Δ^ℓ} with the observations $z_{\Delta^{\ell+1}}$ from finer-scale intervals contained within it ($\Delta^{\ell+1} \subseteq \Delta^\ell$), allocates the text-token budget across the merged observations, and compresses each observation to its assigned capacity:

$$z'_{\Delta^\ell} = \mathsf{Merge}\big(c_{\Delta^\ell}, \{z_{\Delta^{\ell+1}}\}_{\Delta^{\ell+1} \subseteq \Delta^\ell}\big), \quad \mathbf{b}_\ell = \mathsf{Allocate}(\mathbf{z}'_\ell; B_s), \quad z''_{\Delta^\ell} = \mathsf{Compress}(z'_{\Delta^\ell}; b_{\Delta^\ell}). \quad (4)$$

Here, $\mathbf{z}'_\ell$ is the temporally ordered sequence of merged observations at scale ℓ, and $\mathbf{b}_\ell$ contains their allocated token budgets, satisfying $\sum_{\Delta^\ell \in \mathcal{T}^\ell} b_{\Delta^\ell} \leq B_s$. At the finest scale, merging uses only the source caption. Budget allocation assigns less capacity to redundant content and more to distinctive information.

Calibrate. Compression may omit or weaken source information. Before the representation is propagated to the next scale, the Writer verifies it against the source sentences before merge:

$$r_{\Delta^\ell} = \mathsf{Verify}(z''_{\Delta^\ell}, c_{\Delta^\ell}, \{z_{\Delta^{\ell+1}}\}_{\Delta^{\ell+1} \subseteq \Delta^\ell}), \qquad z_{\Delta^\ell} = \mathsf{Repair}(z''_{\Delta^\ell}, r_{\Delta^\ell}; b_{\Delta^\ell}), \quad (5)$$

where $r_{\Delta^{\ell+1}}$ records source-supported information that is missing or weakened after compression. Repair selectively restores omissions within the assigned token budget. The calibrated observation z_{Δ^ℓ} is passed to the next coarser scale when $\ell > 1$. At $\ell = 1$, the outputs form the final textual observations $\{s_i\}_{i=1}^N$, where $s_i = z_{\Delta_i^1}$.

Visual supplement. With the textual memory consolidated, the Writer constructs visual observations by selecting images that complement the text while limiting visual redundancy. For each textual observation s_i, we initialize the selected image set $\mathcal{I}_i = \varnothing$ and greedily update it as

$$\mathcal{I}_i \leftarrow \mathcal{I}_i \cup \{I^\star\}, \quad I^\star = \arg\max_I \left[\rho_{\text{MLLM}}(I, s_i) - \lambda \rho_{\text{DINO}}(I, \mathcal{I}_i)\right]. \quad (6)$$

The maximization ranges over unselected candidate images within the interval of s_i, with scores computed before each update. Here, ρ_{MLLM} estimates the information an image adds beyond the

text with an MLLM, and ρ_{DINO} measures its maximum DINO-feature cosine similarity to selected images. The parameter λ controls the redundancy penalty. Selection continues until the allocated image budget is reached. Each selected image retains its timestamp for localization.

3.3.2 WRITER EVOLUTION

We evolve the Writer using source-preservation feedback without downstream QA supervision. Following the memory construction order, we first optimize and freeze text construction, then optimize visual selection to complement the fixed text. Both stages use the mutation strategy summarized below and detailed in Section A.4.

We guide text construction with two diagnostics: coverage and redundancy. (1) The *coverage diagnostic* identifies information lost during construction by comparing the final textual records with the source captions and decomposing omissions into atomic facts. For each video, we retain the construction trajectory with the highest proportion of missing facts, together with those facts, to guide updates to the responsible operators. (2) The *redundancy diagnostic* identifies equivalent, subsumed, or partially overlapping content across the final records. We check temporal locations and source-caption provenance to distinguish redundant descriptions from distinct recurring events. We then retain the groups whose consolidation would save the most tokens, together with their construction trajectories, to guide updates that reduce repetition while preserving distinct evidence.

With the text-writing procedure fixed, we optimize the Writer's image selection to complement the text with visual observations while reducing redundancy. An MLLM classifies each retained image as *complementary*, *text-overlapping*, or *uninformative* relative to its textual observation, while DINO-feature similarity measures redundancy among images. We aggregate these diagnostics to identify the dominant selection failure and guide targeted updates to residual-value scoring, redundancy penalization, or the retention rule. Each update is evaluated using a complementarity score.

3.4 READER: ACCESSING CONSTRUCTED MEMORY

After the Writer evolution, we construct and freeze video memories, then evolve the Reader using memory–question–answer tuples in a training set. The reading workflow and evolution are illustrated in Figure 2 (c).

3.4.1 READING WORKFLOW

Given a query q, the Reader retrieves text, reformulates the query to retrieve images for unresolved details, and reorganizes the evidence into context:

$$\mathcal{E}_s = \mathsf{Retrieve}_s(q, \{s_i\}) \to q' = \mathsf{Reform}(q, \mathcal{E}_s) \to \mathcal{E}_v = \mathsf{Retrieve}_v(q', \{\mathcal{I}_i\}) \to E = \mathsf{Reorg}(q, \mathcal{E}_s, \mathcal{E}_v). \quad (7)$$

If the text is sufficient, Reform and $\mathsf{Retrieve}_v$ are skipped, with $\mathcal{E}_v = \varnothing$. The frozen answer model generates the answer as $a = A^\star(q, E)$.

3.4.2 READER EVOLUTION

We evolve the Reader from failed QA attempts. For each incorrectly answered pair $(q, \hat{a})$, we use the reference temporal interval $\hat{\Delta}$ and the frozen memory to attribute the failure to the responsible operator: (1) *Text-localization failure*: relevant text associated with $\hat{\Delta}$ exists in memory but is missing from $\mathcal{E}_s$, implicating $\mathsf{Retrieve}_s$. (2) *Visual-localization failure*: an image supporting $\hat{a}$ exists in memory but absent from $\mathcal{E}_v$, implicating Reform or $\mathsf{Retrieve}_v$. (3) *Evidence-utilization failure*: retrieved evidence is obscured by redundant, irrelevant, or poorly ordered context, implicating Reorg.

Information absent from memory is not counted as a Reader localization failure. An LLM diagnoses each attributed failure using the operator's execution trajectory, question, prediction, and reference evidence. These diagnoses are aggregated across questions to identify recurring issues and guide revisions in the next round.

Unified workflow mutation. The Writer and Reader share a mutation procedure. Each round, a coding agent uses diagnostic feedback to edit the responsible operator, then evaluates the revision with the stage-specific score while keeping other components and constraints fixed. A sharp score drop triggers rollback to the best previous version. After two rounds of little score change, the agent

Table 1: **Comparison on long-video benchmarks**, reporting accuracy (%) and query time (s). "Avg" is the mean accuracy across the three benchmarks. "Reusable Limits" specifies memory storage budgets. VLMs receive the full stored representation as input, while our method supplies only a query-relevant memory subset to the answer model.

Methods	Reusable Limits	LVBench	Video-MME-L	LongVideoBench-L	Avg.	Time (s)
Frontier VLMs						
Gemini-2.0-Flash	256 images	48.4	63.0	45.7	52.4	–
OpenAI o3	256 images	56.8	64.7	60.6	60.7	12.9
Qwen3-VL-30B	2048 images	61.9	61.6	59.2	60.9	48.7
Agentic Methods						
PACE (Qwen3-VL-30B)	–	61.4	58.9	55.1	58.5	∼3000
MR. Video (gpt-4o)	–	60.2	61.8	61.6	61.2	∼120
DVD (o3)	–	73.5	67.3	68.6	69.8	∼180
ours-Text (Qwen3-VL-30B)	≤20K text	61.3	64.9	59.7	62.0	3.76
ours (Qwen3-VL-30B)	≤20K text + 160 images	63.1	66.4	61.1	63.5	5.51
ours (gpt-4o)	≤20K text + 160 images	63.4	65.5	61.8	63.6	4.42
ours (o3)	≤20K text + 160 images	74.4	72.8	69.7	72.3	9.69

proposes an exploratory edit beyond the latest diagnosis to help escape a local optimum. Evolution stops at the maximum number of rounds; details appear in Appendix Section A.4.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

Benchmarks and metric. We report accuracy (%) on three long-video QA benchmarks. LVBench (Wang et al., 2025a) contains 103 videos averaging 68 minutes. Video-MME (Fu et al., 2025) contains 900 videos and 2,700 questions across diverse domains and durations; we evaluate its 30-60 minutes subset without subtitles, denoted Video-MME-L. LongVideoBench (Wu et al., 2024) contains 3,763 videos and 6,678 human-annotated questions; we evaluate the long-video subset of its validation set, denoted LongVideoBench-L.

Evolution and evaluation splits. We reserve 30 LVBench videos for development, maintaining video-level separation from evaluation. Of these, 20 provide query-independent source feedback for Writer evolution and QA feedback for Reader evolution. We use all the 30 development videos to select the final programs. The other 73 LVBench videos are held out for evaluation. All LVBench results in this paper are evaluated on the same 73 held-out videos. We additionally evaluate on Video-MME-L and LongVideoBench-L without further adaptation to assess cross-benchmark generalization.

Baselines. We compare VIDEO-EMEM with two complementary groups of strong baselines. The first group comprises three state-of-the-art VLMs, Gemini-2.0-Flash, OpenAI o3, and Qwen3-VL-30B, representing direct long-context video understanding. The second comprises three state-of-the-art agentic methods, PACE (Xu et al., 2026a), MR. Video (Pang & Wang, 2025), and DVD (Zhang et al., 2026b), which iteratively access the source video to gather question-relevant evidence. Comparing with the agentic methods, we use the same answer model as each corresponding baseline, thereby isolating differences in memory construction and video access.

Models and budgets. GPT-4.1 serves as the frozen captioner and writer executor, GPT-5.3-codex as the reflector for both evolution stages, and Qwen3-VL-30B-A3B-Instruct as the default frozen answer model. Unless otherwise stated, our memory contains at most 20K text tokens and 160 images stored at 448×448 resolution. Empirically, the resulting memory reduces storage by $\approx 99.86\%$ on average, occupying only 0.14% of the raw-video storage. All model weights and selected programs remain fixed at test time.

4.2 MAIN RESULTS

Table 1 compares Video-EMem with frontier VLMs and agentic methods for long-video understanding. Following the conventional evaluation setting, VLM baselines receive 256 and 2048 sampled

Table 2: **Effect of evolution strategies.** ✓/✗ indicate whether evolution is performed.

Writer	Reader	LVBench	Video-MME-L	LongVideoBench-L	Avg.
✗	✗	54.2	60.3	54.4	56.3
✓	✗	60.3$_{+6.1}$	64.7$_{+4.4}$	59.2$_{+4.8}$	61.4$_{+5.1}$
✗	✓	56.4$_{+2.2}$	61.5$_{+1.2}$	55.6$_{+1.2}$	57.8$_{+1.5}$
Joint QA Evolution		57.5$_{+3.3}$	62.5$_{+2.2}$	55.7$_{+1.3}$	58.6$_{+2.3}$
✓	✓	**63.1**$_{+8.9}$	**66.4**$_{+6.1}$	**61.1**$_{+6.7}$	**63.5**$_{+7.2}$

Table 3: **Effect of the writer operators for text.**

Writer	LVBench	Video-MME-L	LongVideoBench-L
w/o Merge	57.8	62.2	57.5
w/o Calibrate	59.3	63.6	57.8
w/o Allocate	60.5	63.8	59.1
Full	61.3	64.9	59.7

Table 4: **Effect of visual supplement** (VS).

VS	Frames	LVBench	Video-MME-L	LongVideoBench-L
-	-	61.3	64.9	59.7
Uniform	80	61.5	65.3	59.8
Residual	80	62.1	65.8	60.6
Uniform	160	61.7	65.6	60.3
Residual	160	63.1	66.4	61.1

frames. Agentic baselines access the original video through multiple rounds of retrieval and inference, rather than relying on a compressed reusable memory.

Superior performance. With at most 20K text tokens and 160 images per video, Video-EMem achieves the highest accuracy among the compared methods on all three benchmarks: 74.4% on LVBench, 72.8% on Video-MME, and 69.7% on LongVideoBench using OpenAI o3 as the answer model. These results demonstrate the effectiveness of our memory representation across diverse long-video QA benchmarks. Moreover, Video-EMem consistently outperforms prior agentic methods with the same answer model, supporting the contribution of memory construction and reuse beyond improvements in model capability.

Query efficiency. Video-EMem combines strong accuracy with low per-query latency by accessing preconstructed memory and invoking the answer model only once. The Qwen3-VL-30B variant takes 5.51 s per query, compared with approximately 3,000 s for PACE. With OpenAI o3, Video-EMem takes 9.69 s, compared with approximately 180 s for DVD. These results demonstrate that memory reuse supports efficient querying while preserving information needed for accurate answers.

4.3 Ablation Study

Effect of Writer and Reader evolution. Table 2 shows that Writer and Reader evolution provide complementary gains. Evolving only the Writer improves accuracy by 6.1, 4.4, and 4.8 points on LVBench, Video-MME, and LongVideoBench, respectively. Reader evolution adds 2.2, 1.2, and 1.2 points with the fixed Writer, and larger gains of 2.8, 1.7, and 1.9 points with the evolved Writer. This suggests that better memory construction enables more effective query-specific access. Separately evolving both agents achieves the best results, outperforming joint QA evolution by 5.6, 3.9, and 5.4 points and supporting the use of distinct objectives for writing and reading.

Cross-benchmark generalization. The programs evolved on LVBench transfer to Video-MME and LongVideoBench without adaptation, achieving 66.4% and 61.1% accuracy, respectively. These results improve over the unevolved pipeline by 6.1 and 6.7 points, suggesting that the learned memory construction and access strategies generalize beyond the development benchmark.

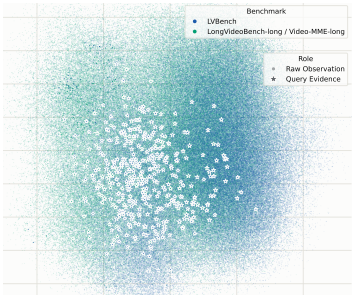

Figure 3: **Query-relevant evidence (white stars) cover only a limited portion of source information (blue and green dots).**

Figure 3 qualitatively illustrates the relationship between source information and evidence relevant to development queries. We project source-caption embeddings from LVBench and other benchmarks, highlighting those associated with the development queries. The highlighted observations represent only part of the source content, motivating source-preservation feedback beyond the observed QA pairs. This projection is illustrative rather than a quantitative measure of information coverage.

Table 2 provides complementary empirical support: separate Writer and Reader evolution consistently outperforms joint QA evolution across all three benchmarks. These results support using

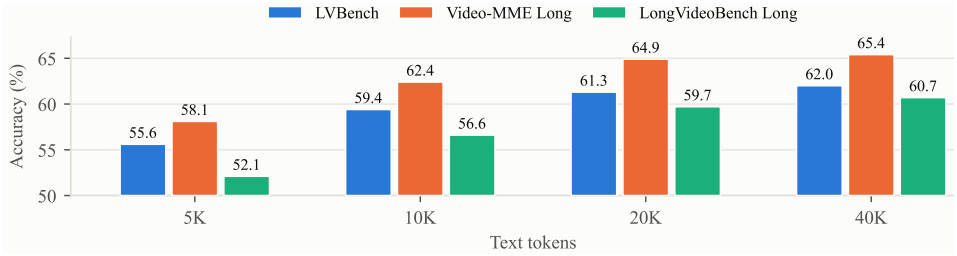

Figure 4: **Effect of text-memory budgets.** Increasing the budget from 5K to 20K tokens substantially improves accuracy, while expanding it to 40K yields only marginal gains. This saturation may partly reflect visual details omitted from source captions, motivating complementary images.

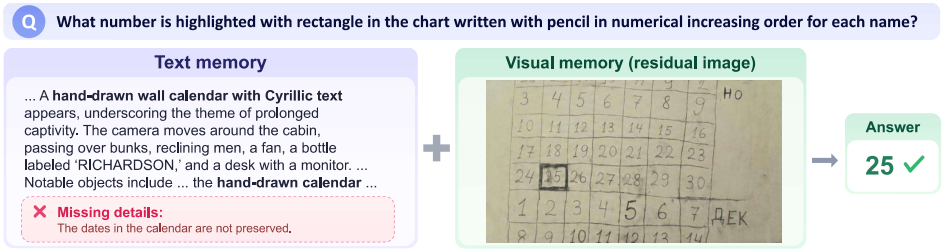

Figure 5: **Residual images preserve visual details omitted from textual memory.** In this example, the textual observations describe the cluttered scene but omit the marked number. The retained image preserves these details, enabling the correct answer.

distinct objectives for memory construction and access, although they do not by themselves establish that the joint strategy overfits development queries.

Effect of Writer operators. We evaluate text-only memory to isolate the contribution of the Writer's textual operators in Table 3. Merge preserves multi-scale source information during hierarchical consolidation, while Allocate assigns capacity according to content importance rather than compressing observations uniformly. Calibrate combines Verify and Repair to detect and restore information lost or distorted during compression. Together, these operators achieve the best balance of coverage, redundancy, and QA accuracy.

Effect of text budget. Figure 4 shows that increasing the text budget from 5K to 20K tokens improves mean accuracy across the three benchmarks by 6.7 points, while doubling it to 40K adds only 0.7 points. These diminishing gains may partly reflect limitations of the captioning model: additional text capacity cannot recover visual details absent from the source captions, motivating complementary visual observations. We therefore use 20K tokens by default to balance accuracy and memory cost, equivalent to approximately 5.6 tokens per second for a one-hour video.

Effect of visual observations. Table 4 shows that residual selection outperforms uniform sampling at the same image budget. With 160 images, it achieves 63.1% accuracy on LVBench versus 61.7% for uniform sampling. Even 80 residual images achieve 62.1%, outperforming 160 uniformly sampled images with half the storage. This demonstrates the benefit of selecting images that complement textual memory. For a one-hour video, 160 images correspond to only about 0.04 frames per second, yet preserve useful visual details omitted from text. In Figure 5, the textual memory describes a hand-drawn calendar but omits its marked numbers; the retained image supplies these details and enables the correct answer.

5 Conclusion

In this paper, we introduced Video-EMem, an evolvable write-once, query-many memory harness for efficient long-video understanding. It encodes each video into compact, time-anchored perceptual observations that combine free-form text with sparse complementary images. With the underlying MLLMs frozen, Video-EMem separately evolves a query-independent Writer using source-preservation feedback and a query-specific Reader using attributed QA failures. Experiments on three long-video QA benchmarks demonstrate improved accuracy, strong cross-benchmark generalization, and substantially lower per-query latency than direct-input and agentic baselines.

AI USE STATEMENT

In this work, generative AI models were used as components of the proposed method and experiments, including video captioning, memory construction, workflow evolution, and answer generation, as detailed in the method and experimental sections. These models were not used to design the research methodology or prepare the manuscript. We also used ChatGPT solely to improve the grammar, clarity, and readability of the manuscript. ChatGPT was not used to generate substantive scientific content, including research hypotheses, methodology, analyses, interpretations, claims, or conclusions. Other tasks listed as requiring disclosure under the ICLR policy were either not performed with generative AI or were not applicable to this work. We take full responsibility for the experimental design, analysis, and final content of the paper.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026.

Anxhelo Diko, Tinghuai Wang, Wassim Swaileh, Shiyan Sun, and Ioannis Patras. Rewind: Understanding long videos with instructed learnable memory. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13734–13743. IEEE, 2025.

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 *European Conference on Computer Vision*, pp. 75–92. Springer, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13504–13514. IEEE, 2024.

Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. Language repository for long video understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5627–5646, 2025.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Saiful Haq, Ashutosh Sharma, Thomas Joshi, Hanna Moazam, Heather Miller, et al. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, volume 2024, pp. 54928–54958, 2024.

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

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, 2024.

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

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

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18221–18232. IEEE, 2024.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967. IEEE, 2025a.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in neural information processing systems*, 36:74530–74543, 2023.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025b.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282. IEEE, 2025c.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13577–13587. IEEE, 2022.

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

Baixuan Xu, Yinyui Xu, Tianshi Zheng, Zhaowei Wang, Weiqi Wang, Haochen Shi, Jiayu Liu, Qing Zong, Xiyu Ren, Xinyu Geng, et al. Finding the right evidence: Factor-guided coarse-to-fine reasoning for long videos. *arXiv preprint arXiv:2608.26355*, 2026a.

Jiaqi Xu, Cuiling Lan, Wenxuan Xie, Xuejin Chen, and Yan Lu. Long video understanding with learnable retrieval in video-language models. *IEEE Transactions on Multimedia*, 2026b.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026c.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. Streammem: Query-agnostic kv cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

Bowen Ye, Yongchao Xu, Zhijian Li, Xiang Yin, Junkai Ma, and Wenzhao Li. Coevo-mem: Co-evolving retrieval policy and memory bank for llm agents. *arXiv preprint arXiv:2608.01739*, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8579–8591. IEEE, 2025.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25599–25609, 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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24042–24051, 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 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21715–21737, 2024a.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024b.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*, 2026a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025.

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. *Advances in Neural Information Processing Systems*, 38:89863–89895, 2026b.

APPENDIX

A ADDITIONAL METHOD DETAILS

A.1 DETAILED VISUAL-SUPPLEMENT PROCEDURE

The visual-supplement stage is applied after the textual records $\{s_i\}_{i=1}^{N}$ have been finalized. Rather than summarizing the video with representative frames, it retains sparse visual evidence that complements the information already preserved in text. The procedure below expands the greedy selection rule in Equation (6), using the memory representation $o_i = (\Delta_i, s_i, \mathcal{I}_i)$ and the global image-count budget B_v defined in Equation (3).

Candidates and selection scores. For each observation o_i, let

$$\mathcal{C}_i = \{(t_{i,j}, I_{i,j})\}_{j=1}^{m_i}, \qquad t_{i,j} \in \Delta_i, \tag{8}$$

denote the candidate timestamp–image pairs from interval Δ_i. The score $\rho_{\mathrm{MLLM}}(I, s_i)$ estimates the residual information that image I provides beyond textual record s_i. Consistent with the main text, we define visual redundancy as

$$\rho_{\mathrm{DINO}}(I, \mathcal{I}_i) = \begin{cases} 0, & \mathcal{I}_i = \varnothing, \\ \max\limits_{(t', I') \in \mathcal{I}_i} \mathrm{sim}(f_{\mathrm{DINO}}(I), f_{\mathrm{DINO}}(I')), & \text{otherwise}, \end{cases} \tag{9}$$

where f_{DINO} is the frozen DINO encoder and $\mathrm{sim}(\cdot, \cdot)$ denotes cosine similarity in its feature space. At each step, the Writer selects

$$(t^\star, I^\star) = \operatorname*{arg\,max}_{(t,I) \in \mathcal{C}_i \setminus \mathcal{I}_i} \left[\rho_{\mathrm{MLLM}}(I, s_i) - \lambda \rho_{\mathrm{DINO}}(I, \mathcal{I}_i)\right], \tag{10}$$

and updates $\mathcal{I}_i \leftarrow \mathcal{I}_i \cup \{(t^\star, I^\star)\}$, subject to $\sum_i |\mathcal{I}_i| \le B_v$. Here, λ controls the penalty for visual redundancy.

A.2 WRITER-EVOLUTION DIAGNOSTICS

Writer evolution follows the same order as memory construction. We first optimize text construction, materialize and freeze the resulting textual records $\{s_i\}_{i=1}^{N}$, and then optimize image selection conditional on those fixed records. The diagnostics below evaluate the output of a candidate Writer; they are not additional construction operators. In particular, diagnostic source coverage is distinct from the construction-time Verify operation in Equation (5).

A.2.1 TEXT-WRITER DIAGNOSTICS

For each final record $s_i = z_{\Delta_i^1}$, we retain the construction trajectory that produced it. A trajectory contains, for every contributing interval $\Delta^\ell \in \mathcal{T}^\ell$, its source caption c_{Δ^ℓ}, merged state z'_{Δ^ℓ}, assigned capacity b_{Δ^ℓ}, compressed state z''_{Δ^ℓ}, verification result r_{Δ^ℓ}, and calibrated state z_{Δ^ℓ}. At each scale, the associated allocation context $\mathbf{z}'_\ell$ and $\mathbf{b}_\ell$ is also retained. These traces connect a loss or redundancy observed in s_i to the decisions of Merge, Allocate, Compress, Verify, and Repair.

Source coverage. Let $\mathcal{S}_i$ be the source-caption sentences that contribute to final record s_i, and let $\mathrm{Sent}(s_i)$ be the sentences retained in that record. We define the diagnostic coverage indicator

$$c(u, s_i) = \mathbf{1}[\exists m \in \mathrm{Sent}(s_i) : \mathrm{Match}(u, m) = 1], \qquad u \in \mathcal{S}_i, \tag{11}$$

where Match is the semantic coverage criterion used by the diagnostic. It tests whether the source-supported information in u remains recoverable from m rather than requiring verbatim overlap. For $|\mathcal{S}_i| > 0$, the record-level coverage is

$$\mathrm{Cov}_i = \frac{1}{|\mathcal{S}_i|} \sum_{u \in \mathcal{S}_i} c(u, s_i), \tag{12}$$

and the text-stage score at evolution round r is

$$g_r^{\text{text}} = \frac{\sum_i \sum_{u \in \mathcal{S}_i} c(u, s_i)}{\sum_i |\mathcal{S}_i|}, \qquad \sum_i |\mathcal{S}_i| > 0. \tag{13}$$

The global score measures source preservation, whereas the record-level values localize where information is lost.

For an uncovered sentence, the diagnostic extracts only atomic facts supported by that source sentence,

$$\mathcal{A}(u) = \text{Decompose}(u), \qquad u \in \mathcal{S}_i, \quad c(u, s_i) = 0. \tag{14}$$

The records with the largest omission rates, their source-supported missing facts, and their construction trajectories form the coverage evidence D_{cov}. This evidence explains what was lost and where the loss arose; it does not replace r_{Δ^ℓ} or participate in construction-time calibration.

Redundancy. Let

$$\mathcal{V} = \bigcup_i \text{Sent}(s_i) \tag{15}$$

be the retained textual sentences for a video. Semantic similarity is used only to propose potentially overlapping pairs in $\mathcal{V}$. The diagnostic then compares their temporal locations and source-caption provenance before classifying them as equivalent, subsumed, or partially overlapping evidence. Semantically similar descriptions of distinct recurring events are therefore kept separate. Confirmed groups are ranked by the token capacity that consolidation would reclaim, and the highest-impact groups and their construction trajectories form the redundancy evidence D_{red}.

Diagnosis and targeted revision. At round r, a bounded subset of D_{cov}, D_{red}, and the associated trajectories forms the phase diagnostic D_r. The coding agent attributes the dominant information loss or capacity waste to exactly one operator in

$$\{\mathsf{Merge}, \mathsf{Allocate}, \mathsf{Compress}, \mathsf{Verify}, \mathsf{Repair}\}. \tag{16}$$

The targeted edit modifies only that operator. All other operators, interfaces, the memory representation, and the text budget B_s remain fixed. The revised program is then evaluated by g_r^{text}, after which the guarded mutation rule in Section A.4 determines the next candidate.

A.2.2 Image-Writer Diagnostics

After text construction is selected, the records $\{s_i\}$ remain frozen throughout image-Writer evolution. For each retained pair $(t, I) \in \mathcal{I}_i$, a fixed multimodal evaluator compares image I with s_i and assigns

$$y(I, s_i) \in \{\text{Complementary}, \text{Text-Overlapping}, \text{Uninformative}\}. \tag{17}$$

A complementary image preserves recognizable evidence not adequately recoverable from s_i; a text-overlapping image mainly repeats the textual record; and an uninformative image provides no sufficiently clear or useful evidence.

Visual redundancy is diagnosed with the same frozen DINO features used by ρ_{DINO}. For each selected image, the diagnostic examines its maximum similarity to the other images in $\mathcal{I}_i$ and treats high-similarity pairs as candidate duplicates. The multimodal evaluator then checks whether those pairs preserve distinct details or unnecessarily repeat the same visual evidence.

Let

$$K = \sum_i |\mathcal{I}_i| \tag{18}$$

be the total number of retained images. When $K > 0$, the diagnostic reports normalized rates

$$R_y = \frac{N_y}{K}, \qquad y \in \{\text{Text-Overlapping}, \text{Uninformative}, \text{Redundant}\}, \tag{19}$$

where N_y is the number of selected image slots assigned to failure type y. When $K = 0$, these rates are not evaluated; instead, the diagnostic records the zero-selection outcome and the corresponding

retention and allocation traces. This avoids treating an empty $\mathcal{I}_i$ as an error when text is sufficient or no image passes retention.

The aggregate labels, redundancy evidence, affected observations, and selection traces form the image-stage diagnostic D_r. The image-stage score g_r^{img} is oriented so that higher values indicate stronger visual complementarity. A targeted revision changes exactly one of the components identified in the main text: residual-value scoring, DINO-based redundancy penalization, or the retention rule. The textual records and all non-targeted selector components remain fixed.

A.3 Reader-Evolution Diagnostics

Reader evolution uses incorrectly answered labeled examples to diagnose failures in evidence access and organization. The memory M, its textual records $\{s_i\}$ and image sets $\{\mathcal{I}_i\}$, and the answer model $A^\star$ remain frozen.

Execution traces. For each incorrectly answered pair $(q, \hat{a})$ with reference temporal interval $\hat{\Delta}$, let $\Xi(q)$ denote the ordered execution log of the reading workflow in the main text. It records:

1. the input query, candidates, rankings, and textual evidence $\mathcal{E}_s$ produced by $\mathsf{Retrieve}_s$;
2. the inputs $(q, \mathcal{E}_s)$, the unresolved information identified by Reform, and its reformulated query q';
3. the image candidates, retrieval decisions, and visual evidence $\mathcal{E}_v$ produced by $\mathsf{Retrieve}_v$, or the decision to skip visual retrieval and set $\mathcal{E}_v = \varnothing$;
4. the inputs $(q, \mathcal{E}_s, \mathcal{E}_v)$ and the organized context E produced by Reorg; and
5. the final prediction $a = A^\star(q, E)$.

The diagnostic judge receives $\Xi(q)$ together with q, a, $\hat{a}$, $\hat{\Delta}$, and the corresponding evidence available in frozen memory.

Failure attribution. The trajectory is assigned one of the three failure types defined in the main text:

1. *Text-localization failure*: relevant text associated with $\hat{\Delta}$ exists in M but is absent from $\mathcal{E}_s$. The responsible operator is $\mathsf{Retrieve}_s$.
2. *Visual-localization failure*: an image supporting $\hat{a}$ exists in M but is absent from $\mathcal{E}_v$. The responsible operator is Reform or $\mathsf{Retrieve}_v$, as determined from the trajectory.
3. *Evidence-utilization failure*: relevant evidence was retrieved, but redundant, irrelevant, or poorly ordered context obscured it in E. The responsible operator is Reorg.

If the evidence needed for $\hat{a}$ was never retained in M, the example is not labeled as a Reader localization failure. After attribution, the judge identifies the specific operator behavior that caused the error. Recurring operator-specific diagnoses and a bounded set of supporting trajectories are aggregated into the round diagnostic D_r. A targeted Reader edit changes only the implicated operator, while the remaining Reader operators, M, and $A^\star$ stay fixed. Reader candidates are evaluated by a higher-is-better QA score g_r^{reader}.

A.4 Guarded Mutation Algorithm

Text-Writer, image-Writer, and Reader evolution use the same guarded program-mutation strategy with phase-specific diagnostics and scores. Let P_r be the workflow program evaluated at round r, D_r its bounded diagnostic feedback, and g_r its evaluation score. All scores are oriented so that higher is better: text-Writer evolution measures source preservation, image-Writer evolution measures visual complementarity, and Reader evolution measures downstream QA performance.

Given the current program and diagnostic context, the coding agent proposes a targeted edit

$$\Delta_r = C(P_r, D_r), \tag{20}$$

where the permitted scope of Δ_r is the component implicated by D_r. All non-targeted components, program interfaces, frozen artifacts, the memory representation, and the budget $B = (B_s, B_v)$ remain unchanged.

A.4.1 BEST-SO-FAR TRACKING

After evaluating the initial program, the best-so-far record is initialized as

$$(P_0^\star, g_0^\star) = (P_0, g_0). \tag{21}$$

For each subsequent round $r \geq 1$, it is updated by

$$(P_r^\star, g_r^\star) = \begin{cases} (P_r, g_r), & g_r > g_{r-1}^\star, \\ (P_{r-1}^\star, g_{r-1}^\star), & \text{otherwise.} \end{cases} \tag{22}$$

Thus, the record changes only when the newly evaluated program achieves a strictly higher score.

A.4.2 MUTATION RULE

The next candidate program is selected according to

$$P_{r+1} = \begin{cases} P_r^\star, & r \geq 1 \text{ and } g_{r-1} - g_r > \tau_{\text{rev}}, \\ \text{Apply}(P_r, \eta_{\text{exp}}), & r \geq 2,\ |g_r - g_{r-1}| < \epsilon, \text{ and } |g_{r-1} - g_{r-2}| < \epsilon, \\ \text{Apply}(P_r, \Delta_r), & \text{otherwise,} \end{cases} \tag{23}$$

where $\tau_{\text{rev}} \geq 0$ is the rollback threshold, $\epsilon > 0$ is the stagnation tolerance, and $\text{Apply}(P, \delta)$ applies edit δ to program P. The rollback branch has priority whenever its condition and the stagnation condition both hold.

Rollback. If the score decreases by more than τ_{rev} relative to the preceding round, the next program is restored to the best-so-far program $P_r^\star$. This prevents a substantially degraded candidate from becoming the basis of subsequent mutations.

Exploration after stagnation. If two consecutive score changes are smaller than ϵ, the coding agent proposes an exploratory edit η_{exp}. Unlike Δ_r, this edit need not target the component implicated by the latest diagnosis. It must nevertheless stay within the legal mutation space of the current evolution phase and preserve the phase's frozen artifacts, program interfaces, memory representation, and budget $B = (B_s, B_v)$. This permits movement beyond the latest diagnosis without coupling the text Writer, image Writer, and Reader evolution phases.

Targeted mutation. When neither rollback nor exploration is triggered, the coding agent applies Δ_r to the current program. At $r = 0$, neither historical guard is available, so the targeted edit is applied. At $r = 1$, only rollback can be evaluated. Exploration becomes eligible from $r = 2$, when two consecutive score changes are available.

A.4.3 STOPPING AND PROGRAM SELECTION

Each evolution phase terminates when its phase-specific convergence criterion is satisfied or its prescribed maximum number of rounds is reached. This convergence criterion is distinct from the stagnation condition in Equation (23): two stagnant score changes trigger exploration rather than immediate termination. If the phase terminates after round R, it returns

$$P_{\text{return}} = P_R^\star \tag{24}$$

rather than the final evaluated candidate P_R. Consequently, every phase returns its highest-scoring observed program.

A.5 MULTI-SCALE SOURCE-CORPUS CONSTRUCTION

We instantiate the multi-scale source corpus in the main text with $L = 3$ temporal scales, ordered from coarsest to finest. Specifically, the full video is uniformly partitioned into

$$|\mathcal{T}^1| = 16, \qquad |\mathcal{T}^2| = 32, \qquad |\mathcal{T}^3| = 64 \tag{25}$$

non-overlapping intervals. The shared video boundaries make the hierarchy nested: every interval $\Delta^\ell \in \mathcal{T}^\ell$ at a coarser scale is the union of two adjacent intervals at the next finer scale. This dyadic

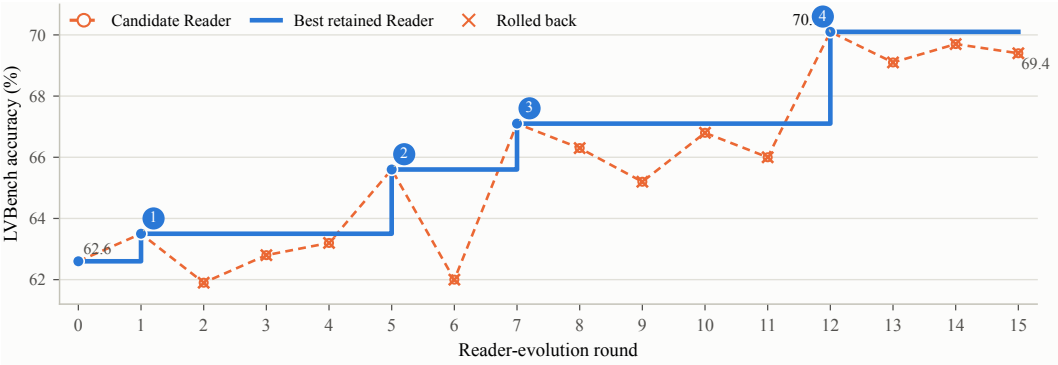

Figure 6: **Trajectory of Reader evolution.**

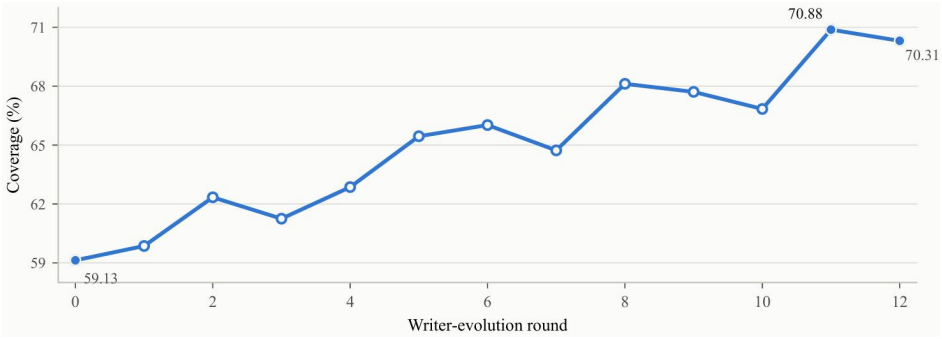

Figure 7: **Trajectory of Writer evolution.**

structure supports the bottom-up consolidation in Equation (4), from the 64-interval scale to the 32-interval scale and finally to the 16-interval scale.

For each interval Δ^{ℓ}, we uniformly sample 50 frames over its temporal extent and provide them to the captioner, which independently generates the source caption $c_{\Delta^{\ell}}$. These captions constitute the multi-scale source corpus used by the Writer; no information is propagated across intervals during caption generation, and cross-scale consolidation begins only in the subsequent writing workflow.

B Evolution Trajectory

We evolve the Writer and Reader 12 rounds and 15 rounds, respectively. The evolution trajectories are shown in Fig. 6 and Fig. 7. To ensure stable selection of the final Reader, we evaluate the Reader obtained at each evolution round on all 30 development videos available during evolution and compute its aggregate QA accuracy over this full set. We select the checkpoint with the highest accuracy as the final evolved Reader.

C Storage Reduction

A key advantage of our compact video memory is that it avoids repeatedly storing and processing the complete video stream. Each video is converted into a bounded persistent memory containing: (1) a textual memory with a 20K-token budget, and (2) 160 sparsely selected visual observations. The visual observations are stored as JPEG images with the resized to 448×448 pixels. This representation retains temporally distributed textual and visual evidence while substantially reducing the persistent storage footprint. Once constructed, the same memory can be reused for all questions associated with the video.

We measure storage using the actual serialized payload rather than estimating it from the number of observations. For a set of videos $\mathcal{V}$, the original-video storage is

$$S_{\text{video}} = \sum_{v \in \mathcal{V}} \text{bytes}(V_v), \tag{26}$$

where $\text{bytes}(V_v)$ is the on-disk size of the original compressed MP4 file. The compact-memory storage is

$$S_{\text{memory}} = \sum_{v \in \mathcal{V}} \left[\text{bytes}\big(M_v^{\text{text}}\big) + \sum_{j=1}^{160} \text{bytes}\big(I_{v,j}^{448}\big) \right], \tag{27}$$

where M_v^{text} is the final merged text serialized in UTF-8 and $I_{v,j}^{448}$ is the encoded 448×448 JPEG image. We exclude model parameters and temporary construction caches, since they are shared infrastructure rather than per-video persistent memory. Based on this accounting, the compact representation occupies only 0.142% of the original-video storage, yielding a 99.858% reduction in persistent storage. Thus, the proposed representation preserves sparse cross-modal evidence while remaining substantially cheaper to store, load, and reuse than the original videos.