LVMH: A META-OPTIMIZATION HARNESS FOR LONG-VIDEO AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video agents must map hours or days of multimodal evidence into a bounded context while respecting token and latency budgets, making memory design a context-management challenge. Existing memory systems largely rely on hand-designed policies, leaving open how to jointly optimize memory design for performance and efficiency. We propose Long Video Meta-memory Harness (LVMH), an automated multi-agent framework that formulates memory-policy development as multi-objective code search around a fixed answer model. To guard against self-certification and development-set overfitting, LVMH separates implementation from independent review and enforces protected observation boundaries. Persistent meta-memory records past proposals, traces, costs, and failures to guide subsequent policy revisions. Through this harness, we discover Self-Evolving Evidence Memory (SEEM), a "text-first, vision-second" policy that answers from compact structured event memory and evidence specialists, consulting localized frames through a selective visual gate. The policy evolved on EgoLifeQA generalizes to unseen benchmarks, outperforming Meta-Harness on both Ego-R1 Bench and Video-MME (L) with GPT-5 mini. Across the three benchmarks, SEEM with GPT-5 achieves 65.7% average accuracy, surpassing the previous state of the art by 7.7 percentage points with approximately $32\times$ lower average evaluation token use and $91\times$ shorter average response time.

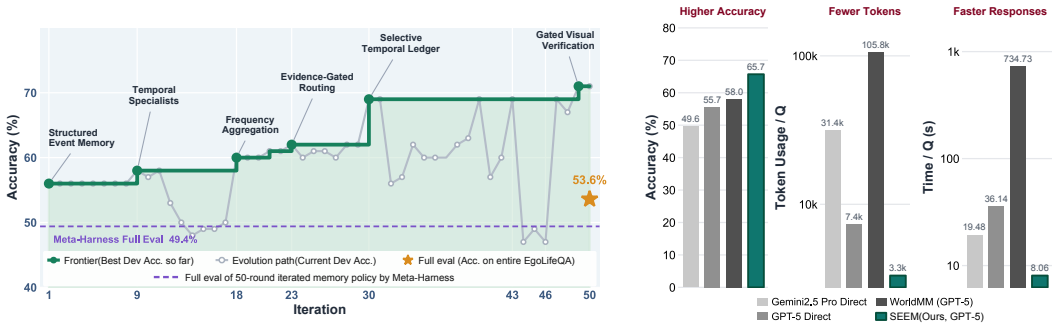


Figure 1: **Left:** The memory-policy frontier on the development set (an EgoLifeQA (Yang et al., 2026) subset), annotated with introduced mechanisms. After 50 iterations each, LVMH generalized better than Meta-Harness (Lee et al., 2026). **Right:** SEEM, the final policy produced by LVMH, surpassed direct-answering and previous state-of-the-art baselines in average accuracy and efficiency across the evaluation benchmarks.

1 INTRODUCTION

Multimodal large language models (MLLMs) (Bai et al., 2025; Wang et al., 2025a; Deepmind, 2026; Ataallah et al., 2024) are making long-video agents increasingly practical for robotics, autonomous driving, and video archives. These agents must find and connect evidence across recordings that span hours or days, even when the relevant event occupies only a few seconds. Recordings can exceed the model's context capacity, so reasoning requires retrieving relevant evidence at low cost.

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.

Existing systems address this challenge with external multimodal memories that store captions, transcripts, visual embeddings, event summaries, or graph-structured records and retrieve a subset at answer time (Wang et al., 2024; Fan et al., 2024; Yeo et al., 2026; Li et al., 2026a; Yin et al., 2026). Although promising, their memory architectures and retrieval policies remain largely designed by hand. Human-inspired categories, such as episodic and semantic memory (Yeo et al., 2026; Long et al., 2025; Fan et al., 2024), need not best match an AI agent's evidence needs. Moreover, improving accuracy together with token efficiency and latency requires coordinated changes to storage, retrieval, and evidence presentation, making manual design difficult. Finally, fixed memory structures limit an agent's ability to iteratively improve its memory policy using evaluation results.

These limitations motivate optimizing *context management itself*. A memory policy determines what to store, how to retrieve it, and when to inspect raw visual frames before supplying evidence to an answer model. We formulate its design as a *multi-objective optimization* problem over executable policies, jointly targeting higher accuracy, lower token use, and shorter latency. With the answer model fixed, the search can revise memory schemas, retrieval rules, and visual routing based on measured performance. Evaluation results thus guide revisions to the code that constructs the context, with each revision assessed for effectiveness and efficiency.

Automating this search requires reliable evaluation and controlled access to feedback. Recent harnesses demonstrate how agents can revise memory designs through code changes and execution results (Zhang et al., 2025; Liu et al., 2026). Two risks become important when applying this approach to long-video memory. *Self-certification* arises when the agent that implements a candidate also decides whether to accept it: apparent gains may hide evaluation errors or regressions. *Development-set overfitting* arises when repeated access to individual examples and detailed feedback encourages question-specific fixes that do not generalize. An effective search process therefore needs to define both who validates a candidate and what information its developer can observe.

We introduce Long Video Meta-memory Harness (LVMH), a multi-agent framework that combines policy search, independent review, and persistent *meta-memory*. Worker agents propose bounded changes and implement candidate policies in isolated workspaces. An orchestrator independently evaluates their accuracy, cost, temporal validity, and regressions before deciding which candidates to promote. Workers receive development inputs and permitted execution traces, while gold answers, rationales, oracle timestamps, and held-out examples remain outside their observation boundary. Meta-memory records proposals, evaluation results, costs, failures, and promotion decisions across iterations. This evidence guides later proposals, helps avoid repeated failures, and supports combining useful components from earlier candidates. Meta-memory thus guides how policies evolve, while video memory supplies the evidence used to answer queries.

The policy discovered through this process is Self-Evolving Evidence Memory (SEEM), which follows a text-first, vision-second strategy. It answers from compact structured event memory and uses specialists for operations such as time-delta computation and repeated-event aggregation. Figure 1 (left) shows how successive policy revisions improve development accuracy. After 50 iterations on a development set from EgoLife, the resulting policy achieves 53.6% accuracy on the full EgoLifeQA set with GPT-5 mini (OpenAI, 2026), beating 49.4% for Meta-Harness. Results on the held-out test split and transfer benchmarks provide further evidence of the policy's effectiveness beyond development data. Figure 1 (right) shows joint gains in effectiveness and efficiency: with GPT-5, SEEM achieves higher average accuracy across the evaluation benchmarks, surpassing the previous state of the art with fewer tokens and lower latency.

In summary, our contributions are:

- We bring meta-optimization with agentic harnesses to long-video agent memory, enabling agent-native memory policies that evolve through evaluation results to jointly improve accuracy, token efficiency, and latency.
- We design Long Video Meta-memory Harness (LVMH), a multi-agent framework intended to limit self-certification and development-set overfitting. Its evolved policy achieves higher validation and test accuracy than Meta-Harness in the reported run.
- Through LVMH, we discover Self-Evolving Evidence Memory (SEEM), a memory policy that combines strong long-video reasoning with efficient inference. It surpasses the previous state of the art in accuracy across three benchmarks, while using fewer evaluation tokens and achieving lower latency.

2 Related Work

Long Video Understanding. Long-video benchmarks require finding sparse events and relating evidence across extended recordings (Mangalam et al., 2023; Fu et al., 2025; Wu et al., 2024; Yang et al., 2026). Much prior work focuses on improving the model through efficient architectures and long-video training (Li et al., 2026b), or post-training for temporal grounding and tool use (Wang et al., 2026; Tian et al., 2025). Retrieval and video agents further enable dynamic evidence acquisition at inference time (Luo et al., 2026; Wang et al., 2024; Fan et al., 2024). We optimize the memory policy that stores, selects, and presents evidence while keeping the answer model fixed.

Agent Memory. Agent memory retains information beyond the immediate context and retrieves it for later decisions (Packer et al., 2024; Sumers et al., 2024; Zhang et al., 2024). Multimodal systems extend this role to audiovisual experiences and spatial records (Long et al., 2025; Zou et al., 2025; Liu et al., 2025), while structured and hierarchical video memories preserve events and their relations (Li et al., 2026a; Yin et al., 2026; Yeo et al., 2025). Many such systems adopt human-inspired categories, such as episodic and semantic memory (Long et al., 2025; Yeo et al., 2026). Although memory contents and retrieval can adapt, their architectures remain largely predefined. We instead iteratively redesign the memory policy through code search, jointly targeting accuracy, token use, and latency.

Meta Harness Optimization. Meta-Harness uses execution feedback to optimize the code surrounding a model (Lee et al., 2026), while MemEvolve and EvolveMem focus on evolving memory for textual agent traces and conversations (Zhang et al., 2025; Liu et al., 2026). These approaches allow full access to all benchmark examples and evaluation feedback during search, which raises concerns about *development-set overfitting*. LVMH is designed to limit this risk by separating development, validation, and test sets for policy search and independent frontier promotion.

3 Problem Formulation

Long Video Understanding. We study long-video understanding through question answering over recordings spanning hours or days. A query may require locating a brief event or connecting evidence across distant moments, while the available frames, captions, and transcripts can exceed the answer model's context capacity. Given a query q, answer space A_q, and video artifacts $\mathcal{X}_q$, a memory policy μ constructs a compact evidence context c_q for a fixed answer model f:

$$c_q = \mu(\mathcal{X}_q, q, A_q; \mathcal{B}), \qquad \hat{y}_q = f(q, A_q, c_q). \tag{1}$$

The policy governs what to store, how to retrieve relevant evidence, and how to assemble it under context and cost budgets $\mathcal{B}$. For a query at time t_q, $\mathcal{X}_q$ contains only evidence available by that time to preserve temporal causality.

Memory-Policy Optimization. We treat context management as multi-objective search over executable policies, targeting higher accuracy, lower token use, and shorter latency while accounting for memory construction and answer-time costs. With f fixed, search can revise memory representations, retrieval rules, and visual evidence selection. At iteration k, candidate policies $\mu \in \mathcal{C}_k$ are evaluated on the development set $\mathcal{D}$, yielding accuracy and cost vectors $J(\mu; \mathcal{D})$ and execution traces. The review operator U updates the policy frontier $\mathcal{F}_k$ and persistent meta-memory $\mathcal{H}_k$:

$$(\mathcal{F}_{k+1}, \mathcal{H}_{k+1}) = U(\mathcal{F}_k, \mathcal{H}_k, \{(\mu, J(\mu; \mathcal{D}), \mathrm{trace}(\mu))\}_{\mu \in \mathcal{C}_k}). \tag{2}$$

The frontier retains accepted policies, while meta-memory records proposals, outcomes, failures, and review decisions to guide subsequent search. This meta-memory supports policy development; the video memory governed by μ supplies evidence for individual queries. Section 4.1 specifies candidate generation and review.

4 Method

4.1 LVMH: Long Video Meta-memory Harness

LVMH is a self-evolving agentic harness that automates research and iterative improvement of long-video memory policies. The harness separates answer-time memory use from policy evolution through a two-layer architecture:

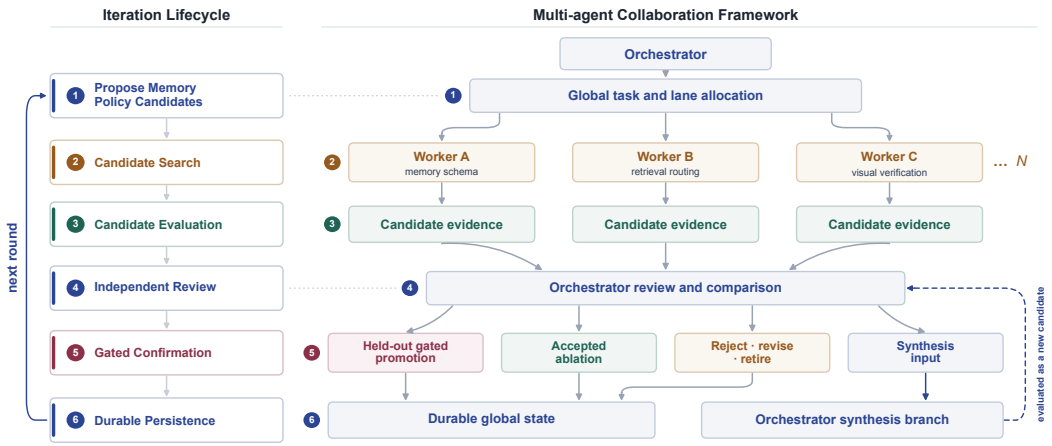


Figure 2: LVMH architecture and iteration lifecycle. *Left*: Policy optimization from hypothesis proposal to meta-memory update. *Right*: The Orchestrator assigns non-overlapping lanes; Workers develop and evaluate bounded candidates in isolated workspaces; the Orchestrator independently reviews and promotes them under controlled evaluation conditions.

- *Memory Layer:* The answer-time multimodal memory policy μ that transforms raw video artifacts $\mathcal{X}_q$ into a compact evidence context c_q for the fixed answer model f.
- *Meta Layer:* The autonomous research and governance system that guides hypothesis formulation, sandboxed implementation, and staged audits. It maintains persistent meta-memory $\mathcal{H}_k$ of proposals, execution traces, costs, and failures to guide iterations.

Policy evolution operates within a strict observation boundary to limit data leakage and development-set overfitting. We split the benchmark into a development set $\mathcal{D}$ for policy search, a held-out validation set $\mathcal{T}$ for frontier promotion, and a test set $\mathcal{S}$ for evaluation. During candidate development, the available task inputs are development queries q, answer spaces A_q, and pre-query video artifacts $\mathcal{X}_q$, alongside permitted runtime execution traces $\mathrm{trace}(\mu)$. Gold answers, rationales, oracle timestamps, and held-out examples are excluded from the development context.

Initialization. Before search, we initialize meta-memory $\mathcal{H}_0$ with four fixed sources that remain unchanged throughout policy evolution. *Research Principles* specify multi-objective optimization of accuracy, token use, and latency under temporal causality $t \leq t_q$ and context budgets $\mathcal{B}$. They prohibit shortcuts tailored to individual questions or datasets and direct search toward memory designs that generalize across long-video understanding tasks. The *Research Wiki* provides a literature review of long-video understanding, while the *Harness Constitution* defines the rules for agent collaboration, evaluation, and shared-state updates. The permitted *development-set artifacts* ground this initial knowledge in the video evidence available for policy development. Subsequent iterations extend meta-memory with search records while preserving these fixed materials.

4.1.1 Multi-Agent Collaboration Framework

Within this observation boundary, the meta layer coordinates Workers and an Orchestrator with separate responsibilities. Workers propose, implement, and locally evaluate candidate policies $\mu \in \mathcal{C}_k$, with read-only access to shared search state. The Orchestrator writes no policy code: it independently reviews candidates and alone controls held-out evaluation, frontier promotion, and updates to meta-memory. This separation guards against self-certification in previous autonomous optimization frameworks (Lee et al., 2026; Liu et al., 2026). As illustrated in Figure 2, each round implements the meta-update operator U through three stages:

Hypothesis Formulation and Sandboxed Implementation. Guided by the fixed initialization materials and accumulated evaluation results in meta-memory $\mathcal{H}_k$, Workers pre-register proposals P_μ specifying a bounded policy change, its expected effects on accuracy and cost, and fallbacks. The Orchestrator admits proposals distinct from previous attempts and assigns them to non-overlapping search lanes. Each Worker then implements its proposal in an isolated workspace. Keeping hypothesis

formulation and implementation with the same Worker preserves the reasoning behind coupled changes to memory indexing, retrieval, and context packing.

Local Development and Independent Review. Workers evaluate candidates on $\mathcal{D}$, recording the evaluation vector $J(\mu; \mathcal{D})$ and execution traces $\mathrm{trace}(\mu)$. They submit candidates for formal review only when results improve on the current frontier's development performance. The Orchestrator independently evaluates each submitted candidate against its incumbent baseline μ_{parent} using the same model f and prompt under identical cache conditions. The audit verifies reported gains and checks temporal causality to prevent future-event leakage.

Gated Evaluation and Meta-Memory Consolidation. Only candidates with independently verified development gains receive Orchestrator-issued authorization tokens for evaluation on the held-out validation set $\mathcal{T}$; this authorization does not expose held-out examples to Workers. Frontier promotion requires passing all five checks across admission, development review, and validation:

- **Novel Admission**: The proposal is bounded and structurally distinct from prior attempts ($\mu \in \mathcal{C}_k \setminus \mathcal{H}_k$).
- **Valid Execution**: The candidate yields a valid trace ($\exists\, \mathrm{trace}(\mu)$) within budget ($\mathrm{cost}(\mu) \leq \mathcal{B}$), bounded by the cost of the previous state of the art (Yeo et al., 2026).
- **Leakage Control**: The generated context must respect temporal causality to prevent future-event leakage ($c_q = \mu(\mathcal{X}_{\leq t_q}, q, A_q; \mathcal{B})$).
- **Dev Dominance**: The policy improves on its parent on $\mathcal{D}$. Within the budget, an increase in cost is allowed only when the relative performance gain is at least as large as the relative cost increase; strict Pareto dominance is not required.
- **Transfer Audit**: The gains persist on the held-out validation set ($\mu \succ_{\mathcal{T}} \mu_{\mathrm{parent}}$).

Passing candidates enter the updated frontier $\mathcal{F}_{k+1}$. The Orchestrator also consolidates proposals, $J(\mu; \mathcal{D})$, $\mathrm{trace}(\mu)$, review decisions, synthesis rules, and reflections into $\mathcal{H}_{k+1}$. This record includes attempts rejected during local development, helping later Workers avoid repeated failures and combine complementary mechanisms. Together, frontier promotion and meta-memory consolidation complete the update U and provide the starting state for the next round.

4.2 Self-Evolving Evidence Memory

SEEM is the final frontier memory policy discovered by LVMH within 50 iterations. SEEM adopts a "text-first, vision-second" strategy, answering primarily from structured event memory and consulting raw visual frames only when needed to verify or correct the initial answer. Figure 3 illustrates its two decoupled stages: offline memory construction and answer-time retrieval.

4.2.1 Stage I: Offline Memory Construction

Stage I processes each video v once, constructing a structured textual event bank $\mathcal{M}_{\text{text}}$ and a raw visual frame bank $\mathcal{M}_{\text{vis}}$:

Structured Textual Event Bank ($\mathcal{M}_{\text{text}}$). The video v is discretized into non-overlapping temporal windows $w_i = [t_i, t_i + \Delta t)$. We set $\Delta t = 30\,\text{s}$ to balance temporal granularity and context density. For each window w_i, an MLLM generates a dense caption. The caption is then parsed into a structured record r_i with five semantic fields:

$$r_i = \big\langle \texttt{EVENT}, \texttt{PEOPLE}, \texttt{OBJECTS}, \texttt{PLACE}, \texttt{DIALOGUE}, w_i \big\rangle. \tag{3}$$

The textual memory is defined as $\mathcal{M}_{\text{text}} = \{r_i\}_{i=1}^{N}$. Factorizing unstructured captions into typed fields enables fine-grained, field-aware matching during retrieval.

Raw Visual Frame Bank ($\mathcal{M}_{\text{vis}}$). Instead of pre-computing dense visual embeddings or per-frame descriptions, SEEM preserves visual evidence as raw pixel arrays. Frames are stored by timestamp as $\mathcal{M}_{\text{vis}} = \{(t, I_t) \mid t \in \mathcal{T}_{\text{sample}}\}$, where $\mathcal{T}_{\text{sample}}$ denotes the set of sampled timestamps (1 FPS) across the video, and I_t is the raw frame at time t. This mapping is constructed with zero model or embedding calls during offline ingestion.

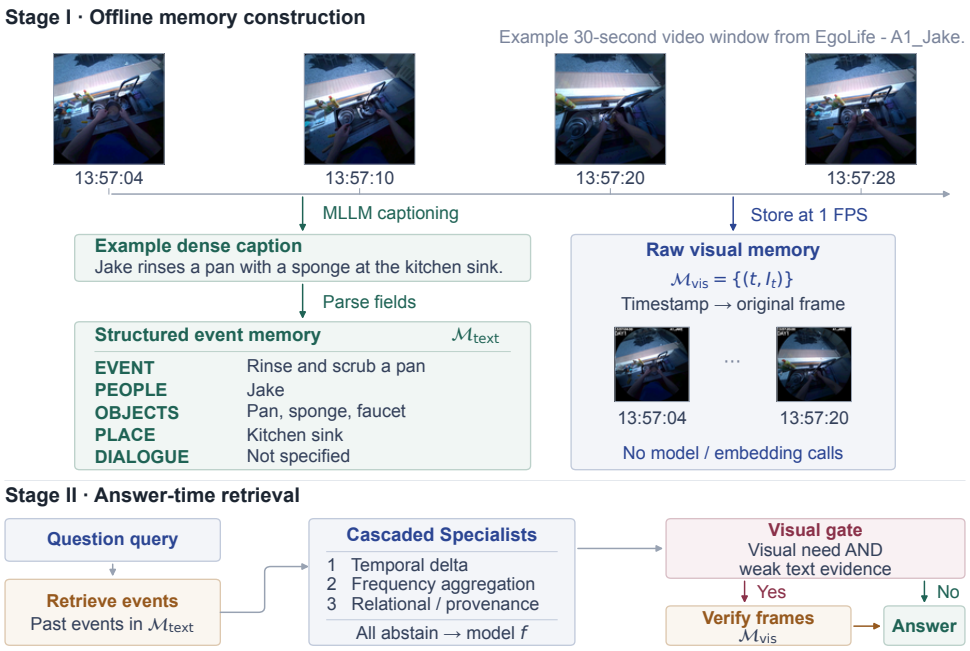


Figure 3: Overview of SEEM with an illustrative EgoLife example (Yang et al., 2026). Stage II uses cascaded specialists and gated visual verification.

4.2.2 Stage II: Answer-Time Retrieval

At answer time, SEEM resolves query q through causal filtering, fielded retrieval, cascaded evidence specialists, and gated visual verification.

1. Temporal Filtering and Fielded Event Retrieval. To preserve causal validity, retrieval uses only records available by the query time: $\mathcal{X}_q \triangleq \{r \in \mathcal{M}_{\text{text}} \mid r.t_{\text{end}} \leq t_q\}$. For each candidate record r, the system computes a field-weighted question relevance score $S_q(r) = \sum_{\phi \in \mathcal{K}} w_\phi(q) \cdot s_\phi(r)$, where $\mathcal{K}$ is the set of semantic fields, $w_\phi(q)$ is the field-specific weight including question-form adjustments, and $s_\phi(r)$ denotes the lexical token overlap between the query and field ϕ in record r. For each answer choice $c \in A_q$, a choice relevance score $S_c(r, c)$ is computed with boosted weights on exact entity and action matches. The total retrieval ranking score $S(r, c)$ balances individual relevance with a crucial *same-record co-occurrence bonus*:

$$S(r, c) = \alpha \cdot S_q(r) + \beta \cdot S_c(r, c) + \gamma \cdot \mathbb{I}\big[S_q(r) > 0 \ \wedge \ S_c(r, c) > 0\big]. \tag{4}$$

This term rewards records where the question and candidate co-occur within the same event window, filtering out unrelated mentions. Retrieval packs the top six global event anchors and top three evidence records per choice into a context buffer capped at 12,500 characters. All scale factors are heuristically determined by the LVMH agent during search; the final policy uses $\alpha = 1$, $\beta = 2$, and $\gamma = 4$. Appendix B.1 provides more details.

2. Cascade of Evidence Specialists. SEEM applies three types of specialists in sequence. *Temporal delta specialists* compute elapsed time from matching events. *Frequency aggregators* count support across distinct timestamps. *Relational and provenance specialists* require queried entities to co-occur within the same event. Each specialist answers only when its evidence requirements are met; otherwise, it abstains and passes the query onward. If all abstain, the model f answers from the packed event context. The selected route produces a text answer $\hat{a}_t$ and route metadata z_t for subsequent verification. Detailed specialist rules appear in Appendix B.2.

3. Selective Visual Escalation and Verification. To recover visual details, SEEM invokes a multimodal verifier only when visual inspection is needed and text evidence is weak:

$$\text{OpenVisual}(q, \hat{a}_t, z_t) = \text{StrongVisualNeed}(q) \ \wedge \ \text{WeakText}(\hat{a}_t, z_t). \tag{5}$$

Table 1: Evaluation results (accuracy, %) on long-video understanding benchmarks. **Bold** marks the best result. Red shading denotes results on the full 500-question EgoLifeQA set, including 100 development, 200 validation, and 200 test questions. Blue shading denotes transfer benchmarks that are not observable to the agent during iterative optimization.

		EgoLifeQA								
Method	Model	Ent.	EvR.	Hab.	Rel.	Task	Avg.	Ego-R1 Bench	Video-MME (L)	Avg.
Base Models										
Qwen3-VL-8B	8B	35.2	30.2	39.3	46.4	46.0	38.6	35.7	48.0	40.8
Gemini 2.5 Pro	–	43.2	40.5	41.0	55.2	52.4	46.4	46.7	55.7	49.6
GPT-5 mini	–	30.4	38.1	37.7	27.2	36.5	33.2	34.3	54.1	40.5
GPT-5	–	47.2	42.1	47.5	53.6	55.6	48.6	46.3	72.3	55.7
Long Video LLMs										
VideoChat-Flash	7B	28.8	32.5	37.7	37.6	38.1	34.2	42.7	44.1	40.3
Time-R1	7B	39.2	50.8	65.6	48.8	47.6	48.8	48.0	37.6	44.8
Video-RTS	7B	40.8	48.4	62.3	48.8	47.6	48.2	47.4	47.9	47.8
RAG-based Video LLMs										
LightRAG	–	40.8	48.4	67.2	50.4	44.4	48.8	52.3	46.6	49.2
HippoRAG	–	48.8	**60.3**	70.5	**60.8**	**66.7**	59.6	56.0	52.1	55.9
Video-RAG	–	49.6	56.3	67.2	55.2	54.0	55.4	49.7	55.4	53.5
Memory-based Video LLMs										
EgoRAG	–	40.0	56.3	62.3	54.4	52.4	52.0	49.0	41.1	47.4
Ego-R1	3B	51.2	53.2	63.9	50.4	50.8	53.0	52.0	42.7	49.2
HippoMM	–	45.6	53.2	70.5	55.2	58.7	54.6	53.0	41.6	49.7
M3-Agent	32B	44.4	54.8	62.3	56.8	54.0	53.5	52.0	55.3	53.6
WorldMM	GPT-5 mini	41.6	42.9	41.0	44.0	49.2	43.4	50.3	52.6	48.8
WorldMM	GPT-5	50.4	48.4	59.0	45.6	58.7	50.8	50.6	72.6	58.0
Automated Harness Optimization										
Meta-Harness	GPT-5 mini	45.6	50.0	52.5	52.8	46.0	49.4	42.0	47.2	46.2
SEEM (ours)	GPT-5 mini	47.2	55.6	65.6	51.2	55.6	53.6	53.0	55.9	54.2
SEEM (ours)	Qwen3-VL-8B	50.4	49.2	63.9	47.2	54.0	51.4	50.3	55.1	52.3
SEEM (ours)	GPT-5	**61.6**	55.6	**78.7**	52.8	**66.7**	**60.6**	**62.0**	**74.4**	**65.7**

When the gate is closed, SEEM returns the text answer directly. When it opens, f inspects timestamp-localized frames from $\mathcal{M}_{\text{vis}}$ and is instructed to retain $\hat{a}_t$ unless visual evidence clearly supports an alternative. If frames are unavailable or verification fails, the text answer is preserved. Appendix B.3 provides details.

5 Experiments

Data and Baselines. We evaluate SEEM on three benchmarks (detailed in Appendix A): EgoLifeQA (Yang et al., 2026), Ego-R1 Bench (Tian et al., 2025), and Video-MME (Fu et al., 2025). For LVMH and Meta-Harness, we randomly partition EgoLifeQA's 500 questions into 100 for policy development, 200 for validation during frontier promotion, and 200 for final testing. The test split is excluded from policy search and frontier promotion. Ego-R1 Bench and Video-MME (L) are used to evaluate transfer. We compare SEEM with video LLMs (OpenAI, 2026; Bai et al., 2025; Google DeepMind, 2026), long-video LLMs (Li et al., 2026b; Wang et al., 2026; 2025b), RAG-based video systems (Luo et al., 2026; Guo et al., 2025; Gutiérrez et al., 2024), memory-based video agents (Tian et al., 2025; Long et al., 2025; Yeo et al., 2026), and harness optimization (Lee et al., 2026).

Implementation. We use GPT-5.5 (OpenAI, 2026) as the search agent for the memory evolution of LVMH and Meta-Harness. Memory construction, retrieval, and gating are implemented with GPT-5 mini (OpenAI, 2026). We evaluate the final memory policy produced by LVMH, namely SEEM. Further implementation details are provided in Appendix B.

5.1 Evaluation

Main Results. With GPT-5, SEEM achieves the highest average accuracy in Table 1, exceeding WorldMM by 7.7 percentage points. With GPT-5 mini, SEEM reaches 54.2% overall, exceeding WorldMM and Meta-Harness by 5.4 and 8.0 percentage points, respectively. Compared to Meta-

Table 2: GPT-5 cost per question: requests, tokens (k), and time (s). Direct uses no memory.

Method	Req.	Tok.	Time
EgoLifeQA			
Direct	1.00	11.34k	43.07
WorldMM	30.97	36.21k	140.51
SEEM	**0.97**	**3.57k**	**9.76**
Video-MME (L)			
Direct	**1.00**	6.93k	24.94
WorldMM	201.20	134.02k	981.70
SEEM	**1.00**	**3.30k**	**7.22**

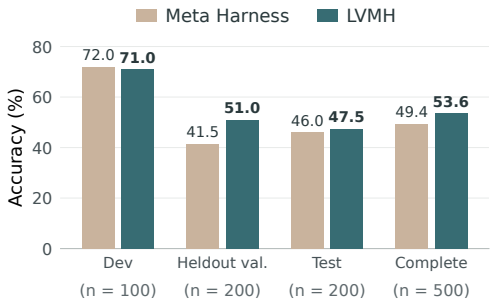


Figure 4: EgoLifeQA accuracy by split for the final frontier policies. LVMH outperforms Meta-Harness on validation and test.

Harness, the gains are 4.2 points on EgoLifeQA, 11.0 on Ego-R1 Bench, and 8.7 on Video-MME (L). Figure 4 shows that SEEM also outperforms Meta-Harness on EgoLifeQA's validation and test splits. These results indicate that LVMH limits development-set overfitting. The policy also improves Qwen3-VL-8B from 40.8% to 52.3% overall, indicating that its benefits extend beyond the backbone used during policy evolution.

Efficiency Analysis. Table 2 reports GPT-5 evaluation costs for SEEM, direct answering, and WorldMM. SEEM uses fewer tokens and has lower per-question latency than both baselines, with no more model requests. On Video-MME (L), it reduces token use by 40.6× and wall-clock time by 136× compared to WorldMM. On EgoLifeQA, specialists reduce the average number of model requests per question below 1. Offline construction also uses fewer tokens than WorldMM (Table 9).

5.2 Ablation

Harness Design. Figure 5 compares four settings: the complete LVMH, LVMH without gated review, LVMH without meta-memory, and a single-agent variant without gated review. We run all four settings concurrently from scratch for 15 iterations each and plot their gains against cumulative token consumption. Without gated review, the agent must rely on validation evaluations to decide whether to promote a candidate to the frontier, substantially increasing token consumption over iterations. Gated review reduces this cost by screening candidates before validation. Retaining meta-memory accelerates development-set improvement. Records of failed attempts and earlier lessons help the agent avoid retrying unsuccessful designs and guide subsequent proposals. The complete framework also maintains better transfer gains and faster development progress.

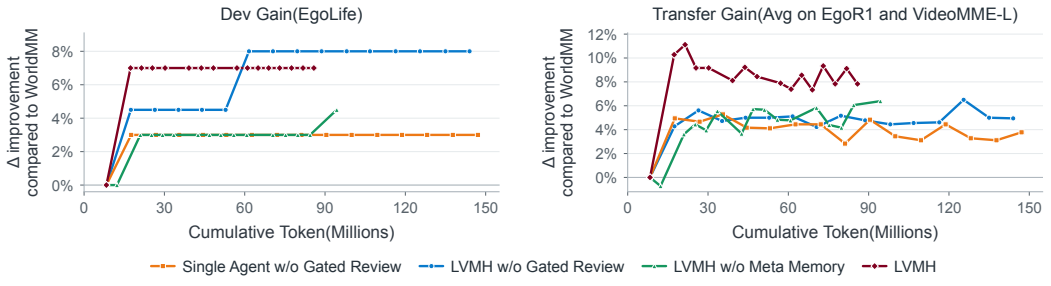


Figure 5: Harness ablation: four settings run concurrently from scratch for 15 iterations. Curves show gains over WorldMM versus cumulative search tokens (millions). Left: EgoLifeQA development. Right: mean transfer gain on Ego-R1 Bench and Video-MME (L). Gated review reduces token use; meta-memory accelerates development gains.

SEEM Component Ablation. Table 3 evaluates SEEM's four components. Removing structured memory causes the largest drop (4.8 percentage points). Removing specialists, the co-occurrence bonus, or gated visual verification reduces accuracy by 2.4, 3.0, or 2.8 points, respectively. Appendix C.1 compares rule-based and LLM gates.

Table 3: Component ablation of SEEM. Each variant removes one component.

Variant	Accuracy (%)
SEEM	**53.6**
w/o Structured Memory	48.8
w/o Specialists	51.2
w/o Co-occurrence Bonus	50.6
w/o Gated Visual Verification	50.8

5.3 Evolution Process Analysis

Figure 6 traces successive revisions that suggest three lessons for long-video memory design.

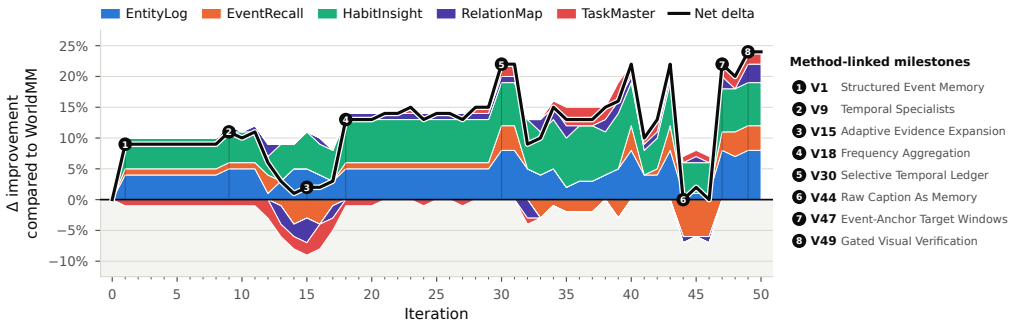


Figure 6: EgoLifeQA development gains by question type, annotated with policy changes.

Efficiency. The search first reduces context size through structured event memory (V1), then reduces model calls through temporal and relational specialists (V9) and frequency aggregation (V18). Memory thus evolves from a compact evidence store to support explicit computation. This suggests two routes to efficiency: compress the evidence supplied to the model and resolve sufficiently supported queries through memory operations.

Information Loss and Localization. Fixed windows and structured compression can discard details and fragment continuous events. Broad narrative expansion (V15) reduces EventRecall, while a selective temporal ledger (V30) better connects relevant events. Replacing fields with raw captions (V44) again causes a regression; expansion around retrieved event anchors (V47) recovers useful neighboring context. Recovering information thus requires accurate localization: preserve structured anchors, then expand around events relevant to the question.

The Role of Visual Information. Text retrieval cannot recover visual details omitted during memory construction. Gated verification (V49) adds a complementary source of evidence, reaching the highest development accuracy in the trajectory. Yet broader visual invocation does not improve accuracy (Appendix C.1). This suggests using text memory for reasoning across long timescales and consulting frames selectively to verify unresolved appearance, spatial, or counting details.

6 Conclusion

We addressed context and budget management for long-video agents through automated meta-optimization of multimodal memory design. LVMH, a multi-agent harness, combines strict observation boundaries and independent orchestrator auditing to mitigate self-certification and development-set overfitting. It discovered SEEM, a lightweight, "text-first, vision-second" memory policy that achieves superior accuracy with significantly fewer tokens and lower latency. Its evolution suggests three lessons for long-video agent design: (1) memory should retrieve and distill key evidence rather than pass all relevant information to the model; (2) visual evidence should complement memory reasoning, with its value balanced against processing cost; and (3) policies based on rules and text can be efficient and effective.

AI USE STATEMENT

In this work, we used generative AI tools to propose or refine hypotheses, design research methodologies or experiments and provide feedback on them, implement methods, assist with translation and writing, and clean and reformat data. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413*, 2024.

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

Google Deepmind. Gemma 4. https://deepmind.google/models/gemma/gemma-4/, 2026.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding, 2024. URL https://arxiv.org/abs/2403.11481.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis, 2025. URL https://arxiv.org/abs/2405.21075.

Google DeepMind. Gemini Pro. https://deepmind.google/models/gemini/pro/, 2026. Accessed: 2026-09-17.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *EMNLP (Findings)*, pp. 10746–10761, 2025.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses, 2026. URL https://arxiv.org/abs/2603.28052.

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, 2026a. URL https://arxiv.org/abs/2605.08271.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, volume 2026, pp. 109089–109117, 2026b.

Yueqian Lin, Jingyang Zhang, Qinsi Wang, Hancheng Ye, Yuzhe Fu, Yudong Liu, Hai Helen Li, and Yiran Chen. Hippomm: Hippocampal-inspired multimodal memory for long audiovisual event understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5968–5977, 2026.

Jiaqi Liu, Xinyu Ye, Peng Xia, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Evolvemem: Self-evolving memory architecture via autoresearch for llm agents, 2026. URL https://arxiv.org/abs/2605.13941.

Junming Liu, Yifei Sun, Weihua Cheng, Haodong Lei, Yirong Chen, Licheng Wen, Xuemeng Yang, Daocheng Fu, Pinlong Cai, Nianchen Deng, Yi Yu, Shuyue Hu, Botian Shi, and Ding Wang. Memverse: Multimodal memory for lifelong learning agents, 2025. URL https://arxiv.org/abs/2512.03627.

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, 2025. URL https://arxiv.org/abs/2508.09736.

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. *Advances in Neural Information Processing Systems*, 38:168008–168033, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding, 2023. URL https://arxiv.org/abs/2308.09126.

OpenAI. GPT-5. https://openai.com/gpt-5/, 2026. Accessed: 2026-09-17.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents, 2024. URL https://arxiv.org/abs/2309.02427.

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.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent, 2024. URL https://arxiv.org/abs/2403.10517.

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. *Advances in Neural Information Processing Systems*, 38:83330–83364, 2026.

Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28114–28128, 2025b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, Bo Li, and Ziwei Liu. Egolife: Towards egocentric life assistant, 2026. URL https://arxiv.org/abs/2503.03803.

Jeong Hun Yeo, Sangyun Chung, Sungjune Park, Dae Hoe Kim, Jinyoung Moon, and Yong Man Ro. Gcagent: Long-video understanding via schematic and narrative episodic memory, 2025. URL https://arxiv.org/abs/2511.12027.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning, 2026. URL https://arxiv.org/abs/2512.02425.

Yufei Yin, Qianke Meng, Minghao Chen, Jiajun Ding, Zhenwei Shao, and Zhou Yu. Videoarm: Agentic reasoning over hierarchical memory for long-form video understanding, 2026. URL https://arxiv.org/abs/2512.12360.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. Memevolve: Meta-evolution of agent memory systems. *arXiv preprint arXiv:2512.18746*, 2025.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model based agents, 2024. URL https://arxiv.org/abs/2404.13501.

Xueyan Zou, Yuchen Song, Ri-Zhao Qiu, Xuanbin Peng, Jianglong Ye, Sifei Liu, and Xiaolong Wang. M3: 3d-spatial multimodal memory, 2025. URL https://arxiv.org/abs/2503.16413.

A Evaluation Datasets and Baselines

We evaluate on two benchmarks for memory over daily egocentric recordings and one benchmark for general video understanding. Table 4 summarizes the benchmarks used in our experiments.

Table 4: Benchmarks used in our experiments. EgoLifeQA contains the 500 questions in the public A1 Jake release. Video-MME (L) contains three questions for each of 300 long videos.

Benchmark	Questions	Role in Our Experiments
EgoLifeQA	500	100 development / 200 validation / 200 test
Ego-R1 Bench	300	Transfer evaluation
Video-MME (L)	900	Transfer evaluation

EgoLifeQA. EgoLifeQA (Yang et al., 2026) evaluates memory and reasoning over daily life. The EgoLife collection contains 266 hours of retained first-person audiovisual footage from six participants living together for seven days. The paper describes 3,000 multiple-choice questions, with 500 questions per participant, but the public QA release contains only the 500-question A1 Jake subset; the full 3,000-question set has not been released. Its five categories are *EntityLog* (entities and their attributes), *EventRecall* (past events), *HabitInsight* (recurring behaviors), *RelationMap* (interpersonal relations), and *TaskMaster* (task-related reasoning). Questions are anchored to a query time and may require evidence from earlier days. Our experiments use this entire public QA release, corresponding to approximately 51.6 hours of Jake's video.

Ego-R1 Bench. Ego-R1 Bench (Tian et al., 2025) contains 300 multiple-choice questions, distributed evenly across six first-person perspectives in EgoLife. Each perspective has 50 questions drawn from human-labeled and human-verified examples. The benchmark emphasizes detailed visual reasoning and spatial and temporal relations across multiple observations. We evaluate on all 300 questions. This benchmark is separate from Ego-R1's training data, which contain question-answer pairs and tool-use reasoning traces. Ego-R1 Bench evaluates transfer across benchmark questions and participant perspectives within the same egocentric recording environment.

Video-MME. Video-MME (Fu et al., 2025) contains 900 videos totaling approximately 254 hours, with 2,700 human-annotated multiple-choice questions. The videos cover six broad domains and 30 subfields, and the questions test 12 perception and reasoning tasks. Video durations range from 11 seconds to one hour, with 300 videos in each of the short, medium, and long groups. We use the long split: 300 videos lasting 30 to 60 minutes and 900 questions. The benchmark also provides audio tracks and, where available, subtitles. Its diverse video sources assess transfer beyond the shared egocentric recording environment of EgoLifeQA and Ego-R1 Bench.

Evaluation protocol. We report multiple-choice accuracy, computed as the proportion of correctly answered questions. For both LVMH and Meta-Harness, we randomly partition the 500 Jake questions into 100 for policy development, 200 for validation during frontier promotion, and 200 for final testing. The test split is excluded from policy search and frontier promotion. The reported 500-question EgoLifeQA results include all three splits; Figure 4 reports accuracy separately for each split. Ego-R1 Bench and Video-MME (L) are used for transfer evaluation. For timestamped egocentric questions, memory access is restricted to evidence available by the query time (Section 4.1).

Baselines. Table 1 groups the baselines into five categories: *base models*, including Qwen3-VL-8B (Bai et al., 2025), Gemini 2.5 Pro (Google DeepMind, 2026), and GPT-5 mini and GPT-5 (OpenAI, 2026); *long-video LLMs*, including VideoChat-Flash (Li et al., 2026b), Time-R1 (Wang et al., 2026), and Video-RTS (Wang et al., 2025b); *RAG-based video systems*, including LightRAG (Guo et al., 2025), HippoRAG (Gutiérrez et al., 2024), and Video-RAG (Luo et al., 2026); *memory-based video agents*, including EgoRAG (Yang et al., 2026), Ego-R1 (Tian et al., 2025), HippoMM (Lin et al., 2026), M3-Agent (Long et al., 2025), and WorldMM (Yeo et al., 2026); and *automated harness optimization*, represented by Meta-Harness (Lee et al., 2026). We reproduce the direct-evaluation results for the base models and the results for WorldMM and Meta-Harness; all other baseline results are taken from their respective papers.

B METHOD DETAILS

We detail SEEM's temporal filtering, fielded event retrieval, evidence specialists, and visual verification (Section 4.2.2). We retain the main-text notation: q is the question, A_q its answer space, t_q its observation cutoff, c_q the packed evidence context, and f the fixed answer model. An individual answer choice is denoted by $c \in A_q$.

Meta-Harness reproduction. We reproduce Meta-Harness using the same GPT-5.5 search agent as LVMH, with the same budget of 50 search iterations and identical EgoLifeQA development, validation, and test splits (100/200/200 questions). For both methods, the test split is excluded from policy search and frontier promotion.

B.1 TEMPORAL FILTERING AND FIELDED EVENT RETRIEVAL

Temporal eligibility. Each textual record summarizes a 30-second event window $[r.t_{\text{start}}, r.t_{\text{end}})$ and retains five fields: `EVENT`, `PEOPLE`, `OBJECTS`, `PLACE`, and `DIALOGUE`. For $r = r_i$, this is the window w_i of duration $\Delta t = 30$ s defined in Section 4.2.1. Using the main-text notation for eligible records in this retrieval stage,

$$\mathcal{X}_q = \{r \in \mathcal{M}_{\text{text}} : r.t_{\text{end}} \le t_q\}. \tag{6}$$

The comparison uses the window's end time: a window that starts before t_q but ends after it is excluded. Both ranking and support counting operate on $\mathcal{X}_q$, so future records cannot influence either the displayed evidence or its support statistics.

Field-aware question matching. Let $T(x)$ denote the lexical token set of text x. The field set $\mathcal{K}$ consists of `EVENT`, `PEOPLE`, `OBJECTS`, `PLACE`, and `DIALOGUE`; it is distinct from the policy frontier $\mathcal{F}_k$. For $\phi \in \mathcal{K}$, let r_ϕ denote that field's text and define $s_\phi(r) = |T(q) \cap T(r_\phi)|$, as in the main text. Matching is performed separately within each field. The base question score is

$$\begin{aligned} S_q^{\text{base}}(r) = 3.0\, s_{\texttt{EVENT}}(r) + 1.5\, s_{\texttt{PEOPLE}}(r) + 1.5\, s_{\texttt{OBJECTS}}(r) \\ + 1.25\, s_{\texttt{PLACE}}(r) + 2.0\, s_{\texttt{DIALOGUE}}(r). \end{aligned} \tag{7}$$

Table 5 lists these coefficients. The score favors agreement with the event action while retaining entity, location, and speech evidence. Question-form adjustments further emphasize `PLACE` for "where" questions, `PEOPLE` for "who" or "with whom" questions, and `DIALOGUE` for speech or conversation questions. We denote the resulting question score by $S_q(r) = \sum_{\phi \in \mathcal{K}} w_\phi(q) s_\phi(r)$. The table gives the base weights w_ϕ^{base}; $w_\phi(q)$ includes the question-form adjustments. These coefficients are fixed retrieval rules in the final policy, selected during LVMH's policy search.

Table 5: Base question-matching weights in fielded event retrieval. Each coefficient multiplies the lexical overlap with its field.

Field	`EVENT`	`PEOPLE`	`OBJECTS`	`PLACE`	`DIALOGUE`
w_ϕ^{base}	3.0	1.5	1.5	1.25	2.0

Choice matching and same-event support. For every answer choice $c \in A_q$, the policy separately computes $S_c(r, c)$ from the choice's overlap with the record's fields. People matches receive weight 3.5, and an exact choice-phrase match adds 5.0 to the choice score. With the same $S_q(r)$, $S_c(r, c)$, and coefficient meanings as Eq. 4, question and choice relevance combine as

$$S(r, c) = S_q(r) + 2.0\, S_c(r, c) + 4.0\, \mathbb{I}[S_q(r) > 0 \ \wedge \ S_c(r, c) > 0]. \tag{8}$$

Thus $\alpha = 1$ multiplies S_q, $\beta = 2$ multiplies S_c, and $\gamma = 4$ is the same-record co-occurrence bonus, exactly as in the main text. The 4.0 bonus is awarded once for each record and choice pair when both scores are positive. It promotes evidence where the question and choice are mentioned in the same event window; lexical co-occurrence alone does not establish that the record entails the answer. For example, if $S_q(r) = 3.0$ and $S_c(r, c) = 3.5$, the combined score is $3.0 + 2.0 \times 3.5 + 4.0 = 14.0$ before temporal reranking. The 5.0 exact-phrase bonus in S_c adds 10.0 to the total.

Temporal ordering and support counts. After lexical scoring, a positional bias favors later records for "last," "latest," or "previous" queries and earlier records for "first" or "earliest" queries. This

reranking operates within the eligible bank and does not change the cutoff in Eq. 6. The policy also scans the eligible bank to compute

$$n_c = \sum_{r \in \mathcal{X}_q} \mathbb{I}[S_q(r) > 0 \ \wedge \ S_c(r, c) > 0]. \tag{9}$$

Each qualifying record contributes exactly 1 to n_c, irrespective of its ranking score; one record may contribute to multiple choices. The count covers all eligible records, including records omitted from the packed context. For habit questions, repeated support across timestamps supplies evidence beyond a single salient event. The structured fallback stores these statistics as `choice_support_counts` and renders them as `field_cooccurrence_count`. They are lexical support counts, not calibrated answer probabilities, and differ from the caption-level `choice_hit_counts` used by the visual gate.

Evidence packing and answer generation. The structured context c_q contains up to 6 globally relevant event anchors and up to 3 evidence records for each answer choice. With 4 choices, this allocates at most $6 + 4 \times 3 = 18$ record slots before the character cap; the slots need not represent 18 distinct events. The context is capped at 12,500 characters. It includes the question type, memory mode, temporal hint, number of eligible records, per-choice support counts, and each selected record's score, timestamp, and available semantic fields. The responder is instructed to check action agreement in `EVENT`, require same-record evidence for people and places, prioritize `DIALOGUE` for speech questions, and aggregate repeated evidence for habits. Global mention frequency alone is insufficient to select an answer. When the structured fallback is used, one text-model call resolves the packed evidence into a single answer letter from A to D.

B.2 Evidence Specialist Rules

The specialists run in the following order. Each requires evidence suited to its operation and abstains when that requirement is unmet.

Temporal Delta Specialists. For relative-time queries such as "When was the last time… ?", the specialist localizes the most recent matching event anchor, computes the time delta relative to t_q, and maps the duration to the answer choices, answering without an LLM call.

Frequency Aggregators. For recurring behavior queries, the specialist aggregates support counts across distinct timestamps. It answers directly only when the leading choice has a clear support margin over the alternatives.

Relational and Provenance Specialists. For queries about interactions, such as meal companions or joint tasks, the specialist binds entities within the same event: all queried entities must co-occur in the same action record before it can answer. Unrelated records cannot establish this association.

Structured Event Fallback. If all specialists abstain, the packed event context is passed to the fixed model f for reasoning across the structured fields: $\hat{a}_t = f(q, A_q, c_q)$. Here the subscript t in $\hat{a}_t$ and z_t denotes the text stage; the query cutoff remains t_q. The resulting text answer $\hat{a}_t$ and route metadata z_t are passed to the visual gate, as are answers produced directly by specialists.

B.3 Visual Gate and Verification

The gate in Eq. 5 requires both a visual need and weak text evidence.

Visual-Need Predicate. $\mathrm{StrongVisualNeed}(q)$ is a deterministic lexical rule applied to the concatenation of the question-type label and question text. It does not inspect the answer choices, text-stage answer, retrieved evidence, or frame availability, and requires no model call. Matching is case-insensitive. At least one term in Table 6 must match.

Five exclusion rules then suppress matches associated with temporal aggregation, identity, preference, or relational wording. Let $\mathrm{Trigger}(q)$ denote a trigger match and $E_j(q)$ an exclusion match. The predicate is

$$\mathrm{StrongVisualNeed}(q) = \mathrm{Trigger}(q) \wedge \neg \left(\bigvee_{j=1}^{5} E_j(q) \right). \tag{10}$$

Table 6: Examples of trigger vocabulary for $\mathrm{StrongVisualNeed}(q)$. A trigger match is necessary but can be overridden by an exclusion rule.

Category	Trigger terms
Color	`color`, `colour`
Counting	`how many`, `number of`, `count`
Position	`left`, `right`, `above`, `below`
Text or screen	`sign`, `text`, `word`, `screen`
Visible state	`visible`, `shown`, `look`, `looks`
Worn or held objects	`wearing`, `wear`, `held`, `holding`
Shape or markings	`shape`, `brand`, `logo`

The exclusions are:

1. **Temporal frequency:** `how many times`, `number of times`, `most often`, `past N days` (with numeric N), `usually`, `often`, `frequency`, or `frequently`.
2. **Unconstrained identity:** questions containing `who`, unless it is immediately followed by `is`, `was`, `were`, `appear(s)`, `look(s)`, `wearing`, `wears`, `holding`, `holds`, `visible`, or `shown`, with a subsequent cue from `wearing`, `wears`, `holding`, `holds`, `visible`, `shown`, `color`, or `colour`. Here `appear(s)` and `look(s)` denote the base and third-person singular forms.
3. **Preference-related wording:** `like`, `likes`, `prefer`, `prefers`, `drink`, or `drinks`.
4. **Nonvisual idiom:** the phrase `look after`.
5. **Position with relation or negation:** a positional term (`left`, `right`, `above`, or `below`) together with `who`, `whom`, `whose`, `with`, `together`, `not`, `n't`, `did not`, `without`, or `participate`.

For example, "Who is wearing the red shirt?" satisfies the visual-need predicate, whereas "How many times did I wear the red shirt?" is excluded as a temporal-frequency query. These rules approximate visual need from wording; they do not establish that the retrieved frames can answer the question. They can also miss visual queries: the preference exclusion suppresses `look like`, and unlisted inflections do not trigger a match.

Text-Evidence Check and Verification. For $\mathrm{WeakText}(\hat{a}_t, z_t)$, let p indicate whether the text answer is parseable, d whether the route was direct, ρ_t the route name, and h_1, h_2 the top two lexical choice-hit counts. The route name ρ_t, direct-route flag d, and counts are read from z_t; r remains reserved for an event record. The text evidence is considered weak unless all reliability checks hold:

$$\mathrm{WeakText}(\hat{a}_t, z_t) = \neg\Big(p \ \wedge \ d \ \wedge \ (\texttt{"direct"} \in \rho_t) \ \wedge \ (h_1 > 2) \ \wedge \ (h_1 - h_2 > 1)\Big). \quad (11)$$

When the gate remains closed, the policy returns $\hat{a}_t$ without visual verification, so the final prediction in Section 3 is $\hat{y}_q = \hat{a}_t$. Otherwise, it ranks caption windows and retrieves at most four relevant frames ($N_{\text{frames}} \leq 4$) from $\mathcal{M}_{\text{vis}}$. The fixed model f inspects these localized frames and is instructed to retain the text answer unless visual evidence clearly supports an alternative. Missing frames or verification failures also leave $\hat{a}_t$ unchanged.

Conservative Visual Verification Prompt. When the visual gate opens, the multimodal verifier is guided by the following prompt:

> *"Current text answer is $\hat{a}_t$. Inspect frames. Maintain $\hat{a}_t$ unless visual evidence unambiguously proves an alternative."*

C Additional Results

C.1 Visual Gate Design

Table 7 compares the evolved rule-based gate with an LLM judge on EgoLifeQA, Ego-R1 Bench, and Video-MME (L). The rule-based gate invokes visual verification 112 times, compared with 450 for the LLM judge, or approximately $4.0\times$ fewer calls.

Table 7: Visual-gate ablation with GPT-5 mini across EgoLifeQA, Ego-R1 Bench, and Video-MME (L). Accuracy is reported in percent. Calls reports visual-verification calls across the three benchmarks, excluding LLM gate-decision calls.

Gate	EgoLife	Ego-R1	VMME-L	Calls
LLM judge	53.0	52.7	55.7	450
Evolved rules (SEEM)	53.6	53.0	55.9	**112**

C.2 Answer-Time Cost with Open Backbones

Deployment cost with an open backbone. Table 8 reports per-question cost when the answering model is the open Qwen3-VL-8B checkpoint served locally, with memory built by GPT-5 mini as everywhere else in this paper. Across the three benchmarks, mean per-question cost remains below 5.7k tokens and 5 s, with at most 1.25 model requests per question on average. On the two personal-memory benchmarks, the request count is below one (0.94 on EgoLifeQA and 0.96 on Ego-R1 Bench) because a fraction of questions is resolved from memory without a model call at all, a floor that a per-question frame-prompting interface cannot reach.

Table 8: Answer-time cost of SEEM with Qwen3-VL-8B as the answering model. Tok./q and Req./q are tokens and model requests per question, Wall/q is end-to-end per-question latency, and LLM is the part of that latency spent inside the model call. N_{QA} is the number of evaluated questions. Questions are answered serially within a run, so Wall/q is measured per-question latency rather than throughput divided by question count.

Benchmark	N_{QA}	Tok./q	Req./q	Wall/q (s)	LLM (s, %)
EgoLifeQA	500	3459	0.94	2.24	1.10 (49%)
Ego-R1 Bench	300	3473	0.96	1.85	0.66 (36%)
Video-MME (L)	900	5630	1.25	4.99	1.54 (31%)

Where the time goes. The last column of Table 8 shows that with a locally served 8B backbone, the model call accounts for 31% to 49% of end-to-end latency; the rest is memory retrieval, frame extraction, and encoding. This breakdown identifies substantial local processing overhead alongside model inference. Table 2 reports total per-question evaluation costs for a different answering configuration without decomposing API time, so its wall times cannot establish the same breakdown.

The modality gate makes latency bimodal. Video-MME (L) latency is bimodal: the median is 0.71 s, the mean 4.99 s, and the p95 20.4 s. The 675 text-only questions average 0.98 s and 3362 tokens, with a 0.70 s median; the 225 gated questions average 17.00 s and 12433 tokens, with a 14.66 s median and 28.4 s p95. Visual escalation thus costs roughly 17× the time and 3.7× the tokens on one quarter of the questions, making the overall mean unrepresentative of either route. EgoLifeQA and Ego-R1 Bench have narrow latency distributions and p95 near 3.5 s.

Comparison with direct frame prompting. Direct prompting of Qwen3-VL-8B with uniformly sampled frames costs 7153 tokens and 6.65 s per question on EgoLifeQA, 1317 tokens and 1.36 s on Ego-R1 Bench, and 4684 tokens and 3.63 s on Video-MME (L). On EgoLifeQA, SEEM uses 2.1× fewer tokens and 3.0× less wall time by retrieving event-anchor text instead of processing frames from a full day. Memory costs more on Ego-R1 Bench (2.6× the tokens and 1.4× the time) and Video-MME (L) (1.2× and 1.4×, respectively). The cost advantage is clearest for repeated queries over long egocentric recordings, where direct prompting re-encodes frames for each question.

C.3 Cost Accounting: Memory Construction and Evaluation

Table 9 separates offline construction from 500-question EgoLifeQA evaluation with GPT-5 mini. SEEM constructs memory from 30-second captions and event-anchor rewrites. WorldMM's evaluation includes index construction and question answering; indexing is distinct from offline construction and counted once. Combined construction and evaluation use 18.72M tokens for SEEM and 36.77M for WorldMM, excluding policy-search costs.

Table 9: Offline construction, evaluation, and combined cost comparison between WorldMM and SEEM on EgoLifeQA (500 questions) with GPT-5 mini.

Phase	Method	Requests	Total Tokens	Time (s)
Offline construction	WorldMM	28,900	25.15M	13,320
	SEEM	12,446	17.08M	2,541
Evaluation (500 questions)	WorldMM	15,233	11.62M	61,594
	SEEM	485	1.647M	1,375
Total	WorldMM	44,133	36.77M	74,914
	SEEM	12,931	18.72M	3,916

By deferring relational reasoning to retrieval over structured text, SEEM removes offline information extraction and semantic graph consolidation, more than halving per-window model calls (Table 10).

Table 10: Comparison of memory construction complexity per 30-second video window.

Method	Operations per 30s window	Calls
WorldMM	Caption (1) + OpenIE (2) + Semantic Ext. (0.1) + Consolidation (1.45)	$\sim$4.6
SEEM (Ours)	Caption (1) + Event-Anchor Rewriting (1)	**2.0**

D Limitations and Discussion

Despite its efficiency at test time, SEEM currently relies on offline caption generation for initial memory ingestion, and its search was bounded by modular policy templates. In future work, we plan to extend LVMH towards online streaming memory construction for continuous real-time video streams, expand the harness search space to discover dynamic multi-scale hierarchical graphs, and deploy the evolved policies to physical embodied robotic platforms.

Although LVMH does not fully resolve development-set overfitting, our experiments and ablation study show better generalization than Meta-Harness and the single-agent baseline in the reported runs. An important contribution of this work is to highlight development-set overfitting in automated harness evolution, emphasize its impact on policy search, and introduce a framework that improves generalization in our evaluations. The information available to a coding agent, including task examples, execution traces, and evaluation feedback, can strongly shape its hypotheses, evolution trajectory, and resulting policies. Restricting direct access to evaluation examples therefore does not by itself ensure generalization. At a conceptual level, harness self-evolution by coding agents resembles neural-network optimization: evaluation feedback serves as an objective signal analogous to a loss, while the agent interprets its observations to propose update directions, rather than computing gradients. This perspective makes the design of agent observations an important part of the optimization process. We view understanding this process as an important research direction and plan to systematically study how the content and granularity of observations affect policy evolution, overfitting, and generalization, with the goal of developing harnesses that generalize more reliably.