MEMLIFE: CURATING AND REASONING OVER LONG-TERM EGOCENTRIC VIDEO MEMORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term egocentric video enables personalized AI assistants to reason about daily life. However, as video histories grow to hundreds of hours spanning months or years, reprocessing raw clips for every query becomes computationally prohibitive. Memory systems offer a scalable alternative by compacting videos into text representations, but often fail on practical benchmarks: either the memory does not preserve key evidence, or the retriever fails to locate relevant entries due to retrieval competition in growing search spaces. To address these challenges, we introduce MEMLIFE, a multimodal memory system that constructs entity-grounded, first-person text episodes and retrieves them via a time-indexed agentic reader. Without training or query-time video access, MEMLIFE improves over the strongest training-free baseline by 4.6–12.0% across four long-horizon benchmarks. To further improve memory quality, we propose MEMOPT, a reinforcement learning framework that optimizes the memory writer to produce faithful, informative, and retrievable memories. MEMOPT consistently improves MEMLIFE by 2.7–5.0% across different video and question distributions, with gains that generalize across writer and reader backbones and memory systems.

1 INTRODUCTION

If an AI assistant could record a user's life experiences as egocentric videos, captured by wearable devices such as smart glasses and GoPro cameras (Grauman et al., 2022), *could it then answer any question the user asks about their past?* At first glance, this is a Retrieval-Augmented Generation (RAG) problem over egocentric videos (Yang et al., 2025; Alam et al., 2026). However, question answering (QA) over such *dense* memories is substantially harder. On the data side, videos accumulate over time to *prohibitive volumes*, and scenes and events *recur with subtle differences*. On the system side, the *latency budget* for sifting through long, similar memories is tight, and the *context window* is too small to hold many visual frames. It is therefore crucial to *compact raw videos into efficient representations* that can serve as the primary source of evidence for downstream tasks. Natural-language text stands out as an appealing choice: orders of magnitude more compact than visual frames, natively consumable by Large Language Models (LLMs), and interpretable to users.

We follow the line of work that converts multimodal video recordings into textual descriptions (Islam et al., 2024; Yang et al., 2025), either to facilitate retrieval of key evidence (Luo et al., 2025; Ren et al., 2026) or to directly support downstream generation (Yeo et al., 2026; Yin et al., 2026). However, deciding what to write poses a fundamental tradeoff (Figure 1): *aggressive* compression may discard information that future questions ask about, whereas *conservative* compression retains trivial details that dilute retrieval and inflate QA-time processing. Worse, writers may *hallucinate* content unsupported by the video, planting false evidence. In this paper, we address a critical question for QA over long-term memory: how can a system strike the right balance to *remember only what is worth remembering, faithfully,* and *in a form that is easy to recall*?

We answer this question in two steps: we first design a strong writer by hand, then learn a better one. Our first contribution, MEMLIFE, is an agentic memory system whose writer follows two principles. First, it *anchors memories in time and grounds entities across modalities*, aligning spoken references with the people and objects observed in each episode. Second, it *narrates in the first person*, matching how users phrase questions about their own lives (e.g., "Where did I put my passport?") and thereby narrowing the query-memory gap in agentic retrieval. To exploit these

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.

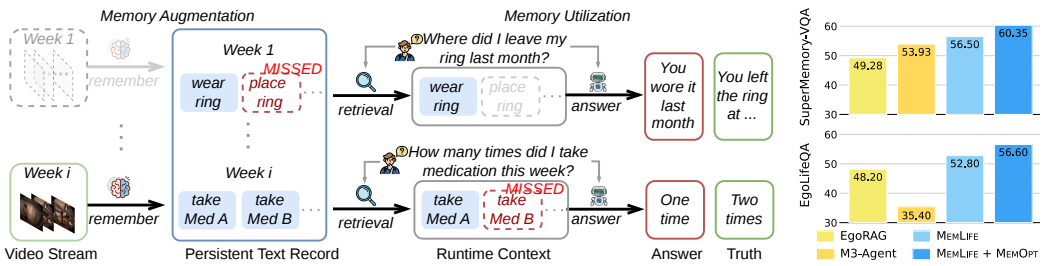


Figure 1: Failure modes of long-term video memory systems. Dotted objects denote lost information (e.g., Week-1 video is deleted). Systems fail when relevant evidence is omitted during memory writing or missed during retrieval. Our proposed solutions outperform prior state-of-the-art systems.

memories, MEMLIFE's reader combines agentic semantic search with time-scoped memory fetching, and presents retrieved episodes in chronological order for reasoning over long histories. In the presence of source videos, the agentic reader selectively invokes video-retrieval tools to sample raw video frames whenever visual details are required. Without writer training or query-time video access, MEMLIFE improves over the strongest training-free baseline by up to 12.0% in accuracy.

Our second insight is that *learning what to write* is, by itself, a powerful lever for long-term memory QA. We therefore take a bold step: rather than applying Reinforcement Learning (RL) to improve final answer quality (Guan et al., 2026; Yan et al., 2026; Wang et al., 2025b; Li et al., 2026), we apply RL *only* to memory writing, and examine whether this alone improves memory QA. We propose MEMOPT, a learning framework that rewards *Faithful*, *Informative*, and *Retrievable* Memories, a reward system we call *FIRM*. Faithfulness penalizes hallucinated memories unsupported by the source video; informativeness encourages the writer to preserve details critical to answering future questions; and retrievability favors concise descriptions that enable easy and precise retrieval. Although MEMOPT trains only the writer and does not rely on supervision from stronger models (Long et al., 2026; Zou et al., 2026), it further improves MEMLIFE by 2.7–5.0%.

Our paper makes the following contributions.

- We introduce MEMLIFE, a multimodal memory system that compresses egocentric videos into time- and entity-anchored, first-person episodes, and reasons over them with a versatile agentic reader.
- We propose MEMOPT, a learning framework whose FIRM reward optimizes the memory writer with multi-granular feedback on faithfulness, informativeness, and retrievability, to address the write-time failure modes we identified.
- We show that MEMOPT combined with MEMLIFE outperforms the strongest prior baseline by 4.0–17.0%, while reducing the memory size by up to 31× (Appendix E). The trained writer transfers well: it consistently improves accuracy when plugged into other memory systems (e.g., EgoRAG) and backbones, and on out-of-domain videos and question types.

2 RELATED WORK

Writing memory over extended video horizons. Memory-augmented video paradigms differ in how they convert continuous streams into persistent representations. Retrieval-oriented systems encode local clips as flat text descriptions or visual-text indexes (Fan et al., 2025; Islam et al., 2024; Luo et al., 2025). To control memory growth, streaming architectures maintain fixed-budget recurrent buffers that continuously compress past frames (Guan et al., 2026; He et al., 2024; Qian et al., 2024; Song et al., 2024; Jin et al., 2025), though they often struggle to preserve distant or fine-grained details. Hierarchical frameworks summarize video streams across temporal tiers (Islam et al., 2024; Yang et al., 2025), while structured memory agents organize representations around entities, events, or scene graphs (Goletto et al., 2025; Long et al., 2026; Ren et al., 2026; Yeo et al., 2026; Yin et al., 2026). However, high-level abstractions suffer from information loss, and graph maintenance becomes computationally prohibitive. Furthermore, prior memory construction relies on heuristic prompt engineering rather than optimizing memory generation via task feedback.

Reading memory across multi-session histories. Personal memory systems rely on readers to locate evidence scattered across extended temporal histories. Passive retrieval-based readers execute single-shot semantic or temporal queries over text indices (Lewis et al., 2020; Luo et al., 2025; Yang et al., 2025; Ren et al., 2026). Recent egocentric architectures introduce specialized access mechanisms—such as Memory Pointer Prompting (Ye et al., 2025) or multi-turn reasoning loops that reformulate queries, fetch time intervals, and inspect visual frames (Yeo et al., 2026; Yin et al., 2026; Gao et al., 2023; Wang et al., 2025a; Tian et al., 2026; Zhang et al., 2025). While effective during inference (Chandrasegaran et al., 2024; Wang et al., 2026b), using multi-turn readers during writer post-training conflates memory quality with reader execution noise. Because task accuracy depends on sampled tool calls, end-to-end task rewards provide a noisy, computationally prohibitive supervision signal for writer optimization.

Training video memory writers. To move beyond heuristic prompt design, recent paradigms train memory writers using learned policies. One direction relies on supervised fine-tuning or imitation learning, distilling memory construction from proprietary demonstrations or QA instructions (*e.g.*, EgoButler (Yang et al., 2025), M3-Agent (Long et al., 2026)). Another direction employs reinforcement learning, optimizing policies like TaskMem (Zou et al., 2026) and VST (Guan et al., 2026) against downstream task accuracy or QA preferences. However, static distillation restricts writer adaptability, while training purely on end-to-end task rewards introduces severe execution noise from reader reasoning. In contrast, MEMOPT provides supervision from verified evidence, source video, and fixed reader actions, decoupling writer post-training from reader execution noise.

3 METHODOLOGY

3.1 PROBLEM DEFINITION AND SOLUTION OVERVIEW

We begin by defining the *Video Memory QA* problem. Consider a stream of video (optionally egocentric) $\mathcal{V} = (c_1, \ldots, c_T)$ of T segments. Each segment can be represented as a triplet $c_t = (F_t, S_t, \tau_t)$, where F_t denotes the sampled visual frame, S_t denotes the aligned audio transcriptions, and $\tau_t = [s_t, e_t]$ denotes the starting and ending time. Memory QA takes a question q arriving at time τ_q, and provides the answer based on the prior memory fragments:

$$\mathcal{V}_q = \{c_t \in \mathcal{V} : e_t \leq \tau_q\}. \tag{1}$$

Our first solution MEMLIFE converts a video stream into persistent episodic memory and uses an agentic reader to retrieve and reason over relevant entries. Formally, MEMLIFE employs a writer W_θ with parameters θ; the writer generates a textual description d_t for each memory segment m_t:

$$d_t \sim W_\theta(\cdot \mid F_t, S_t), \qquad m_t = (d_t, \tau_t). \tag{2}$$

Thus, the memory repository stores $\mathcal{M}(\mathcal{V}) = \{m_t\}_{t=1}^{T}$. For a question q, question answering uses the available memory $\mathcal{M}_q$, optionally with the corresponding source-video history $\mathcal{V}_q$, where

$$\mathcal{M}_q = \{m_t \in \mathcal{M}(\mathcal{V}) : e_t \leq \tau_q\}. \tag{3}$$

Our second solution, MEMOPT, is a training framework that improves the memory writer by optimizing its parameters θ. Figure 2 gives the overview of the two solutions.

3.2 MEMLIFE SYSTEM FOR MEMORY WRITING AND READING

Writer: MEMLIFE separates question-independent memory construction from question-dependent memory access. Because future questions are unknown during writing, each entry must preserve information across modalities and express it in a form that supports later retrieval. The MEMLIFE writer applies three designs for this purpose.

Multimodal fusion. Information needed by future questions may appear in either the visual stream or speech. The writer therefore jointly interprets the frames and transcript such that evidence from both modalities can be preserved in one memory entry.

Entity grounding. A transcript may mention an entity by its name, which provides an important cue for future QA. The writer aligns the speech with the memory segment and uses the identified name to refer to the visual referents in the description.

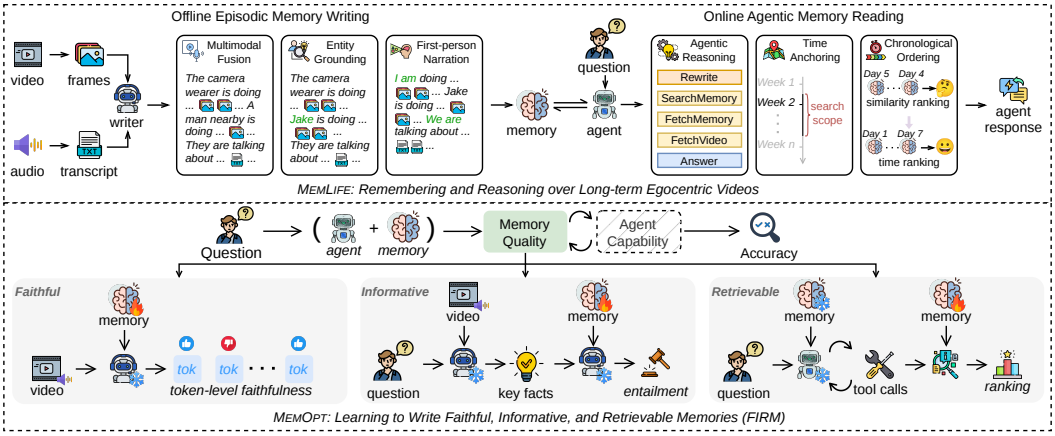


Figure 2: Overview of MEMLIFE's episodic writer and time-indexed agentic reader, together with MEMOPT's decomposed writer supervision. Direct source-video access is optional for the reader.

First-person narration. Memory questions for egocentric videos naturally refer to the user as "I." The writer therefore adopts the same first-person perspective, making its descriptions easier to match future egocentric questions.

MEMLIFE generates a description and its embedding for every clip independently, such that it avoids sequential dependencies and error propagation, and keeps total computation and storage linear in the recorded history. We also explored conditioning the writer on textual or multimodal context from the preceding segments, but neither variant improves aggregate accuracy (Appendix I).

Reader: The MEMLIFE reader is an agentic system that takes a question q and its timestamp τ_q as input and operates over multiple rounds by selecting actions from the action space $\mathcal{A}$:

$$\mathcal{A} = \{\textsc{Rewrite}(u, I|q, \tau_q),\ \textsc{SearchMemory}(\bar{M}|u, I, k),\ \textsc{FetchMemory}(\bar{M}|I),\ \textsc{FetchVideo}(\bar{V}|I, f),\ \textsc{Answer}(a|q, \bar{M}, \bar{V})\}. \tag{4}$$

With REWRITE, the agent reasons over the current information, and transforms the input (q, τ_q) into a targeted search query u and/or a time interval I over the history. For a search query u, SEARCHMEMORY conducts the similarity search and returns k relevant entries $\bar{M}$ from the stored memories within interval I. With only interval I, the agent can call either FETCHMEMORY, which returns all text memory entries $\bar{M}$ within I, or FETCHVIDEO, which returns raw multimodal fragments $\bar{V}$ and f sampled frames within I. Finally, ANSWER generates the answer a based on retrieval results.

The FETCHVIDEO tool is disabled when source videos are unavailable at reading time. For video-available settings (MEMLIFE-V), we store low-resolution redacted videos due to storage and privacy concerns, and sample limited frames to optimize computation.

3.3 MEMOPT FRAMEWORK FOR OPTIMIZING MEMORY WRITER

MEMOPT updates the writer parameters θ through supervision, while keeping the agentic reader fixed. We next present our FIRM reward model, the major recipe to improve memory writing.

Theoretical foundation. The goal of MEMOPT is to teach the writer what is worth remembering and which form is easy to recall. We next show the theoretical quantification.

Let Q and A denote a random question and its answer, while $\mathcal{V}$, $\mathcal{M}$, and C_Q denote the available video history, its corresponding memory, and the context retrieved by the reader for Q. To isolate the quality of the written memory, we restrict the reader's evidence source to $\mathcal{M}$, excluding direct access to $\mathcal{V}$ that could otherwise bypass the memory. Because the writer rewrites $\mathcal{V}$ into $\mathcal{M}$, and the reader constructs C_Q only from $(Q, \mathcal{M})$, their joint distribution factorizes as

$$p(Q, A, \mathcal{V}, \mathcal{M}, C_Q) = p(Q, A, \mathcal{V}) \cdot p_\theta(\mathcal{M} \mid \mathcal{V}) \cdot p(C_Q \mid Q, \mathcal{M}). \tag{5}$$

Let $H(|)$ denote conditional entropy and $I(|)$ conditional mutual information. Under this factorization, the additional uncertainty about the answer A when using the retrieved context C_Q instead of the video history $\mathcal{V}$ decomposes exactly as

$$\underbrace{H(A \mid Q, C_Q) - H(A \mid Q, \mathcal{V})}_{\text{total information loss}} = \underbrace{I(A; \mathcal{V} \mid Q, \mathcal{M})}_{\Delta_{\text{inf}}\text{: memory writing loss}} + \underbrace{I(A; \mathcal{M} \mid Q, C_Q)}_{\Delta_{\text{ret}}\text{: memory recall loss}}. \tag{6}$$

The equation shows that answer-relevant information can be lost when the writer maps $\mathcal{V}$ to $\mathcal{M}$ (Δ_{inf}) and when the reader retrieves C_Q from $\mathcal{M}$ (Δ_{ret}). These gaps motivate the informativeness and retrievability rewards, but do not measure whether $\mathcal{M}$ is grounded in $\mathcal{V}$. Unsupported memory can distort the final answer, while final-answer error also reflects answerer reasoning. We therefore assess faithfulness directly against the source video. Appendix K provides the complete derivation.

Faithfulness. Faithfulness asks whether every claim in a candidate is supported by its source segment. Because unsupported content may occupy only a few tokens, the feedback must also identify where it occurs. For candidate $y_i = (y_{i,1}, \ldots, y_{i,L_i})$ with L_i generated tokens under examination, we prompt the same frozen model to check the generated memory against its source segment by reproducing supported content exactly and minimally correcting unsupported spans, thereby localizing the grounding feedback. We denote by $p_{\text{faith}}(w \mid c, y_i, y_{i,<t})$ the probability that the evaluator generates a possible next token w and compute the faithfulness reward as

$$R_{\text{faith},i,t} = 1 - \left[\max_w p_{\text{faith}}(w \mid c, y_i, y_{i,<t}) - p_{\text{faith}}(y_{i,t} \mid c, y_i, y_{i,<t})\right]. \tag{7}$$

The reward lies in $[0, 1]$. It equals 1 when the candidate token is the evaluator's most probable continuation and decreases when the evaluator favors a correction.

Informativeness. Informativeness asks whether the candidate itself preserves the answer-relevant evidence supplied by its source segment, independent of reader behavior. We represent the required evidence as a source-grounded key fact and test whether the candidate entails it. A frozen copy of the default writer model serves as both the key-fact *extractor* and entailment *judge*. The question and answer are used only to construct the training reward, leaving the writer question-independent.

Formally, let $c = c_t$ be a segment, y be a candidate memory generated by the writer, $\mathcal{Q}(c)$ denote the set of questions for which c provides verified evidence, and assume each question $q \in \mathcal{Q}(c)$ is paired with a correct answer a_q. For each (q, a_q), the *extractor* identifies the observed fact $k_{c,q}$ that supports the answer. Let $P_{\text{ent}}(y \Rightarrow k_{c,q})$ denote the *judge*'s estimated probability that y entails this fact. We average this probability across the relevant questions,

$$R_{\text{inf}}(c, y) = \frac{1}{|\mathcal{Q}(c)|} \sum_{q \in \mathcal{Q}(c)} P_{\text{ent}}(y \Rightarrow k_{c,q}). \tag{8}$$

Retrievability. Retrievability asks whether the relevant memory can be discovered at QA time. For a pair (c, y), we approximate Δ_{ret} by checking whether y is returned for each question in $\mathcal{Q}(c)$. For efficiency and stability, at the start of each epoch, we run the MEMLIFE reader on every training question and cache its memory-access actions. For each candidate y, we replay these fixed actions to obtain the returned context $\mathcal{C}_q(y)$ without rerunning reader reasoning. The retrievability reward is

$$R_{\text{ret}}(c, y) = \frac{1}{|\mathcal{Q}(c)|} \sum_{q \in \mathcal{Q}(c)} \mathbf{1}[y \in \mathcal{C}_q]. \tag{9}$$

Multi-granular group-relative optimization. For each training segment c, the writer samples a group $g = \{y_i\}_{i=1}^G$ of G candidates. MEMOPT combines the three dimensions in FIRM multiplicatively and obtains the reward of token t in candidate i as

$$x_{i,t} = R_{\text{faith},i,t} R_{\text{inf}}(c, y_i) R_{\text{ret}}(c, y_i). \tag{10}$$

A token receives high credit only when it is faithful and its memory is informative and retrievable.

Standard group-relative optimization normalizes one scalar reward per candidate and broadcasts the resulting advantage to all of its tokens. MEMOPT must instead preserve variation from the token-level faithfulness signal. For candidate y_i with L_i generated tokens, we compute

$$\bar{x}_i = \frac{1}{L_i} \sum_{t=1}^{L_i} x_{i,t}, \qquad \mu_g = \frac{1}{G} \sum_{i=1}^{G} \bar{x}_i, \qquad \sigma_g^2 = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{L_i} \sum_{t=1}^{L_i} (x_{i,t} - \mu_g)^2. \tag{11}$$

Averaging within each candidate before computing the group statistics prevents longer memories from dominating the normalization. The token-level advantage is

$$\widetilde{A}_{i,t} = (x_{i,t} - \mu_g)/(\sigma_g + \epsilon_{\text{n}}), \tag{12}$$

where ϵ_{n} stabilizes normalization. Training then follows GRPO (Shao et al., 2024).

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate on SuperMemory-VQA (Alam et al., 2026), EgoLifeQA (Yang et al., 2025), and two extended settings. SuperMemory-LVQA combines all ten SuperMemory-VQA histories while retaining the original test questions, expanding the retrieval space with cross-subject distractors. EgoLife-EQA tests questions about recurring events whose supporting evidence is manually verified against the source recordings. We train MEMOPT on SuperMemory-VQA subjects S1–S6, validate on S7–S8, and test on S9–S10. All other benchmarks are used for test only. More details about data are provided in Appendix A.

Models and baselines. Qwen3.5-9B is used as the backbone for both writer and reader across systems. The generalizability study also evaluates Qwen3.6-27B. Training-free baselines include Video ReCap (Islam et al., 2024), EgoRAG (Yang et al., 2025), Video-RAG (Luo et al., 2025), VideoARM (Yin et al., 2026), EGAgent (Rege et al., 2026), and WorldMM (Yeo et al., 2026). Trained baselines include EgoButler (Yang et al., 2025), VST (Guan et al., 2026), TaskMem (Zou et al., 2026), and M3-Agent (Long et al., 2026). Appendix B provides further details.

Sections 4.2, 4.3, 4.4, 4.5 address the following research questions (RQs):

- **RQ1.** Does MEMLIFE outperform existing systems on long-term egocentric video memory question answering? Does MEMOPT further improve performance?
- **RQ2.** Do MEMLIFE and MEMOPT actually improve memory quality?
- **RQ3.** How generalizable is MEMOPT and its trained writer?
- **RQ4.** Is each component in MEMLIFE and MEMOPT important?

Additional experiments and analyses can be found in the Appendix.

4.2 PERFORMANCE COMPARISON TO BASELINES

Among systems without training, MEMLIFE outperforms baselines in accuracy across benchmarks in Table 1. Enabling video access through MEMLIFE-V produces only modest changes, showing that the gains do not depend on revisiting the original recordings. On SuperMemory-LVQA, the methods maintain accuracy close to their SuperMemory-VQA results despite lower annotated recall. While having cross-subject distractors, SuperMemory-LVQA may also contain subject interactions that provide useful context outside annotated evidence, which explains the accuracy-recall inconsistency.

Table 1: Comparison with existing video memory systems. Oracle Context bypasses retrieval by supplying all annotated source-video evidence directly to the reader. MEMLIFE-V permits source-video access. Bold and underlined values mark the best and second-best results within each group.

Method	SuperMemory-VQA		EgoLifeQA		SuperMemory-LVQA		EgoLife-EQA	
	Accuracy	Recall	Accuracy	Recall	Accuracy	Recall	Accuracy	Recall
Reference								
Oracle Context	67.58	100.00	66.20	100.00	67.58	100.00	61.00	100.00
Without Memory-Writer Training								
Video ReCap	36.28	54.01	35.20	27.40	39.17	16.79	38.00	27.00
EgoRAG	49.28	66.79	48.20	29.60	47.51	<u>47.90</u>	38.00	13.00
Video-RAG	43.98	59.54	38.80	20.80	49.28	35.31	28.00	6.00
VideoARM	39.33	70.23	36.00	45.60	40.93	9.35	40.00	**58.00**
EGAgent	43.66	48.09	34.00	21.80	41.73	26.53	26.00	16.00
WorldMM	42.05	74.62	42.20	43.80	45.91	24.05	27.00	35.00
MEMLIFE	<u>56.50</u>	<u>80.53</u>	<u>52.80</u>	<u>48.40</u>	<u>56.18</u>	46.18	**52.00**	44.00
MEMLIFE-V	**57.78**	**84.35**	**53.20**	**50.20**	**57.95**	**48.47**	<u>50.00</u>	<u>45.00</u>
With Memory-Writer Training								
EgoButler	36.92	49.05	44.00	25.00	38.68	36.64	34.00	18.00
VST	37.56	56.87	33.60	3.40	31.94	0.00	36.00	5.00
TaskMem	50.88	65.08	46.20	40.20	47.83	29.96	35.00	22.00
M3-Agent	53.93	54.20	35.40	9.60	54.90	13.74	23.00	4.00
MEMLIFE + MEMOPT	<u>60.35</u>	<u>81.49</u>	<u>56.60</u>	<u>50.60</u>	<u>58.91</u>	<u>43.13</u>	**57.00**	<u>41.00</u>
MEMLIFE-V + MEMOPT	**60.83**	**83.78**	**57.60**	**53.00**	**61.96**	**49.62**	**57.00**	**43.00**

With writer training, MEMOPT improves MEMLIFE accuracy on all benchmarks and outperforms every trained baseline. Although trained only on SuperMemory-VQA, it also improves MEMLIFE performance on the other three benchmarks, demonstrating transfer across video distributions, question types, and memory scales. Recall changes are mixed, indicating the accuracy gains also reflect more answer-useful memory content rather than only retrieving annotated evidence.

4.3 ANALYSIS OF MEMORY QUALITY

To separate retrieval from the quality of stored memory, Figure 3 reports retrieval recall, standard answer accuracy under normal retrieval, and oracle accuracy when evidence-aligned entries are given directly to the reader. Compared with EgoRAG, MEMLIFE improves recall in every category and raises oracle accuracy overall and in most categories, indicating gains in both retrieval and memory content. MEMOPT provides category-dependent gains, improving retrieval for some question types and the answer usefulness of stored content for others. Interestingly, on RelationMap, EgoRAG matches the optimized MEMLIFE system in standard accuracy despite lower recall and oracle accuracy. Its normal retrieval may therefore surface alternative useful context outside the annotations, while its evidence-aligned memories do not reliably preserve the relations needed for answering.

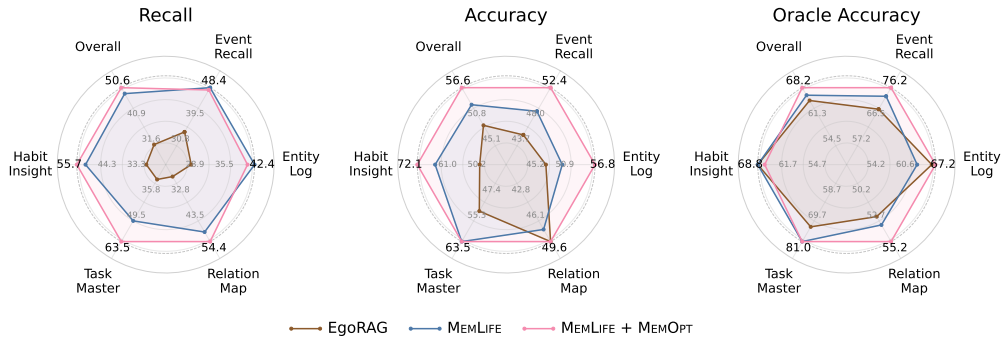


Figure 3: Memory quality across EgoLifeQA question categories. Oracle accuracy is measured by providing memory entries aligned with annotated evidence directly to the reader.

Table 2 illustrates how MEMOPT changes the stored content. In the first example, the untrained writer mistakes lentils for corn, while MEMOPT corrects the object without losing the surrounding action. In the second, the original memory uses a vague pronoun and omits the relevant food, whereas MEMOPT identifies the person and records the baked chicken needed to answer the question. These examples show that MEMOPT removes unsupported details while making answer-relevant entities and events explicit, improving both faithfulness and informativeness.

Table 2: Examples of memory corrections learned through MEMOPT. Red highlights errors and yellow highlights corrected content.

Frames		
Question	Q: *[...] Did I add the milk before or after the eggs when making the batter?* A: *You did not add milk or eggs to the batter; you only added lentils, salt, and spices.*	Q: *[...] Did I set a reminder for a backup dinner plan?* A: *No [...] However, you did mention earlier that you have baked chicken available.*
MEMLIFE	I am in a kitchen [...] transfer yellow corn kernels from a small food processor bowl [...] move toward the sink area [...]	[...] preparing food [...] they respond to my question about being hungry by saying they can wait for the food [...]
MEMLIFE + MEMOPT	I am in a kitchen [...] yellowish-orange granular material, which appears to be cooked lentils [...] to the sink area [...]	[...] I am preparing a meal, specifically baked chicken, and offering it to B, who indicates they are not hungry and can wait.

4.4 GENERALIZABILITY OF MEMOPT TRAINING

We then examine whether MEMOPT depends on the writer backbone by optimizing both Qwen3.5-9B and Qwen3.6-27B. Table 3 shows that MEMOPT improves every writer–reader pairing on both benchmarks. Changing the writer scale produces only modest differences, indicating that the training benefit does not depend on a particular writer backbone.

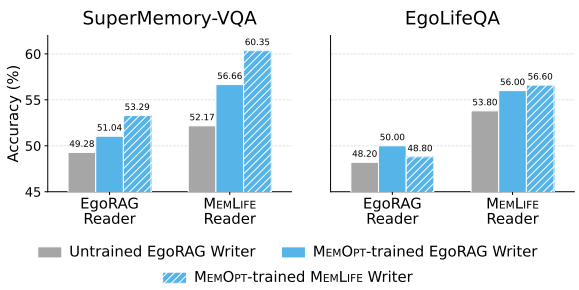


Figure 4: Generalizability of MEMOPT writer training across memory systems.

The Qwen3.6-27B reader further tests whether the optimized writer transfers beyond the Qwen3.5-9B reader used to collect retrievability supervision. Using the stronger reader substantially raises absolute accuracy for both writers. MEMOPT continues to improve every setting, although its gains become smaller with the stronger reader, suggesting that reader capacity can compensate for some deficiencies in written memory while writer optimization remains beneficial.

Table 3: Answer accuracy (%) across writer and reader backbones.

Writer	Training	Qwen3.5-9B Reader		Qwen3.6-27B Reader	
		SuperMemory-VQA	EgoLifeQA	SuperMemory-VQA	EgoLifeQA
Qwen3.5-9B	Zero-shot	56.50	52.80	66.93	59.60
	MEMOPT	**60.35**	**56.60**	**67.90**	**60.20**
Qwen3.6-27B	Zero-shot	57.14	55.20	66.77	59.80
	MEMOPT	**60.67**	**56.60**	**68.06**	**60.60**

Beyond backbone changes, we study whether MEMOPT transfers across memory systems. We optimize the EgoRAG writer and evaluate the original and optimized memories with both the native EgoRAG reader and the MEMLIFE agentic reader. We also evaluate the optimized MEMLIFE writer with the EgoRAG reader. Figure 4 shows that training improves the native EgoRAG pipeline on both benchmarks. Using MEMLIFE to read optimized EgoRAG memories provides further gains, while the highest performance is achieved with the trained MEMLIFE writer.

4.5 ABLATION STUDIES

To analyze how different components in our proposed methods contribute to the overall performance, we first ablate each component in the writer and reader designs of MEMLIFE, with the results shown in Table 4. For the writer design, we progressively add entity grounding and first-person narration to multimodal fusion, and examine how performance will change.

Table 4: Ablation studies on the writer and reader components in MEMLIFE. Accuracy is reported for SuperMemory-VQA and EgoLifeQA. Recall on the long-term EgoLifeQA task is also reported.

Writer Ablation								
Multimodal Fusion	Entity Grounding	First-person Narration	SuperMemory-VQA		EgoLifeQA		EgoLifeQA Recall	
			Zero-shot	MEMOPT	Zero-shot	MEMOPT	Zero-shot	MEMOPT
✔	✗	✗	52.33	56.98	52.80	52.60	48.60	47.00
✔	✔	✗	54.09	59.87	52.00	54.40	45.60	47.60
✔	✔	✔	56.50	60.35	52.80	56.60	48.40	50.60
Reader Ablation								
Agentic Reasoning	Time Anchoring	Chronological Ordering	SuperMemory-VQA		EgoLifeQA		EgoLifeQA Recall	
			Zero-shot	MEMOPT	Zero-shot	MEMOPT	Zero-shot	MEMOPT
✔	✗	✗	56.02	58.27	50.40	51.80	50.60	48.20
✔	✔	✗	56.50	58.59	50.60	54.00	50.00	50.00
✔	✔	✔	56.50	60.35	52.80	56.60	48.40	50.60

From the upper block of Table 4, we observe that entity grounding generally improves accuracy, particularly after MEMOPT training, but provides little benefit to recall. First-person narration further improves accuracy and consistently raises EgoLifeQA recall, supporting its role in aligning written memories with wearer-centered queries.

We then ablate the MEMLIFE reader by adding time anchoring and chronological ordering to agentic reasoning. The lower block of Table 4 shows that having time anchoring in the search tool provides its clearest benefit on optimized EgoLifeQA memories, where its accuracy gain is accompanied by higher recall. Chronological ordering further improves accuracy despite small or mixed recall changes, indicating that preserving event order primarily benefits reasoning over retrieved evidence. The complete reader performs best overall, with the largest gains appearing after writer optimization.

For the design of MEMOPT, we compare various supervision signals used to train the writer. Table 5 shows that teacher imitation with Qwen3.6-27B provides only modest gains, while final-answer accuracy supervision produces inconsistent changes across benchmarks. Among the proposed reward dimensions, retrievability alone raises recall on both benchmarks but does not consistently improve accuracy. Adding informativeness strongly benefits SuperMemory-VQA, although its gain does not transfer to EgoLifeQA. With faithfulness as a regularizer, the complete objective achieves the highest accuracy on both benchmarks while retaining higher recall than the untrained writer.

Table 5: Comparison of supervision signals used during memory writer training.

Writer Supervision					SuperMemory-VQA		EgoLifeQA	
Teacher	Accuracy	Retrievability	Informativeness	Faithfulness	Accuracy	Recall	Accuracy	Recall
✗	✗	✗	✗	✗	56.50	80.53	52.80	48.40
✔	✗	✗	✗	✗	57.78	83.59	53.00	49.00
✗	✔	✗	✗	✗	57.46	79.01	52.60	47.80
✗	✗	✔	✗	✗	53.45	82.63	54.40	**53.20**
✗	✗	✔	✔	✗	60.03	**84.35**	51.40	47.60
✗	✗	✔	✔	✔	**60.35**	81.49	**56.60**	50.60

Finally, we perform ablation studies on the faithfulness granularity and reward aggregation strategy used in MEMOPT. Following the probability-based judgment used for informativeness, the sequence-level variant assigns every token the same score about whether the complete memory is supported by its source segment. Different from multiplicative aggregation, the additive variant sums the three rewards. As shown in Table 6, under additive aggregation, token-level faithfulness maintains similar SuperMemory-VQA accuracy with slightly lower recall, but improves both metrics on out-of-domain EgoLifeQA. With token-level faithfulness fixed, multiplicative aggregation further improves accuracy on both benchmarks, and achieves the best performance overall.

Table 6: Ablation of faithfulness granularity and reward aggregation.

Training Design		SuperMemory-VQA		EgoLifeQA	
Faithfulness	Aggregation	Accuracy	Recall	Accuracy	Recall
Sequence-level	Additive	58.75	83.78	55.00	46.40
Token-level	Additive	58.91	81.68	56.00	49.40
Token-level	Multiplicative	60.35	81.49	56.60	50.60

5 CONCLUSION

We introduced MEMLIFE, an agentic memory system for long-term egocentric video that constructs time- and entity-anchored, first-person episodes and accesses them through time-scoped retrieval and chronological evidence organization. We further proposed MEMOPT, which applies reinforcement learning only to the memory writer through the FIRM objective for faithful, informative, and retrievable memories. MEMLIFE outperforms state-of-the-art training-free systems, while MEMOPT provides further gains that transfer across writer and reader backbones, memory systems, and out-of-domain video and question distributions. Together, the results demonstrate that learning what to remember is an effective and generalizable approach to long-term video question answering.

AI USE STATEMENT

We used generative AI tools solely to aid and polish the writing of this paper. All AI-assisted edits were reviewed by the authors, who take responsibility for the final content.

REPRODUCIBILITY STATEMENT

The appendix documents the data splits, baseline and evaluation protocols, model configurations, training procedure, checkpoint selection, and prompts needed to reproduce our results. We will release our code upon publication.

REFERENCES

Samiul Alam, Shakhrul Iman Siam, Michael J. Proulx, James Fort, Richard Newcombe, Hyo Jin Kim, and Mi Zhang. Supermemory-vqa: An egocentric visual question-answering benchmark for long-horizon memory. arXiv preprint arXiv:2606.00825, 2026.

Keshigeyan Chandrasegaran, Agrim Gupta, Lea M. Hadzic, Taran Kota, Jimming He, Cristobal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. Hourvideo: 1-hour video-language understanding. In *Advances in Neural Information Processing Systems*, 2024.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. arXiv preprint arXiv:2504.19413, 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 *Computer Vision – ECCV*, 2025.

Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn. arXiv preprint arXiv:2306.08640, 2023.

Gabriele Goletto, Tushar Nagarajan, Giuseppe Averta, and Dima Damen. Amego: Active memory from long egocentric videos. In *Computer Vision – ECCV*, 2025.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: Videollms can watch and think simultaneously. In *Computer Vision – ECCV*, 2026.

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

Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2025.

Taeil Kim, Kangsan Kim, and Sung Ju Hwang. Agent memory distillation: Empowering small llm agents with hierarchical teacher memory. arXiv preprint arXiv:2608.07169, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, 2020.

Yilong Li, Suman Banerjee, and Tong Che. Ember: Efficient memory via budgeted evidence retention for long-horizon agents. arXiv preprint arXiv:2606.05894, 2026.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *International Conference on Learning Representations*, 2026.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-rag: Visually-aligned retrieval-augmented long video comprehension. In *Advances in Neural Information Processing Systems*, 2025.

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, 2024.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. Secom: On memory construction and retrieval for personalized conversational agents. In *International Conference on Learning Representations*, 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 Annual ACM Symposium on User Interface Software and Technology*, 2023.

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

Aniket Rege, Arka Sadhu, Yuliang Li, Kejie Li, Ramya Korlakai Vinayak, Yuning Chai, Yong Jae Lee, and Hyo Jin Kim. Agentic very long video understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2026.

Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Zhiyu Shen, Ziming Wu, Fuming Lai, Shaobing Lian, and Yanghui Rao. MemBuilder: Reinforcing LLMs for long-term memory construction via attributed dense rewards. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2026.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Agentic chain-of-tool-thought for ultra-long egocentric video reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(10):12116–12131, 2026. doi: 10.1109/TPAMI.2026.3697367.

Juntong Wang, Haoyue Zhao, guanghui Pan, Xiyuan Wang, Yanbo Wang, Qiyan Deng, and Muhan Zhang. SAGE: A self-evolving agentic graph-memory engine for structure-aware associative memory. In *Frontiers in Graph Machine Learning for the Large Model Era*, 2026a.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *Computer Vision – ECCV*, 2025a.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-α: Learning memory construction via reinforcement learning. arXiv preprint arXiv:2509.25911, 2025b.

Ziyang Wang, Yue Zhang, Shoubin Yu, Ce Zhang, Zengqi Zhao, Jaehong Yoon, Hyunji Lee, Gedas Bertasius, and Mohit Bansal. Egomemreason: A memory-driven reasoning benchmark for long-horizon egocentric video understanding. In *Conference on Language Modeling*, 2026b.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In *Advances in Neural Information Processing Systems*, 2025.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2026.

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, Bo Li, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, et al. Egolife: Towards egocentric life assistant. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, Jiasen Lu, and Yinfei Yang. MMEgo: Towards building egocentric multimodal LLMs for video QA. In *International Conference on Learning Representations*, 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*, 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*, 2026.

Kai Zhang, Xinyuan Zhang, Hongda Jiang, Shiun-Zu Kuo, Hyokun Yun, Ejaz Ahmed, Shereen Oraby, Ziyun Li, Sanat Sharma, Ann Lee, Ahmed A Aly, Anuj Kumar, Raffay Hamid, and Xin Luna Dong. Salimory: Orchestrating cognitive memory for conversational agents. arXiv preprint arXiv:2606.04120, 2026.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 89863–89895. Curran Associates, Inc., 2025. doi: 10.52202/085713-3005. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/8190b210e9808e54ee16263b673a847d-Paper-Conference.pdf.

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

Tao Zou, Yichen He, Tian Qiu, Yuan Lin, and Hang Li. Task-focused memorization for multimodal agents. arXiv preprint arXiv:2605.31075, 2026.

A Dataset and Baseline Details

A.1 Datasets and Splits

SuperMemory-VQA contains ten subjects recorded across multiple sessions (Alam et al., 2026). We exclude 82 questions whose annotated evidence refers to source videos unavailable in the public release, leaving 4,771 questions. We partition these questions by subject, using S1–S6 for training, S7–S8 for validation, and S9–S10 for testing. These splits contain 3,425, 723, and 623 questions, respectively. EgoLifeQA contains 500 multiple-choice questions about a continuous seven-day recording (Yang et al., 2025). We use its complete question set only for testing.

SuperMemory-LVQA increases the retrieval space by concatenating the histories of all ten SuperMemory-VQA subjects into one memory store. It retains the 623 questions from the SuperMemory-VQA test split, with the remaining subject histories as additional distractors.

EgoLife-EQA contains 100 test questions about recurring events across multiple days in EgoLife. We identify candidate events from the official captions and formulate questions about them. Two annotators independently verify whether each candidate evidence segment supports its question's annotated answer, reaching 92.1% agreement. They resolve disagreements through discussion, and we discard questions without verified evidence or an unambiguous answer.

Table 7 groups the EgoLife-EQA questions into three categories. Frequency-counting questions ask on which days an activity occurred and may have multiple correct options, such as "*On which days during the seven-day period did I go grocery shopping? Choose all that apply.*" Routine questions ask what activity typically occurs at a given time of day, such as "*What do I usually check on my phone in the morning?*" Comparison questions ask which of two activities occurred on more days, such as "*Which activity did I do in the bedroom on more days—browsing social media or browsing products online?*"

Table 7: Composition of EgoLife-EQA by question type. Correct options, evidence segments, and distinct evidence days are averaged per question.

Type	# Questions	# Options	Avg. Correct Options	Avg. Evidence Segments	Avg. Distinct Days
Frequency counting	45	8	2.33	2.31	2.31
Routine	31	2–4	1.00	2.97	2.97
Comparison	24	3	1.00	8.04	5.54
All	100	–	1.60	3.89	3.29

Table 8 compares the scale of the accessible video history and evidence annotations across the four evaluation settings, where only SuperMemory-VQA supplies supervision for MemOpt.

Table 8: Dataset statistics. Video hours and time spans are averaged over the history available to each question. Evidence segments and their total duration are averaged per question.

Dataset	# Questions	Avg. Video Hours	Avg. Time Span (days)	Avg. Evidence Segments	Avg. Evidence Duration (s)
SuperMemory-VQA	4771	3.97	12.50	1.34	57.1
EgoLifeQA	500	22.65	2.80	1.10	32.3
SuperMemory-LVQA	623	47.90	127.36	1.24	56.3
EgoLife-EQA	100	43.06	6.33	3.89	116.7

A.2 Baselines

Without task-specific writer training. Video ReCap recursively builds clip-, segment-, and video-level captions by combining visual features with captions from the preceding hierarchy (Islam et al., 2024). EgoRAG builds clip-, hour-, and day-level memories and retrieves relevant

clips using visual and textual similarity (Yang et al., 2025). Video-RAG indexes OCR, ASR, and object-detection text, then provides retrieved text and sampled video frames to a VLM (Luo et al., 2025). The remaining systems perform adaptive multimodal access. VideoARM constructs a query-conditioned hierarchical memory online while inspecting progressively narrower regions of the source video (Yin et al., 2026). EGAgent plans over a temporal entity scene graph with visual-frame and transcript search (Rege et al., 2026). WorldMM iteratively retrieves from multiscale episodic and semantic graphs and a visual memory (Yeo et al., 2026).

With learned memory construction. EgoButler combines EgoRAG with EgoGPT, which is fine-tuned for both visual-audio captioning and question answering (Yang et al., 2025). VST trains a single streaming VideoLLM to generate both textual memory and final answers through supervised fine-tuning and answer-based reinforcement learning (Guan et al., 2026). TaskMem instead optimizes a memorization policy with model-judged quality rewards followed by task-relevance preference learning, leaving QA to a separate answer generator (Zou et al., 2026). M3-Agent trains an entity-centric episodic and semantic memory writer through imitation of synthetic demonstrations, then separately trains its memory-search controller with reinforcement learning (Long et al., 2026).

Evaluation protocol. For EgoButler, VST, and TaskMem, we use the released EgoGPT-7B, VST-32B, and TaskMem-30B checkpoints, respectively,[1] where some have much larger capacity than our tested 9B model. Because M3-Agent does not release a checkpoint, we reproduce its trained memorizer and controller with the same Qwen3.5-9B backbone as our method. All other replaceable model components also use Qwen3.5-9B. Appendix C shows additional experimental results on the reproduced training methods with matched backbone models and training data.

All methods receive the same questions, answer options, and causal video histories, and we recompute their metrics using a common answer parser. The *Oracle Context* reference in Table 1 bypasses retrieval by directly supplying the answerer with all annotated source-video evidence for each question. Its retrieval recall is therefore 100% by construction. Among baselines without task-specific writer training, Video-RAG, VideoARM, EGAgent, and WorldMM retain query-time access to visual evidence, whereas the default MEMLIFE operates only on written memory. Among learned systems, we implement VST and M3-Agent with models trained on both memory construction and their answering or control components. EgoButler, TaskMem, and MEMOPT instead adopt only a trained memory writer while keeping the downstream reader fixed.

A.3 EVALUATION METRICS

Let $\mathcal{Q}$ denote the test questions, A_q the set of annotated correct option labels, and $\widehat{A}_q$ the predicted set produced by the answer parser. These sets contain one label for single-choice questions. For multi-select EgoLife-EQA questions, a prediction is correct only if it exactly matches the complete annotated set. We compute answer accuracy over all test questions as

$$\mathrm{Acc} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}[\widehat{A}_q = A_q]. \tag{13}$$

Retrieval recall is computed over the subset $\mathcal{Q}_G \subseteq \mathcal{Q}$ containing questions with at least one annotated temporal evidence interval. Let G_q and H_q denote the source-indexed temporal intervals annotated for q and returned to the reader, respectively. A question counts as retrieved when at least one returned interval overlaps an annotated interval from the same source video. We compute

$$\mathrm{Recall} = \frac{1}{|\mathcal{Q}_G|} \sum_{q \in \mathcal{Q}_G} \mathbf{1}[\exists g \in G_q,\ h \in H_q \text{ such that } h \cap g \neq \varnothing]\,. \tag{14}$$

For a single-shot reader, H_q is its retrieved context. For an agentic reader, H_q is the union of evidence returned by all executed search and fetch actions. This union measures the evidence actually available during the interaction rather than evidence recoverable by an unexecuted query.

[1] Official checkpoints: EgoGPT-7B, VST-32B, and TaskMem-30B.

B Implementation Details

B.1 Models and Inference

Both the default MemLife writer and reader use Qwen3.5-9B. The backbone study additionally uses Qwen3.6-27B as the writer and reader, producing all four combinations of the two models. Each writer generates one memory store per dataset, and that same store is evaluated by both reader backbones. Swapping the reader therefore does not regenerate or alter the memory. All reader weights remain frozen, including during MemOpt training.

The writer processes each 30-second segment independently from eight frames at 704-pixel resolution and its speech transcript. A trailing segment shorter than one second is folded into the preceding segment. Memory descriptions are generated greedily with a maximum of 512 tokens. We embed each description using BAAI/bge-large-en-v1.5 and perform exact inner-product search over normalized embeddings.

At evaluation, the agentic reader can call SearchMemory and FetchMemory for at most ten rounds. Semantic search returns at most 32 entries, while interval-based fetching returns at most 64 entries. The default reader has no access to source video. The MemLife-V variant additionally allows the reader to inspect up to 50 sampled frames and the transcript from a selected temporal interval. Both writer and reader generations use greedy decoding for deterministic results. More details about the writer and reader prompts are in Appendix L.

B.2 MemOpt Training

All evaluators use Qwen3.5-9B, and both their parameters and the reader parameters remain frozen. We train the writer for three epochs and rebuild the validation memory after each epoch. We select the checkpoint with the highest sum of answer accuracy and retrieval recall on S7–S8, then evaluate it once on each test benchmark. Table 9 summarizes the training configuration.

The Kullback–Leibler regularizer is applied directly to the loss against the frozen initial writer rather than incorporated into the reward, so it does not affect the group-relative advantages. Training rollouts are sampled at temperature 1.0 to provide within-group variation, whereas deployment uses greedy decoding.

Table 9: MemOpt training hyperparameters.

Group	Parameter	Value
Sampling	Group size G (rollouts per segment)	5
	Rollout temperature / top-p / top-k	1.0 / 1.0 / disabled
	Maximum prompt length (tokens)	12,288
	Maximum response length (tokens)	512
Optimization	Learning rate	1×10^{-6}
	Weight decay	0.01
	Learning-rate warmup	none
	Gradient-norm clip	1.0
Objective	Policy clipping ratio ϵ_{p} (symmetric)	0.2
	Advantage clip κ	3.0
	KL penalty β (loss term)	0.01
	Entropy coefficient	0
Schedule	Segments per optimizer step	32
	Mini-batch size	16
	Inner epochs per step	1
	Training epochs	3

B.3 Retrievability Trace Collection and Replay

At the beginning of each training epoch, we build a memory bank with the current writer and run the frozen agentic reader on every training question associated with at least one verified evidence segment. For each question, we cache the search queries, time intervals, retrieval budgets, and fetched intervals issued before the final answer. These traces are reused to score all candidate memories sampled during that epoch.

To evaluate a candidate y for segment c, we replace only the corresponding entry in the memory bank and replay every cached action associated with questions in $\mathcal{Q}(c)$. For a search action, we recompute the candidate's similarity and rank it against all entries eligible under that action's causal and temporal constraints. The action returns y only when its rank falls within the recorded retrieval budget. For a fetch action, y is returned when the requested interval contains c. This replay determines membership in $\mathcal{C}_q(y)$ without executing a new agent reasoning trajectory. The memory bank and traces are rebuilt after each epoch.

B.4 Token-Level Group-Relative Objective

Equations 11 and 12 define the token advantages used for optimization. We clip each $\widetilde{A}_{i,t}$ to $[-\kappa, \kappa]$, then recenter and whiten the values over generated response tokens while excluding prompt and padding positions. We denote the resulting advantage by $A_{i,t}$. Let $\pi_{\theta_{\text{old}}}$ denote the policy that sampled the current candidates. Its token probability ratio with the updated writer is

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid c, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid c, y_{i,<t})}. \tag{15}$$

The MemOpt policy objective is

$$\mathcal{L}_{\text{MemOpt}}(\theta) = -\mathbb{E}_{c,i,t}\Big[\min\big(\rho_{i,t}(\theta)A_{i,t}, \text{clip}(\rho_{i,t}(\theta), 1-\epsilon_{\text{p}}, 1+\epsilon_{\text{p}})A_{i,t}\big) - \beta \widehat{D}^{\text{KL}}_{i,t}\Big], \tag{16}$$

where ϵ_{p} is the policy clipping ratio, $\widehat{D}^{\text{KL}}_{i,t}$ is the per-token Kullback–Leibler estimate between π_θ and the frozen initial writer π_{ref}, and β controls its strength. Table 9 lists the numerical settings.

C Controlled Comparison with Trained Baselines

The main comparison uses official checkpoints when available, preserving the systems released by their authors but leaving differences in model scale and training data. We therefore conduct an additional controlled comparison by reproducing VST and TaskMem with Qwen3.5-9B and training them only on the SuperMemory-VQA training split used by MemOpt. Their results consequently differ from the released-checkpoint results in Table 1. We repeat the M3-Agent results from that table because its existing reproduction already uses the same backbone and training split.

Table 10: Controlled comparison with trained baselines. All methods use Qwen3.5-9B and task-specific training data only from SuperMemory-VQA. EgoLifeQA is evaluated out of distribution, and bold marks the best result.

Method	SuperMemory-VQA		EgoLifeQA	
	Accuracy	Recall	Accuracy	Recall
VST	56.50	65.08	29.40	7.60
TaskMem	48.96	51.15	48.20	35.00
M3-Agent	53.93	54.20	35.40	9.60
MemLife + MemOpt	**60.35**	**81.49**	**56.60**	**50.60**

Table 10 shows that MemLife with MemOpt achieves the highest accuracy and recall on both the in-domain SuperMemory-VQA test set and out-of-domain EgoLifeQA. None of these controlled runs uses EgoLifeQA for task-specific training. Moreover, MemOpt updates only the memory writer, whereas VST and M3-Agent also adapt their answering or control components. Together with the primary comparison in Table 1, these results indicate that the gains from MemOpt are not explained by backbone scale or differences in task-specific training data.

D STABILITY ANALYSIS

In the main experiments, we use deterministic decoding across all methods for fairness and reproducibility. To assess stability under stochastic decoding, we repeat inference five times at temperature 1.0 for MEMLIFE, MEMLIFE-V, and EgoRAG, the competing system with the highest average accuracy. Table 11 compares the greedy results from the main evaluation with the mean and standard deviation over five sampled runs.

Table 11: Stability across reader decoding settings. $T = 0.0$ reports greedy decoding, while $T = 1.0$ reports mean $\pm$ standard deviation over five runs.

Method	SuperMemory-VQA				EgoLifeQA			
	Accuracy		Recall		Accuracy		Recall	
	$T = 0.0$	$T = 1.0$	$T = 0.0$	$T = 1.0$	$T = 0.0$	$T = 1.0$	$T = 0.0$	$T = 1.0$
EgoRAG	49.28	49.18 ± 0.95	66.79	66.79 ± 0.00	48.20	46.48 ± 0.95	29.60	28.80 ± 0.00
MEMLIFE	56.50	55.31 ± 1.53	80.53	80.69 ± 0.32	52.80	50.16 ± 1.83	48.40	47.48 ± 1.36
MEMLIFE-V	57.78	58.78 ± 0.41	84.35	84.58 ± 0.64	53.20	52.44 ± 1.72	50.20	49.72 ± 0.30
Method	SuperMemory-LVQA				EgoLife-EQA			
	Accuracy		Recall		Accuracy		Recall	
	$T = 0.0$	$T = 1.0$	$T = 0.0$	$T = 1.0$	$T = 0.0$	$T = 1.0$	$T = 0.0$	$T = 1.0$
EgoRAG	47.51	48.02 ± 0.43	47.90	48.85 ± 0.00	38.00	36.20 ± 2.39	13.00	14.00 ± 0.00
MEMLIFE	56.18	55.31 ± 0.91	46.18	45.00 ± 0.97	52.00	50.80 ± 4.66	44.00	42.20 ± 2.95
MEMLIFE-V	57.95	58.17 ± 1.24	48.47	48.51 ± 1.93	50.00	53.20 ± 3.27	45.00	45.00 ± 0.71

The performance ordering remains stable under sampled decoding. MEMLIFE and MEMLIFE-V outperform EgoRAG in accuracy on every benchmark and generally provide higher recall. The gains remain consistent across runs, and the accuracy improvements of MEMLIFE over EgoRAG are significant on all four benchmarks ($p < 0.05$). The recall of EgoRAG has a zero standard deviation, since it has a fixed retrieval process instead of using agentic search as MEMLIFE.

E EFFICIENCY ANALYSIS

Table 12 compares storage and question-time costs across various systems, which are the key efficiency concerns in runtime question answering. Storage is measured over matched source histories and normalized by video duration, while reading time excludes model initialization.

Table 12: Storage and reading efficiency of training-free video memory systems. Storage is normalized by source-video duration, and reading time is averaged per question. Dashes indicate methods without a persistent memory store.

Method	Memory Storage (MB/h)		Reading Speed (s/question)	
	SuperMemory-VQA	EgoLifeQA	SuperMemory-VQA	EgoLifeQA
Video ReCap	3.86	3.66	<u>20.4</u>	54.2
EgoRAG	4.98	5.49	33.2	<u>43.5</u>
Video-RAG	<u>1.03</u>	<u>3.44</u>	60.9	76.9
VideoARM	–	–	47.5	110.4
EGAgent	21.50	22.20	200.3	230.7
WorldMM	2.09	3.88	44.7	57.1
MEMLIFE	**0.68**	**0.67**	**13.1**	**20.7**

MEMLIFE has the smallest footprint among systems with persistent storage, which has nearly unchanged storage per video hour across the two benchmarks. It also provides the fastest reading time. Its informative text episodes and targeted retrieval typically expose sufficient evidence within a few agent rounds, while chronological ordering reduces the subsequent reasoning burden. The default reader also avoids processing source-video tokens at question time. Together, these choices limit

unnecessary generation and multimodal processing, helping explain the lower latency relative to baselines with different execution patterns.

F Ablation of Retrievability Approximation

Computing retrievability requires specifying how a future question will access a candidate memory. We compare two approximations. The simplified variant uses the original question directly as a fixed semantic-search query and scores whether the candidate is retrieved. It therefore requires neither a reader policy nor a reader rollout during writer training. Standard MemOpt instead runs the MemLife agentic reader at the beginning of each epoch and collects its memory-access actions. These actions reflect the agent's reformulated queries and time-scoped retrieval decisions, but remain fixed within the epoch to avoid candidate-dependent variation. After writer training, we evaluate both variants with the standard MemLife agentic reader, which performs multi-round memory access and chronologically orders the retrieved entries before answering.

Table 13: Effect of the training-time retrievability approximation on answer accuracy (%).

Training-time proxy	SuperMemory-VQA	EgoLifeQA
None (zero-shot)	56.50	52.80
Original question	57.62	55.40
Agent actions	60.35	56.60

Both approximations outperform the zero-shot writer on both benchmarks. The original-question proxy already provides useful supervision without requiring a training-time reader rollout. Agent actions perform best on both datasets, indicating that reformulated queries and time-scoped accesses expose retrieval failures that the original question alone may miss. We therefore use agent actions in the standard MemOpt configuration, while retaining the original-question proxy as a simpler, reader-policy-independent alternative.

G Reward Dynamics during MemOpt Training

To examine how the three reward signals evolve during optimization, we record their scores over three epochs, comprising 135 training steps. Retrievability and informativeness are averaged across the sampled candidate memories. Since faithfulness is evaluated at the token level, we first take the minimum token score within each candidate and then average these minima across candidates. This conservative aggregation prevents a few unsupported tokens from being obscured by many well-supported ones. In Figure 5, the light curves show the resulting per-step scores, while the bold curves show their smoothed trends.

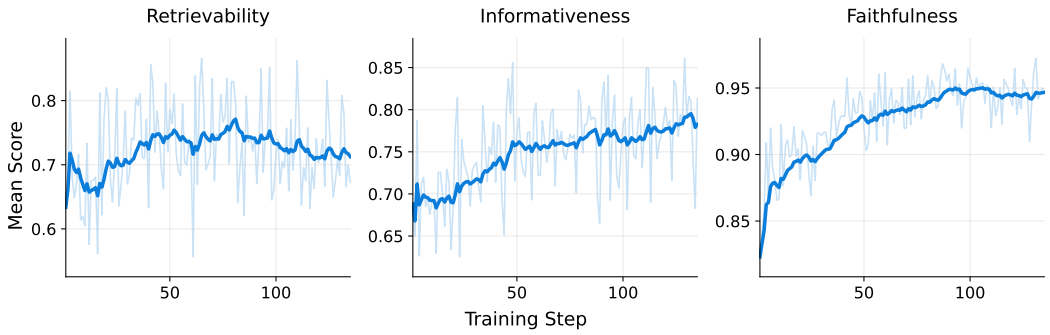


Figure 5: Dynamics of the three reward signals during MemOpt training.

Following an initial fluctuation, retrievability improves mainly during the early stages of training and stabilizes sooner, although its per-step values remain noisy because they depend on each candidate's

position within the evolving memory store and the cached reader actions. Informativeness and faithfulness improve for longer, indicating that the writer continues to preserve more answer-relevant evidence and produce better-grounded memories. Together, these trends suggest that MEMOPT improves all three dimensions without a sustained trade-off in retrievability.

H DISCUSSION ON MEMORY READER TRAINING

Although MEMOPT focuses on the writer, the agentic reader can also be trained to improve memory access and answer generation. We compare training neither component, the writer alone, the reader alone, and both components. For reader reinforcement learning, the reward is $(\text{accuracy} + \text{recall})/2$. To reduce training cost, we cap trajectories at six interaction rounds, compared with ten rounds at evaluation. A trajectory containing a formatting error or exhausting this budget will receive a reward of -1.

Table 14: Effects of writer and reader training. All values are percentages.

Writer trained	Reader trained	SuperMemory-VQA		EgoLifeQA	
		Accuracy	Recall	Accuracy	Recall
✗	✗	56.50	80.53	52.80	48.40
✔	✗	60.35	81.49	56.60	50.60
✗	✔	61.00	89.50	51.40	59.00
✔	✔	63.08	87.98	54.20	56.80

Reader training substantially improves both accuracy and recall on the in-domain SuperMemory-VQA benchmark. Adding writer training to the trained reader further improves accuracy with only a small decrease in recall, indicating that the optimized memories preserve more answer-useful information rather than merely maximizing evidence retrieval. On the out-of-domain EgoLifeQA benchmark, reader training still improves recall but does not consistently improve answer accuracy, whereas writer-only training achieves the highest accuracy. This contrast suggests that reader policies more readily specialize to the training question and source-video distributions, as well as the resulting interaction trajectories. Writer training instead improves the persistent memory itself, allowing different readers and evaluation settings to benefit from the same higher-quality memory or writing policy. We therefore center MEMOPT on writer optimization. Reader training provides complementary in-domain gains, while making these gains consistent under distribution shift remains future work.

I PREDECESSOR CONTEXT FOR MEMORY WRITING

MEMLIFE writes each video segment independently. To examine whether continuity across adjacent segments benefits memory construction, we compare this design with two alternatives that condition the writer on the immediately preceding segment. The text variant provides both the previous transcript and memory entry, while the multimodal variant additionally provides the previous frames. The independent variant (None) omits the predecessor block and uses the multimodal information from the corresponding segment as the only context.

Table 15: Answer accuracy across SuperMemory-VQA question types under different forms of predecessor context for memory writing. Bold values indicate the best result in each column.

Predecessor Context	Conversational Memory	Intent Recall	In-Context Retrieval	Timeline Reconstruction	Object-Location Memory	Visual Recall	Overall
None	63.41	77.42	**43.18**	**57.41**	**52.29**	44.12	**56.50**
Text	62.60	**86.02**	37.50	48.15	45.87	**45.10**	54.25
Multimodal	**69.92**	82.80	37.50	47.22	46.79	43.14	54.90

Table 15 shows that predecessor context produces mixed effects. Text context improves intent and visual recall, while multimodal context improves conversational and intent recall. Both variants reduce performance on in-context retrieval, timeline reconstruction, and object-location memory, and neither improves overall accuracy. We therefore retain independent segment writing, which also avoids sequential dependencies and error propagation during memory construction.

J Memory from Text and Multimodal Inputs

Text-input agent memory systems maintain persistent information from language-based histories, such as conversations, documents, or natural-language observations. They determine what information to store and how to update, organize, and retrieve it for later use. Existing systems store observations and synthesize reflections (Park et al., 2023), manage tiered memory stores (Packer et al., 2024; Kang et al., 2025), or maintain and consolidate information from conversations (Zhong et al., 2024; Chhikara et al., 2025). Other architectures organize histories through segmentation, compression, or dynamically linked and graph-structured memories (Pan et al., 2025; Xu et al., 2025; Wang et al., 2026a).

Beyond architectural design, recent methods learn what to retain or how to manage memory. They optimize memory operations or construction from downstream feedback (Yan et al., 2026; Wang et al., 2025b), retain source evidence under a fixed budget (Li et al., 2026), or provide denser supervision for different stages and components of memory construction and use (Shen et al., 2026; Zhang et al., 2026). Agent Memory Distillation follows a different, training-free strategy. It constructs hierarchical procedural memories from successful trajectories generated by a stronger teacher agent (Kim et al., 2026).

In these settings, the source experience has already been expressed as text. Long-term egocentric video introduces an additional challenge before memory management can begin. The writer must identify entities, actions, and speech from raw visual and audio observations, preserve their temporal context, and avoid introducing details unsupported by the source. Once evidence is omitted or incorrectly described at this stage, a downstream text-memory architecture cannot recover it without revisiting the video. Text-input memory systems could therefore complement MemLife by organizing the episodic descriptions after they are written, rather than replacing our system as video-to-text writers.

K Theoretical Foundation of FIRM

Let Q and A denote a question and its answer, V the available video history, M the memory written from that history, and C the retrieved context. Let $H(\cdot \mid \cdot)$ and $I(\cdot;\cdot \mid \cdot)$ denote conditional entropy and conditional mutual information. For the memory-only reader, their joint distribution factorizes as

$$p(V, Q, A, M, C) = p(V, Q, A)p_\theta(M \mid V)p(C \mid M, Q),$$

where θ denotes the writer parameters. This restates Equation 5. Because the writer constructs M only from V and the reader constructs C only from M and Q,

$$I(A; M \mid V, Q) = 0, \qquad I(A; C \mid M, Q) = 0.$$

The memory-quality decomposition then follows as

$$\begin{aligned}
&H(A \mid C, Q) - H(A \mid V, Q) \\
&= I(A; V \mid Q) - I(A; C \mid Q) \\
&= \big[I(A; V \mid Q) - I(A; M \mid Q)\big] + \big[I(A; M \mid Q) - I(A; C \mid Q)\big] \\
&= \big[I(A; V \mid M, Q) - I(A; M \mid V, Q)\big] + \big[I(A; M \mid C, Q) - I(A; C \mid M, Q)\big] \\
&= I(A; V \mid M, Q) + I(A; M \mid C, Q). \qquad (17)
\end{aligned}$$

Applying the chain rule to the two bracketed differences and substituting the conditional independences above yields the final line. Its two terms are the informativeness and retrievability gaps, respectively.

To relate these gaps to downstream QA, let $\hat{A} \sim q_\phi(\cdot \mid Q, C)$ denote the answerer's prediction. The quantity $q_\phi(A \mid Q, C)$ is therefore the probability assigned to the ground-truth answer, while $p_\theta(A \mid Q, C)$ is its conditional distribution under the joint process above. Taking expectations under $p_\theta(Q, A, V, M, C)$, the negative log-likelihood decomposes as

$$\begin{aligned}\mathcal{L}_{\mathrm{NLL}}(\theta, \phi) &:= \mathbb{E}_{p_\theta}\big[-\log q_\phi(A \mid Q, C)\big] \\ &= \underbrace{H(A \mid Q, V)}_{\text{irreducible uncertainty}} + \underbrace{I(A; V \mid Q, M)}_{\text{informativeness gap}} + \underbrace{I(A; M \mid Q, C)}_{\text{retrievability gap}} \\ &\quad + \underbrace{\mathbb{E}_{p_\theta(Q,C)}\, \mathrm{KL}(p_\theta(A \mid Q, C) \,\|\, q_\phi(A \mid Q, C))}_{\substack{\text{prediction mismatch} \\ \text{(includes faithfulness effects)}}}.\end{aligned} \tag{18}$$

Proof. For fixed (Q, C), averaging $-\log q_\phi(A \mid Q, C)$ over $A \sim p_\theta(A \mid Q, C)$ gives their cross-entropy. This equals $H(A \mid Q, C)$ plus the KL term in Equation 18. Equation 17 expands the conditional entropy as $H(A \mid Q, V) + I(A; V \mid Q, M) + I(A; M \mid Q, C)$, yielding the result. ■

The final term is affected by both the memory and the answerer. Unsupported content in C can shift q_ϕ away from p_θ and mislead the answerer, while reasoning errors can produce the same mismatch even when C is fully grounded. It therefore mixes a writer-controlled faithfulness effect with answerer-dependent noise and cannot directly supervise faithfulness. This explains why final-answer feedback gives noisy credit to the writer.

Together, these results motivate the three FIRM signals. Informativeness targets answer-relevant evidence lost during writing, and retrievability targets stored evidence missed during memory access. Faithfulness directly compares the memory with its source, isolating hallucination-related noise without absorbing answerer reasoning error.

For a rigorous assessment of memory quality, MEMOPT uses the default MEMLIFE reader during training and disables direct access to the source videos. This prevents the reader from bypassing the written memory when collecting answer-relevant information.

L PROMPT TEMPLATES

L.1 WRITER PROMPT

Figure 6 shows the complete system instruction and user template used by the MEMLIFE writer. The prompt asks it to preserve visible and spoken evidence, ground named entities in visual observations, and narrate the resulting memory in the first person.

L.2 READER SYSTEM PROMPTS

Figures 7 and 8 provide the agentic-reader prompts used in our experiments. The reader emits either one tool call or a final answer per round. Outputs that match neither format are treated as parse failures and scored as incorrect. All requested intervals are clipped to the question time, preventing access to future segments.

L.3 EVALUATOR PROMPTS

Figures 9–11 provide the exact templates for grounded key-fact extraction, informativeness evaluation, and faithfulness evaluation. The key-fact extractor and faithfulness evaluator receive eight frames and the transcript of the current 30-second segment, whereas the entailment judge receives only the key fact and candidate memory.

Prompt for the MEMLIFE memory writer

System instruction

You are writing a memory entry for one short segment of an egocentric video. This entry is the only record of this moment that a later question-answering system will see, so capture the concrete, factual details it might be asked about.
Write in the FIRST PERSON as the person wearing the camera (the video is shot from their point of view): refer to yourself as 'I'/'my'/'me', not as 'the camera wearer', 'the camera operator', or 'the person'.
The evidence lives in TWO equally important channels, and you must use both:
1. What is SEEN — the people, their actions, the objects they handle (brand names, text on labels, colors), and the place. Refer to any OTHER person by name when they are a named speaker in the transcript (tags like [B] or [Shure]); add stable visual attributes (clothing, hair, glasses) as a complement to the name, and use visual attributes alone only when no name is available.
2. What is SAID or HEARD — who speaks and the substance of what they say; paraphrase salient statements, questions, decisions, plans, names, numbers, times, and places mentioned in the transcript.
Treat the spoken transcript as a primary source of fact, not background noise: if something is only stated aloud, it still belongs in the memory. Write ONE paragraph of plain prose, up to ∼450 tokens; finish every sentence and do not truncate. No JSON, no bullets, no headers. Describe only this segment; do not speculate beyond the evidence.
Write a COMPLETE, SELF-CONTAINED first-person paragraph describing what happens in this segment (actions, objects, places, people, state changes).

User template

`[eight frames sampled from the segment, 704 px]`
What is said in this segment:
`"""{transcript}"""`
Write the memory entry, grounding it in the frames and transcript above.

Figure 6: Complete system instruction and user template for the MEMLIFE memory writer.

System prompt for the default MEMLIFE reader

You are a memory reader. You answer multiple-choice questions about an egocentric video by calling tools to inspect a pre-built memory.
You have two tools:
1. `search_memory(query: str, time_anchor: [t_start, t_end] = [0, <now>], top_k: int = 32):`
retrieve the memory entries most relevant to the query, restricted to clips within `[t_start, t_end]`. Returns the `top_k` most similar clips in time order — each with its description and time range. Omit `time_anchor` to search all memory up to the current time; pass a narrower `[t_start, t_end]` to focus on a period (e.g. a specific day).
2. `fetch_memory( time_anchor: [t_start, t_end]):`
return every memory clip within `[t_start, t_end]` in time order, each with its description.
Each turn, respond with EXACTLY ONE of these two lines:

```
TOOL: {"name": "<tool>", "args": {...}}
ANSWER: <letter>
```

Emit no other text on the response line. End your final round with `'ANSWER: <letter>'` once you have enough evidence.
Examples:

```
Q: "How often do I cook dinner?" ->
TOOL: {"name": "search_memory",
       "args": {"query": "cooking dinner"}}
Q: "What did I buy yesterday?" ->
TOOL: {"name": "search_memory",
       "args": {"query": "buying purchase shopping",
                "time_anchor": [yesterday_start,
                                yesterday_end]}}
Q: "List everything I bought today." ->
TOOL: {"name": "fetch_memory",
       "args": {"time_anchor": [t_start, t_end]}}
```

Figure 7: System prompt for the default MEMLIFE reader.

Additional source-video tool for MEMLIFE-V

3. `fetch_video( time_anchor: [t_start, t_end], num_frames: int = 30):`
sample `num_frames` frames and the transcript from the video within `[t_start, t_end]`.

```
Q: "What was on my desk when I left?" ->
TOOL: {"name": "fetch_video",
       "args": {"time_anchor": [t_start, t_end],
                "num_frames": 30}}
```

Figure 8: Additional prompt block for MEMLIFE-V. The tool count is updated to three, and all other text follows Figure 7.

Prompt template for grounded key-fact extraction

System instruction
You are auditing ONE 30-second segment of a first-person video against a question about the wearer's day and that question's known-correct answer. The segment has been labelled as evidence for the question. State, in one sentence, the KEY FACT that is actually visible or audible in THIS segment and that helps establish the correct answer.

Rules:
1. The key fact must be something you can SEE in this segment's frames or HEAR in its transcript. Never infer it from the question or from the answer.
2. Write it as a standalone declarative sentence about the segment, naming the concrete objects, people, places, actions or spoken words involved. Do NOT mention the question, the options, or the answer, and do not reuse an option's wording.
3. If this segment does NOT actually contain anything supporting the correct answer - the evidence label is wrong, or the segment is unrelated background - say so instead of inventing a link.
4. A segment can be genuine evidence while showing only PART of what the answer needs; in that case state the part it does show.

Output EXACTLY two lines and nothing else:
SUPPORTS: YES
KEY FACT: `<one sentence>`
or
SUPPORTS: NO
KEY FACT: NONE

User template
This segment
`{eight frames sampled from the segment}`

What is said in this segment:
`"""{transcript}"""`

Question a later reader will ask about this day
`"{question}"`
Options:
`{options}`
Correct answer: `{letter}. {correct_answer}`

What key fact does THIS segment contribute toward that correct answer?
Output the two lines.

Figure 9: Prompt template for grounded key-fact extraction.

Prompt template for informativeness evaluation

System instruction
You are a strict entailment judge. You are given a TARGET FACT about one 30-second segment of a first-person video, and a MEMORY that was written for that same segment. Decide whether the memory states the target fact, or states something from which the target fact follows directly. A paraphrase counts. A vaguer statement that omits the specific detail in the target fact does not. Use only these two texts - no outside knowledge and no guessing. Answer with exactly one word: Yes or No.

User template
TARGET FACT
`"{key_fact}"`

MEMORY
`"{candidate_memory}"`

Does the memory state or directly entail the target fact? Answer Yes or No.
Answer:

Figure 10: Prompt template for informativeness evaluation.

Prompt template for faithfulness evaluation

System instruction
You are correcting a written memory of one 30-second segment of a first-person video. You are given the segment's frames and transcript, and ONE memory of that segment. A claim is unsupported if it asserts an object, person, place, action, quantity, or spoken content that is not actually visible in the segment's frames or audible in its transcript. A memory that is vague, brief or incomplete is still supported.

If every claim in the memory is supported, repeat the memory back EXACTLY as given, character for character. If some claim is not supported, output the memory with those claims removed, or weakened to what the segment actually shows, changing as few words as possible. Do not add detail and do not improve the writing.

Output only the memory text. No preamble, no explanation, no quotation marks.

User template
The segment
`{eight frames sampled from the segment}`

What is said in this segment:
`"""{transcript}"""`

Memory of this segment
`"{candidate_memory}"`

Output this memory with any unsupported claims corrected, or unchanged if every claim is supported.
Memory:

Figure 11: Prompt template for faithfulness evaluation.