EvoMM: Reinforced Self-Evolving Multimodal Agentic Memory

Tong Zhao[1], Chenghao Zhang[1], Yucheng Tian[2]
Yuyang Hu[1], Yutao Zhu[1], Zhicheng Dou[1]*
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]South China University of Technology
zhaotong7@ruc.edu.cn, dou@ruc.edu.cn

Abstract

Multimodal large language model (MLLM) agents increasingly operate over long histories of dialogues, images, and evolving facts, yet finite context windows and weak persistent memory cause forgetting, temporal inconsistency, and loss of precise past evidence. Existing multimodal memory agents either discard visual evidence by reducing images to captions or treat memory as a static store that is built once and never revised during reasoning; self-evolving agents further accumulate experience via prompting without tight coupling to retrieval and memory revision. Closing this loop with reinforcement learning additionally suffers from credit assignment difficulties over long, multimodal action trajectories. To address these challenges, we propose `EvoMM`, a reinforced self-evolving multimodal agentic memory framework that unifies memory construction, retrieval, and evolution within a single agentic loop. `EvoMM` maintains a mutable two-level memory of turn-level short memories and session-level long summaries, organized by a heterogeneous multimodal memory graph with temporal, semantic, image-co-reference, and keyword edges. An iterative `PLAN`→`RETRIEVE`→`REFLECT`→`ANSWER` policy then revises long memory on the fly and distills structured retrieval experience into a cross-query store for reuse. To optimize this policy, we further introduce `PRR-GRPO` (Planning-Retrieval-Reflection GRPO), which augments outcome rewards with gated, action-aware process rewards and performs dense credit assignment over a memory-action tree with dual-scale advantage normalization. Experiments on Mem-Gallery, MMLongBench, and LoCoMo show that `EvoMM` consistently outperforms strong text-only, multimodal, and agentic memory baselines, establishing a training paradigm for memory that reads, revises, and evolves with each query.

1 Introduction

Multimodal large language models (MLLMs) are increasingly deployed as agents that reason over long-horizon user histories of dialogues, images, and temporally evolving facts [44, 10]. Yet finite context windows and weak persistent memory cause forgetting, temporal inconsistency, and failure to recover precise past evidence [19, 43, 50]. These challenges are acute in multimodal settings, where evidence spans modalities and abstraction levels, motivating a principled treatment of persistent and mutable memory.

To mitigate these limitations, a growing body of work has explored memory-augmented agents that explicitly store and retrieve historical information [5, 4, 22, 23, 15]. However, existing multimodal memory systems exhibit a clear progression of unresolved limitations. Caption-based approaches [7,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

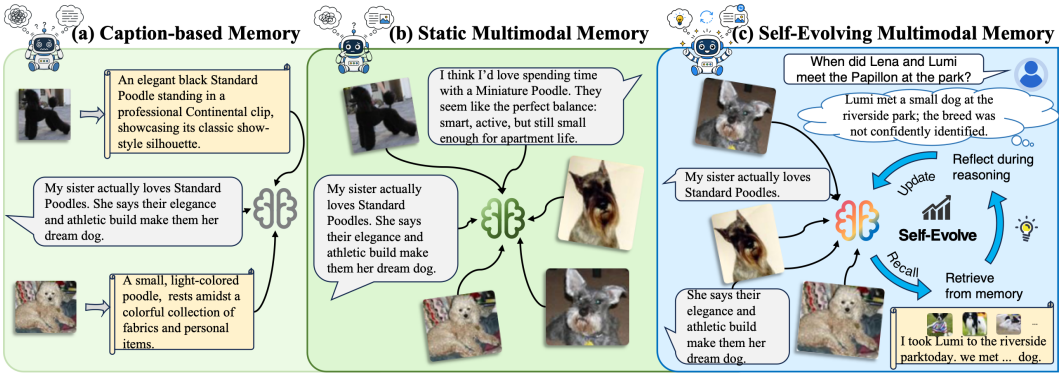

Figure 1: Motivation of EvoMM. **(a)** Caption-based memory discards visual evidence; **(b)** static multimodal memory keeps images but is built once and never revised; **(c)** EvoMM closes the retrieve–reflect–update loop, enabling temporally grounded multimodal reasoning over evolving user histories.

23] convert images into textual descriptions at construction time, irreversibly discarding fine-grained visual details such as spatial layout, color, and object identity (Figure 1a). Recent multimodal systems preserve raw images or visual representations alongside text [22, 24, 53], improving access to visual evidence. However, most still emphasize retrieval-time selection or evidence reweighting rather than fine-grained memory revision during reasoning; when revision is supported, it is often limited to coarse-grained overwrites of session summaries [31, 59, 18]. Meanwhile, self-evolving agents accumulate experience via prompting [37, 41, 57, 26], but treat it as an auxiliary side channel decoupled from memory retrieval and revision [1, 55, 35]. This motivates a unified framework in which memory construction, retrieval, and self-evolution are jointly optimized rather than addressed in isolation.

A natural way to realize such joint optimization is reinforcement learning (RL), which has recently been used to train agents as adaptive policies rather than fixed prompting pipelines [34, 12, 51, 33, 36]. Nevertheless, applying RL to multimodal memory-augmented agents remains challenging: multimodal queries typically induce longer action trajectories than text-only queries [39, 45, 29], making a single final reward markedly sparser and exacerbating the credit assignment problem, so existing training paradigms provide limited supervision for which intermediate memory operations actually contribute to successful long-horizon reasoning. Together, these observations raise a central research question: *How can we build a multimodal memory-augmented agent that unifies memory construction, retrieval, and evolution under dense process supervision?*

We propose EvoMM (Figure 1c), a reinforced self-evolving multimodal agentic memory framework that closes the loop among retrieval, reflection, and memory revision. EvoMM maintains a mutable two-level memory of raw and long memories, organized by a heterogeneous multimodal memory graph linking memory pages via temporal, semantic, image-co-reference, and keyword relations. The agent runs an iterative PLAN→RETRIEVE→REFLECT→ANSWER loop that retrieves evidence, revises long memory on the fly, and answers from an increasingly refined state. Across queries, EvoMM further distills retrieval experience into an experience store for reuse without parameter updates. To optimize this end-to-end, we introduce PRR-GRPO (Planning-Retrieval-Reflection GRPO), an RL framework that augments outcome rewards with action-aware process rewards and performs dense credit assignment over a memory-action tree.

Our contributions can be summarized as follows:

- We expose how decoupled memory construction, retrieval, and revision limit multimodal memory-augmented agents, motivating their unification under dense process supervision.
- We propose EvoMM, a self-evolving multimodal memory-augmented agent unifying a memory graph, on-the-fly revision, and a cross-query experience store in one agentic loop, with PRR-GRPO, a process-supervised RL framework with dense credit assignment over long-horizon trajectories.
- On Mem-Gallery, MMLongBench, and LoCoMo, EvoMM consistently outperforms text-only, multimodal, and agentic memory baselines, yielding a paradigm for memory that reads, revises, and evolves per query.

2 Related Work

Memory-Augmented Multimodal Agents Research has shifted from text-only retrieval to multimodal memory-augmented agents that store and reason over visual–textual experiences across long-horizon interactions [5, 4, 23]. Early frameworks target external image–text retrieval or knowledge-grounded multimodal QA, but typically treat memory as a static store and retrieval as a one-shot procedure [5, 7, 22, 50, 40]. More agentic settings require search and memory update over time when user history contains heterogeneous evidence [31, 18, 53], and recent multimodal agents move in this direction by combining long-term memory with iterative reasoning over multimodal observations [23, 24, 20]. However, on-the-fly memory revision during reasoning and reusable retrieval experience across queries are rarely supported. Our `EvoMM` bridges this gap by unifying memory construction, retrieval, and revision within a single agentic loop.

Self-Evolving LLM Agents Self-improving language agents refine their behavior through interaction and accumulated experience, going beyond classical continual learning under predefined task boundaries [17, 30, 26, 37]. One line improves agents within a single episode by prompting them to critique and revise intermediate outputs [26, 11]; another improves them across episodes by storing reflections, lessons, or skills for reuse [37, 41, 57]. Most rely on prompt-based updates over loosely structured memories, without systematically refining retrieval strategies or optimizing how experience is written back. In parallel, RL methods such as GRPO and process supervision improve reasoning via outcome and step-level signals [34, 12, 21, 42], but are mostly developed for single-turn reasoning and do not address long-horizon credit assignment in multimodal agentic settings where planning, retrieval, reflection, and memory update are tightly coupled [51, 33]. In contrast, `EvoMM` treats self-evolution as a trainable agentic process: it accumulates structured retrieval experience, refines memory through reflection, and optimizes the full policy end-to-end with `PRR-GRPO`.

3 Problem Formulation

We focus on multimodal agentic memory reasoning over long-horizon user histories. Given a multimodal dialogue history $\mathcal{H} = \{(u_t, a_t, \mathcal{I}_t)\}_{t=1}^{T}$, where u_t, a_t, and $\mathcal{I}_t$ denote the user utterance, assistant response, and images at turn t, and a query $q = (q^{\text{text}}, q^{\text{img}})$, the goal is to produce an answer $\hat{y}$ grounded in evidence drawn from $\mathcal{H}$. The system maintains an external memory $\mathcal{M} = (\mathcal{M}^{\text{raw}}, \mathcal{M}^{\text{long}}, \mathcal{G})$ of turn-level raw records, session-level long summaries, and a heterogeneous multimodal memory graph $\mathcal{G}$ linking raw pages by temporal, semantic, and visual relations. An experience store $\mathcal{X}$ accumulates retrieval strategies distilled from previous queries, and a graph-aware retriever R returns evidence $E_k = R(\mathcal{M}, \mathcal{G}, q, r_k)$ given an internal query r_k. Unlike standard one-shot retrieval, this formulation accommodates *mutable* memory revisable during reasoning and reusable retrieval skills that evolve across queries.

We model reasoning as a policy π_θ that iteratively selects actions $z_k \in \{\texttt{PLAN}, \texttt{RETRIEVE}, \texttt{REFLECT}, \texttt{ANSWER}\}$ together with their free-form contents c_k, and produces a trajectory $\tau = (z_1, c_1, \ldots, z_K, c_K, \hat{y})$ conditioned on q, $\mathcal{M}$, and $\mathcal{X}$, maximizing expected trajectory reward:

$$\theta^\star = \arg\max_\theta \; \mathbb{E}_{\tau\sim\pi_\theta}\big[r(\tau)\big], \tag{1}$$

where $r(\tau)$ is a composite reward that jointly evaluates the final answer and intermediate agentic actions, framing memory construction, retrieval, and evolution as integrated components of a unified trainable policy.

4 Method

We introduce `EvoMM`, a reinforced self-evolving multimodal agentic memory framework. As discussed in Section 1, effective long-horizon multimodal reasoning requires not only retrieving past evidence, but also re-checking and refining long memory against the retrieved historical evidence, while accumulating reusable retrieval strategies across queries. To address these requirements, `EvoMM` unifies planning, retrieval, and reflection within a single agentic loop (Section 4.1) and trains this loop end-to-end with `PRR-GRPO` (Section 4.2), a trajectory-supervised reinforcement learning framework that provides dense credit assignment over a memory-action tree.

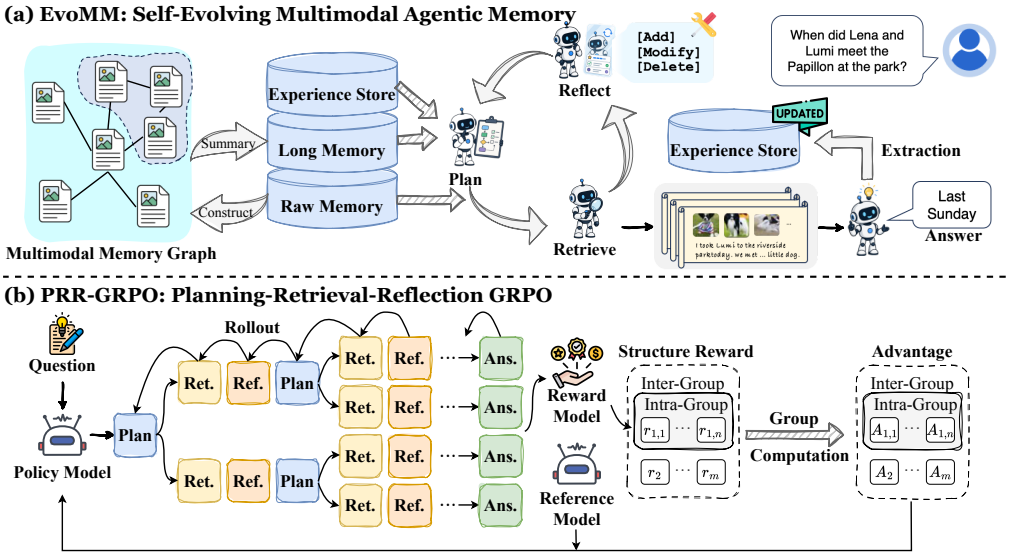

Figure 2: Overview of EvoMM. **(a)** Over a graph-structured two-level memory and a cross-query experience store, the agent iteratively PLANs, RETRIEVEs, REFLECTs (ADD/MODIFY/DELETE), and ANSWERs; **(b)** PRR-GRPO optimizes this loop with structured rewards and dual-scale (intra/inter-group) advantage normalization.

4.1 EvoMM

Memory Construction. EvoMM constructs a two-level memory bank $\mathcal{M} = (\mathcal{M}^{\text{raw}}, \mathcal{M}^{\text{long}})$ from $\mathcal{H}$, balancing fine-grained evidence with compact session summaries. The raw memory $\mathcal{M}^{\text{raw}} = \{m_i\}_{i=1}^{N}$ stores one page per turn, $m_i = (d_i, I_i, K_i, \sigma_i, p_i)$, where d_i, I_i, K_i, σ_i, and p_i are the user–assistant text, image set, entity list, extractive summary, and temporal position (session id and turn index). The long memory $\mathcal{M}^{\text{long}} = \{l_s\}_{s=1}^{S}$ contains one abstractive summary per session, $l_s = \pi_\theta(\texttt{summarize}; \{m_i\}_{i\in\mathcal{S}_s})$, covering key facts, dates, conflicts, and image-grounded information. Raw memories thus supply fine-grained evidence for retrieval, while long memories give π_θ a compact overview for planning and act as a mutable knowledge base revisable during reasoning.

Multimodal Memory Graph. A flat retriever overlooks the cross-turn and cross-modal structure essential for long-horizon reasoning. We therefore organize $\mathcal{M}^{\text{raw}}$ as a heterogeneous graph

$$\mathcal{G} = \big(\mathcal{V},\ \mathcal{E}_{\text{tmp}} \cup \mathcal{E}_{\text{sem}} \cup \mathcal{E}_{\text{img}} \cup \mathcal{E}_{\text{kw}}\big), \tag{2}$$

where each node $v_i \in \mathcal{V}$ corresponds to a raw page m_i and inherits its text, entities K_i, images I_i, and position p_i. Edges capture complementary relations: $\mathcal{E}_{\text{tmp}}$ connects consecutive turns within a session to preserve discourse order; $\mathcal{E}_{\text{sem}}$ links each page to its top-K_g neighbors above cosine threshold θ_{sem} on dense embeddings; $\mathcal{E}_{\text{img}}$ connects pages sharing an image identifier; and $\mathcal{E}_{\text{kw}}$ is a stopword-filtered inverted index from content lemmas to page sets. The graph is built once per dialogue history and cached, with lookup tables by session, turn, and image; during reasoning it is queried but never modified.

Self-Evolving Agentic Memory Retrieval. EvoMM performs retrieval as an iterative agentic process rather than a one-shot lookup. Given a query q, the policy π_θ generates a trajectory $\tau = (z_1, c_1, \ldots, z_K, c_K)$ with action tags $z_k \in \{\texttt{PLAN}, \texttt{RETRIEVE}, \texttt{REFLECT}, \texttt{ANSWER}\}$ and free-form content c_k, sampled as $(z_k, c_k) \sim \pi_\theta(\cdot \mid \mathbf{h}_{k-1})$ with $\mathbf{h}_0$ initialized from q, $\mathcal{M}^{\text{long}}$, and the retrieved experiences $\mathcal{X}_q$ (defined below). A [PLAN] step inspects $\mathcal{M}^{\text{long}}$ and q to verbalize what information is needed and which sessions are relevant; the PLANNING prompt (Appendix E) structures this into a plan over three retrieval tools (keyword, vector, page_index). A [RETRIEVE] step turns the plan into retrieval queries r_k and invokes graph-augmented hybrid retrieval: a base retriever yields a seed set S_k from keyword, entity, and dense multimodal similarity, which the memory graph expands into

$$E_k \ = \ S_k \ \cup \ \text{Top}_M\big(\text{Expand}_{\mathcal{G}}(S_k,\, q;\, \boldsymbol{\alpha})\big), \tag{3}$$

where $\mathrm{Expand}_{\mathcal{G}}$ scores each neighbor as a weighted sum of temporal, semantic, image-co-reference, and keyword edges with a single set of fixed weights $\boldsymbol{\alpha}$ shared across all queries; merged hits are summarized into a working result by the INTEGRATE prompt and the state is updated as $\mathbf{h}_k = [\mathbf{h}_{k-1};\ \texttt{[RETRIEVE]}\ r_k;\ \texttt{[EVIDENCE]}\ \mathrm{format}(E_k)]$. A [REFLECT] step reconciles evidence with the current draft via three sub-prompts that share π_θ: INFOCHECK judges whether the working result covers every sub-question in q, GENERATEREQUESTS formulates the next round of focused queries, and SELFCORRECTION produces a minimally edited draft of $\mathcal{M}^{\text{long}}$. The diff between the original draft and SELFCORRECTION's `corrected_answer` is converted into discrete operations $\{(\mathrm{op}_j, \delta_j)\}$ with $\mathrm{op}_j \in \{\text{ADD}, \text{MODIFY}, \text{DELETE}\}$ and applied in place,

$$\mathcal{M}^{\text{long}} \leftarrow \mathrm{Apply}\big(\mathcal{M}^{\text{long}},\ \{(\mathrm{op}_j, \delta_j)\}_j\big), \tag{4}$$

correcting outdated or contradictory summaries on the fly so subsequent [PLAN]/[RETRIEVE] steps operate on an increasingly accurate knowledge state. The loop iterates up to $K_{\max}$ times before an [ANSWER] step emits $\hat{y} = \pi_\theta(\texttt{ANSWER} \mid \mathbf{h}_{K-1})$.

Cross-Query Skill Evolution. Beyond the inner loop, `EvoMM` improves retrieval across queries through an experience store $\mathcal{X}$. After each trajectory, π_θ is invoked once more with the experience-extraction prompt (Appendix E.8) to distill an entry $x = (q,\ \{r_k^\star\},\ \kappa,\ \xi,\ \mathcal{L}_x)$, where $\{r_k^\star\}$ are the effective search queries, κ a key insight, ξ a brief self-assessment, and $\mathcal{L}_x$ a set of transferable lessons. A quality gate $\mathrm{quality}(x) \geq \eta$ admits only high-quality entries: $\mathcal{X} \leftarrow \mathcal{X} \cup \{x\}$. During training the gate scores effective queries, insight, lessons, and the answer F1 jointly; at evaluation we use a reference-free proxy over the JSON content (Appendix E), so test-time entries are never gated by ground-truth labels. For a subsequent query q', the top-P entries $\mathcal{X}_{q'} \subset \mathcal{X}$ retrieved by stopword-filtered keyword overlap with the question are prepended to the initial context, $\mathbf{h}_0 = [\mathrm{system};\ \mathrm{format}(\mathcal{X}_{q'});\ \mathcal{M}^{\text{long}};\ q']$, allowing π_θ to reuse strategies without parameter updates.

4.2 PRR-GRPO: Planning-Retrieval-Reflection GRPO

Because the agentic loop couples planning, retrieval, reflection, and memory update, outcome-only optimization is ineffective for long trajectories. We therefore train π_θ with GRPO augmented with action-aware process rewards and dense credit assignment, termed `PRR-GRPO`.

Rollout. For each prompt we sample G trajectories $\{\tau^{(g)}\}_{g=1}^{G}$ through the same loop. At [PLAN] nodes (depth $\leq D_{\max}$), up to B alternative branches are explored in parallel and the highest-scoring branch is retained, yielding a memory-action tree $\mathcal{T}$ that exposes intermediate decision points to credit assignment.

Structured Reward. We decompose the trajectory reward into an outcome reward on the final answer $\hat{y}$ versus the reference $y^\star$ and a process reward over intermediate actions, so the policy is not supervised solely by sparse end-of-trajectory signals. The outcome reward combines stopword-filtered token F1, exact match (EM), and a soft containment score SoftEM that gives partial credit when the canonicalized form of one string is contained in the other,

$$r_{\text{ans}} = w_1\,\mathrm{F1}(\hat{y},\ y^\star) + w_2\,\mathrm{EM}(\hat{y},\ y^\star) + w_3\,\mathrm{SoftEM}(\hat{y},\ y^\star), \tag{5}$$

and a conciseness term defined over the length ratio $\rho = |\hat{y}|/|y^\star|$, which applies a piecewise-linear decay once ρ exceeds a length-adaptive soft cap ρ_{soft} (set as a function of $|y^\star|$ alone, see Appendix B) and adds a tail penalty $r_{\text{tail}} \leq 0$ for excessive tokens after [ANSWER],

$$r_{\text{conc}} = f_{\text{conc}}(\rho;\ \rho_{\text{soft}}) + r_{\text{tail}}, \quad r_{\text{tail}} = -\min\big(\kappa_{\text{tail}},\ \alpha_{\text{tail}} \cdot \max(|\hat{y}| - \lceil \rho_{\text{soft}} \cdot |y^\star| \rceil,\ 0)\big). \tag{6}$$

Beyond final-answer supervision, we score the reasoning actions $z_k \in \{\texttt{PLAN}, \texttt{RETRIEVE}, \texttt{REFLECT}\}$ along five axes: **(1)** evidence relevance $r_{\text{evd}} = |\mathrm{tok}(y^\star) \cap \mathrm{tok}(\bigcup_k E_k)|/|\mathrm{tok}(y^\star)|$, the fraction of reference-answer tokens covered by retrieved pages; **(2)** plan specificity r_{pln}, averaging detail-and-actionability scores over [PLAN] nodes; **(3)** reflection validity r_{ref}, the proportion of well-formed ADD/MODIFY/DELETE updates with non-trivial justifications; **(4)** format compliance r_{fmt}, the fraction of valid action tags with extra credit for opening at [PLAN] and ending at [ANSWER]; and **(5)** experience quality r_{exp} over the extracted experience entry. All process rewards are normalized to $[0, 1]$ and gated by $r_c \leftarrow r_c \cdot \mathbb{1}[r_{\text{ans}} \geq \eta_{\text{gate}}]$, suppressing process credit for trajectories with

insufficiently accurate final answers and avoiding reward leakage to well-structured but incorrect traces. The trajectory reward is then

$$r(\tau) = \lambda_{\text{ans}}\, r_{\text{ans}} + \lambda_{\text{conc}}\, r_{\text{conc}} + \sum_{c\in\mathcal{P}} \lambda_c\, r_c, \quad \mathcal{P} = \{\text{evd, pln, ref, fmt, exp}\}, \tag{7}$$

where λ_{ans} and λ_{conc} receive the largest weights to prioritize answer quality and $\{\lambda_c\}$ provide auxiliary shaping signals.

Dense Credit Assignment. GRPO normalizes $r(\tau)$ across the G rollouts to obtain a per-trajectory advantage A_j that is broadcast to every response token, but this single scalar cannot distinguish which `[PLAN]/[RETRIEVE]/[REFLECT]` segments drove success or failure. We therefore add a *node-level* outcome credit that modulates A_j token-wise. Segmenting the response by action boundaries into nodes of $\mathcal{T}$, we propagate the answer-side outcome score bottom-up:

$$r_n = \begin{cases} r_{\text{ans}}, & \text{if } n \text{ is a leaf (terminal } \texttt{[ANSWER]} \text{ node)}, \\ \gamma \cdot \dfrac{1}{|\text{ch}(n)|} \sum_{n'\in\text{ch}(n)} r_{n'}, & \text{otherwise}, \end{cases} \tag{8}$$

with depth discount $\gamma \in (0, 1]$, so $r_n \in [0, 1]$ measures how strongly the subtree rooted at n contributed to a correct answer; gated process rewards in Eq. (7) intentionally do *not* enter r_n, since their role is to shape A_j rather than label individual actions. We then modulate each token's advantage by a non-negative confidence gate

$$\tilde{A}_j = A_j \cdot w_{\text{node}(j)}, \qquad w_n = 0.5 + 0.5 \cdot r_n \in [0.5,\, 1.0]. \tag{9}$$

Because $w_n \in [0.5, 1.0]$, the gate (i) contracts $|\tilde{A}_j| \le |A_j|$ on every token, preserving the sign of the update; (ii) for high-r_{ans} rollouts, leaves keep weight ≈ 1 while shallow ancestors are damped by γ^d, focusing gradient on answer-producing tokens; (iii) for low-quality rollouts, r_n collapses so $w_n \to 0.5$ uniformly, halving noisy gradient. Thus $\tilde{A}_j$ provides dense, outcome-anchored credit without changing the learning direction.

Dual-Scale Advantage and Loss. Within-group normalization alone is unstable when prompts in a batch have varying difficulty. We normalize at two scales: *intra-group* $A^{(g)}_{\text{intra}} = (r^{(g)} - \bar{r})/(\sigma_r + \epsilon)$ within each prompt's G rollouts, and *inter-group* $A^{(p)}_{\text{inter}} = (\bar{r}^{(p)} - \bar{\bar{r}})/(\sigma_{\bar{r}} + \epsilon)$ across the batch $\mathcal{B}$. The final advantage is $A = A_{\text{intra}} + A_{\text{inter}}$, clamped to $[-10, 10]$. The training loss is a clipped policy gradient with entropy bonus and KL regularization,

$$\mathcal{L} = \mathbb{E}_j\big[-\min\big(\rho_j \tilde{A}_j,\ \text{clip}(\rho_j, 1-\epsilon, 1+\epsilon')\tilde{A}_j\big)\big] - \beta_{\text{ent}} H(\pi_\theta) + \beta_{\text{kl}}\, \text{KL}(\pi_\theta \| \pi_{\text{ref}}), \tag{10}$$

where $\rho_j = \pi_\theta(a_j)/\pi_{\theta_{\text{old}}}(a_j)$ is the per-token importance ratio.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate `EvoMM` on three long-horizon benchmarks: Mem-Gallery [3] for multimodal conversational memory, MMLongBench [25] for long-context multimodal document QA, and LoCoMo [27] for long-horizon multi-session dialogue with strong temporal and multi-hop requirements. Following prior work [48, 54], we split Mem-Gallery into train/val/test with an 8:1:1 ratio; for MMLongBench and LoCoMo we follow the standard evaluation protocols in their original papers.

Baselines. We compare `EvoMM` against four families of baselines: *raw-context* prompting, *text-only memory*, *multimodal memory*, and *agentic memory*. On Mem-Gallery, the text-only group consists of Full Memory (Text), FIFO, NaiveRAG, Reflexion [37], HippoRAG [13], MemGPT [28], and MemoryOS [16]; the multimodal group consists of Full Memory (MM), UniversalRAG [52], NGM [9], AUGUSTUS [14], and MuRAG [5]; and the agentic group consists of Generative Agent [31], GAM [47], A-Mem [46], M3-Agent [23], and WorldMM [53]. On MMLongBench, we follow the official long-context document-QA protocol [25] and report Direct (raw context with all K pages), a multimodal RAG baseline (MuRAG [5]) and an agentic-memory baseline (WorldMM [53]),

Table 1: Main results on Mem-Gallery with `Qwen3-VL-4B-Instruct` as the shared backbone. All metrics are reported as percentages (F1: token F1; EM: exact match; LLM-J: LLM-judge score). Per-category labels follow Mem-Gallery: AR (answer refusal), CD (contradiction detection), FR (factual recall), KR (knowledge retrieval), MR (multi-hop reasoning), TR (temporal reasoning), VR (visual recall), VS (visual search), TTL (long-range titling). Best per column is in **bold**.

Method	Per-category F1									Overall		
	AR	CD	FR	KR	MR	TR	VR	VS	TTL	F1	EM	LLM-J
Text-only memory												
Full Memory (Text)	95.45	16.67	29.70	8.91	32.81	14.34	17.22	12.29	25.09	29.88	20.00	39.41
FIFO	**100.00**	16.67	21.74	10.27	31.41	8.33	0.48	3.23	8.55	21.97	14.71	24.71
NaiveRAG	96.11	16.67	55.42	29.17	40.45	42.85	33.84	61.29	76.15	59.86	42.94	73.82
Reflexion [37]	95.48	16.67	28.88	8.58	33.06	13.89	17.22	14.44	24.07	29.89	20.00	38.24
HippoRAG [13]	95.31	50.00	47.03	**41.06**	45.77	55.56	58.91	74.57	84.49	65.28	48.82	78.27
MemGPT [28]	90.81	16.67	44.41	40.73	35.17	42.82	25.00	56.24	39.77	47.27	28.24	68.53
MemoryOS [16]	96.29	33.33	54.85	9.63	29.36	55.56	16.67	71.23	73.11	56.18	43.74	71.76
Multimodal memory												
Full Memory (MM)	95.62	16.67	28.38	11.20	32.44	13.89	17.22	15.51	38.36	33.74	25.29	39.71
UniversalRAG [52]	95.53	33.33	56.97	37.05	52.69	43.25	52.56	79.57	78.76	67.54	47.06	80.35
NGM [9]	96.11	16.67	47.80	32.98	40.13	42.98	33.89	79.57	85.15	64.75	49.41	77.65
AUGUSTUS [14]	85.79	16.67	56.74	37.02	47.09	43.26	17.22	79.25	83.37	63.83	46.47	80.76
MuRAG [5]	95.42	33.33	58.69	33.78	48.78	43.33	53.57	81.40	84.56	69.05	51.18	80.59
Agentic memory												
Gen. Agent [31]	95.91	16.67	21.71	21.62	32.70	20.83	8.81	26.77	58.74	40.67	20.82	50.59
GAM [47]	95.34	16.67	54.46	27.91	38.05	55.56	25.79	59.03	76.38	59.32	41.76	66.18
A-Mem [46]	92.17	33.33	48.67	11.37	48.94	58.33	35.67	67.31	72.42	58.45	45.64	72.27
M3-Agent [23]	91.55	33.33	46.06	16.94	49.83	43.93	33.33	67.37	66.90	57.02	42.35	69.35
WorldMM [53]	90.97	33.33	58.54	17.00	**53.80**	36.11	34.62	69.68	76.74	62.16	43.53	69.12
`EvoMM` *w/o RL*	86.67	50.00	59.39	34.84	46.10	63.89	67.19	**83.76**	87.60	71.67	56.65	82.82
`EvoMM` (`PRR-GRPO`)	95.43	**66.67**	**60.36**	34.90	44.03	**63.89**	**75.51**	81.83	**90.70**	**74.05**	**60.00**	**85.59**

under the same backbone, with $K \in \{8, 16, 32, 64, 128\}$ over the LongDocURL, MMLongDoc, and SlideVQA subsets. On LoCoMo, we compare with Long-LLM, RAG, A-Mem [46], Mem0 [6], MemoryOS [16], HippoRAG [13], LightMem [8], and GAM [47]. Full baseline configurations are provided in Appendix C.

Implementation Details. We use Qwen3-VL-4B-Instruct [2] as the policy, reference, and training-time judge, GME-Qwen2-VL-2B [56] as the dense retriever, and all-MiniLM-L6-v2[2] with a keyword–entity index for graph construction and lexical retrieval. To avoid evaluator-policy alignment, Mem-Gallery LLM-J is scored by an independent frozen Qwen3-32B [49]; MMLongBench and LoCoMo use rule-based metrics. The agent runs at most three `PLAN`→`RETRIEVE`→`REFLECT` iterations and keeps up to 10 evidence pages. `PRR-GRPO` is trained for 200 steps with 8 rollouts per prompt, learning rate 5×10^{-7}, F1-gated process rewards, and KL regularization on $8\times$NVIDIA A800 GPUs using Slime+Megatron-LM and SGLang rollouts [60, 38, 58]. On LoCoMo, we use an off-the-shelf Qwen2.5-14B-Instruct backbone without `PRR-GRPO`, isolating the gain from the agentic loop. Full hyperparameters are in Appendix B.

5.2 Main Results

Comparison with text-only memory baselines. On Mem-Gallery (Table 1), `EvoMM` achieves the strongest overall performance (74.05 F1, 60.00 EM, 85.59 LLM-J), exceeding the best text-only system HippoRAG by 8.77 F1 and 11.18 EM, and surpassing NaiveRAG (59.86 F1) and MemoryOS (56.18 F1). Gains are most pronounced where multimodal grounding is needed - visual recall (VR: 75.51 vs. 58.91), visual search (VS: 81.83 vs. 74.57), and long-range titling (TTL: 90.70 vs. 84.49) -

[2] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

Table 2: Results on MMLongBench with `Qwen3-VL-4B-Instruct` as the shared backbone. We report document-QA accuracy (%) averaged over the three subsets MMLongDoc, LongDocURL, and SlideVQA, sweeping the per-query page budget $K \in \{8, 16, 32, 64, 128\}$. The *Avg.* column averages over K. Best per column is in **bold**.

Method	K=8	K=16	K=32	K=64	K=128	Avg.
Backbone: Qwen3-VL-4B-Instruct (off-the-shelf)						
Direct (raw context)	66.53	63.39	59.74	57.02	53.99	60.13
MuRAG [5]	64.98	64.53	61.18	59.13	58.86	61.73
WorldMM [53]	66.37	63.72	61.57	60.18	58.95	62.15
`EvoMM` *w/o RL*	66.65	64.29	62.40	60.15	59.11	62.52
Backbone: Qwen3-VL-4B-Instruct fine-tuned with `PRR-GRPO`						
Direct (raw context)	66.35	63.40	59.93	57.29	54.25	60.24
`EvoMM` (`PRR-GRPO`)	**66.89**	**64.87**	**63.17**	**60.95**	**60.28**	**63.23**

Table 3: Results on LoCoMo with an off-the-shelf `Qwen2.5-14B-Instruct` as the shared backbone. Best per column is in **bold**.

Method	Single Hop		Multi Hop		Temporal		Open Domain	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Long-LLM	46.05	39.56	32.08	24.46	30.51	24.45	14.89	11.41
RAG	47.87	42.79	26.38	19.54	30.78	25.97	14.16	10.52
A-Mem	33.75	30.04	22.09	15.28	27.19	22.05	13.49	10.74
Mem0	42.58	35.15	31.73	24.82	28.96	26.24	15.03	11.28
MemoryOS	46.33	41.62	38.19	29.26	32.24	27.86	20.27	15.94
HippoRAG	42.45	37.14	27.57	20.62	30.66	26.83	19.74	15.81
LightMem	34.92	31.22	25.45	19.61	32.03	27.70	15.81	11.81
GAM	55.63	50.32	41.49	34.03	42.89	37.78	30.97	26.76
`EvoMM`	**59.14**	**52.6**	**45.69**	**38.19**	**50.06**	**44.11**	**31.87**	**27.86**

indicating that flat textual indexing, even with PageRank or hierarchical stores, struggles to recover multimodal evidence and maintain temporal consistency.

Comparison with multimodal memory baselines. `EvoMM` also outperforms multimodal baselines that index both text and images: $+5.00$ F1 and $+8.82$ EM over the strongest anchor MuRAG (69.05/51.18), and ahead of UniversalRAG/AUGUSTUS (67.54/63.83 F1). The gains are most visible on contradiction detection (CD: 66.67 vs. at most 33.33), temporal reasoning (TR: 63.89 vs. 43.33), and TTL (90.70 vs. 85.15), which all require reconciling conflicting facts across sessions and tracking evolving evidence - supporting the design choice that multimodal memory should be mutable and queried iteratively rather than treated as a static index.

Comparison with agentic memory baselines. Among agentic systems, the strongest competitors WorldMM and A-Mem still trail `EvoMM` (62.16 / 58.45 vs. 74.05 F1). Even without RL, `EvoMM` *w/o RL* reaches 71.67 F1 / 56.65 EM, exceeding all agentic baselines and confirming that the unified `PLAN`→`RETRIEVE`→`REFLECT`→`ANSWER` loop with the memory graph and experience store is itself a strong inductive prior. Adding `PRR-GRPO` brings further $+2.38$ F1, $+3.35$ EM, and lifts LLM-J from 82.82 to 85.59, with the largest per-category gains on CD ($+16.67$), VR ($+8.32$), and TTL ($+3.10$), showing that dense action-aware process supervision sharpens decisions when correctness hinges on a few pivotal retrieval and reflection steps that final-answer rewards alone cannot credit.

Performance on long-context multimodal QA. Table 2 evaluates `EvoMM` on MMLongBench under the official long-context document-QA protocol. `EvoMM` (`PRR-GRPO`) leads every column, raising average accuracy from 60.13 (*Direct*, all K pages) / 61.73 (MuRAG) / 62.15 (WorldMM) to 63.23,

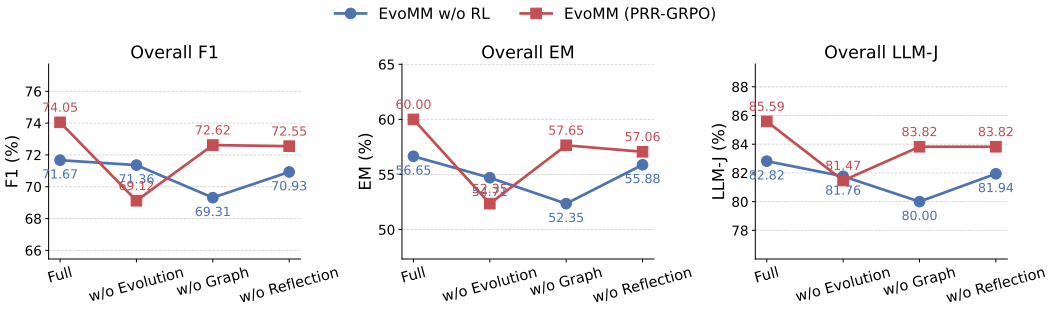

Figure 3: Single-factor ablations on Mem-Gallery. *Full* denotes the complete EvoMM system and reproduces the corresponding results in Table 1.

Table 4: Comparison of training variants on Mem-Gallery. Metrics are percentages, reported as mean ± standard deviation over three seeds. Best per column is in **bold**.

Training	F1	EM	LLM-J
EvoMM *w/o RL*	71.6 ± 0.3	56.5 ± 0.4	82.7 ± 0.3
Vanilla GRPO	72.2 ± 0.5	57.1 ± 0.5	83.1 ± 0.4
PRR-GRPO	$\mathbf{73.8 \pm 0.4}$	$\mathbf{59.6 \pm 0.5}$	$\mathbf{85.2 \pm 0.4}$

with the gap over *Direct* widening from $+0.36$ at $K{=}8$ to $+6.29$ at $K{=}128$ as in-context interference among long interleaved text–image sequences grows. The $+1.50$ / $+1.08$ leads over MuRAG / WorldMM at the same top-10 retrieval budget show that the gain comes from graph-augmented retrieval and the iterative PLAN→RETRIEVE→REFLECT→ANSWER loop, not from noise filtering.

Performance on long-horizon dialogue memory. On LoCoMo, EvoMM achieves the best F1 and BLEU-1 across *all* four question types, surpassing the strongest agentic baseline GAM by 3.51, 4.20, 7.17, and 0.90 F1 on Single-Hop, Multi-Hop, Temporal, and Open-Domain. The largest gain is on Temporal (50.06 vs. 42.89), aligning with the temporal edges and the reflection step that revises stale long-memory summaries. The consistency across question types indicates that benefits are not tied to a particular format but reflect the joint optimization of retrieval, reflection, and memory update.

5.3 Ablation Studies

We conduct single-factor ablations on Mem-Gallery to quantify the contribution of three core components: the multimodal memory *graph* (*w/o Graph* disables graph-based expansion), on-the-fly *reflection* (*w/o Reflection* removes self-correction and updates to $\mathcal{M}^{\text{long}}$), and cross-query *evolution* (*w/o Evolution* disables the experience store $\mathcal{X}$). All variants are evaluated with both the off-the-shelf backbone (*EvoMM w/o RL*) and the PRR-GRPO-trained policy, as shown in Figure 3.

The complete model consistently outperforms all ablated variants, indicating that the three components are complementary. Cross-query *evolution* yields the largest gain after RL: removing $\mathcal{X}$ drops the trained policy by 4.93 F1 and 7.65 EM versus only 0.31 F1 for the untrained backbone, suggesting that PRR-GRPO effectively learns to exploit the experience store, consistent with r_{exp}. The memory *graph* matters most before optimization - disabling it costs 2.36 F1 and 2.82 LLM-J on the untrained backbone, shrinking to 1.43 F1 after PRR-GRPO - showing that learned search behavior partly compensates for missing structural expansion while still benefiting from graph priors. Removing *reflection* leads to smaller but consistent regressions (0.74-1.50 F1) with a larger effect on the trained policy, indicating that the optimized policy better exploits revised long memory in later steps. Overall, graph expansion, reflection, and evolution address different failure modes (missed evidence, stale summaries, repeated rediscovery of strategies) and jointly drive the robustness of EvoMM.

Training robustness and cost. Table 4 compares training variants over three seeds. PRR-GRPO achieves 73.8 ± 0.4 F1, 59.6 ± 0.5 EM, and 85.2 ± 0.4 LLM-J, exceeding vanilla GRPO by 1.6, 2.5, and 2.1 points, respectively, and EvoMM *w/o RL* by 2.2, 3.1, and 2.5 points. The small standard

Table 5: Reported training cost per run and token counts per training step. Time is in hours; k denotes 10^3 tokens.

Training	Time/run	GPU-h/run	Sampled output tokens/step	Processed tokens/step
Vanilla GRPO	1.70 h	6.79	13.0k	33.2k
PRR-GRPO	2.24 h	8.97	17.9k	40.6k

deviations indicate limited variation across seeds, supporting the benefit of process-supervised optimization beyond vanilla GRPO. These gains come with higher training cost (Table 5): time per run increases from 1.70 to 2.24 hours and GPU-hours from 6.79 to 8.97 (approximately 32% in both cases), while sampled output and processed tokens per step rise from 13.0k / 33.2k to 17.9k / 40.6k, quantifying the additional training budget associated with the improved memory policy.

6 Conclusion

We presented EvoMM, a reinforced self-evolving multimodal agentic memory framework that unifies memory construction, retrieval, and evolution within a single trainable loop over a two-level mutable memory and a heterogeneous multimodal memory graph. The loop is optimized end-to-end with PRR-GRPO, which combines gated action-aware process rewards with dense credit assignment over a memory-action tree. Across Mem-Gallery, MMLongBench, and LoCoMo, EvoMM consistently surpasses strong text-only, multimodal, and agentic memory baselines, with the largest gains on multimodal grounding, contradiction reconciliation, and long-horizon temporal reasoning. These results suggest that treating memory revision and retrieval-skill accumulation as trainable agentic processes is a promising path toward lifelong multimodal agents.

Acknowledgments and Disclosure of Funding

This work was supported by Beijing Natural Science Foundation No. L233008, Beijing Municipal Science and Technology Project No. Z231100010323009, National Natural Science Foundation of China No. 62272467 and No. 625B2178.

References

[1] Marc-Antoine Allard, Arnaud Teinturier, Victor Xing, and Gautier Viaud. Experiential reflective learning for self-improving LLM agents. *CoRR*, abs/2603.24639, 2026. doi: 10.48550/ARXIV.2603.24639. URL https://doi.org/10.48550/arXiv.2603.24639.

[2] 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.

[3] Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-gallery: Benchmarking multimodal long-term conversational memory for mllm agents. *arXiv preprint arXiv:2601.03515*, 2026.

[4] Davide Caffagni, Federico Cocchi, Nicholas Moratelli, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Wiki-llava: Hierarchical retrieval-augmented generation for multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1818–1826, 2024.

[5] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5558–5570. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.375. URL https://doi.org/10.18653/v1/2022.emnlp-main.375.

[6] 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.

[7] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024.

[8] Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*, 2025.

[9] Matthew Fisher. Neural graph memory: A structured approach to long-term memory in multimodal agents. 2025.

[10] Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 1, 2025.

[11] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=Sx038qxjek`.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, Tao Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638, 2025. doi: 10.1038/S41586-025-09422-Z. URL `https://doi.org/10.1038/s41586-025-09422-z`.

[13] Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada,*

December 10 - 15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

[14] Jitesh Jain, Shubham Maheshwari, Ning Yu, Wen-mei Hwu, and Humphrey Shi. Augustus: An llm-driven multimodal agent system with contextualized user memory. *arXiv preprint arXiv:2510.15261*, 2025.

[15] Hongda Jiang, Xinyuan Zhang, Siddhant Garg, Rishab Arora, Shiunzu Kuo, Jiayang Xu, Aaron Colak, and Xin Luna Dong. Memory-qa: Answering recall questions based on multimodal memories. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 24244–24266. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.1234. URL https://doi.org/10.18653/v1/2025.emnlp-main.1234.

[16] Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25972–25981, 2025.

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

[18] Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*, 2024.

[19] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=Cw2xlg0e46.

[20] Niu Lian, Yuting Wang, Hanshu Yao, Jinpeng Wang, Bin Chen, Yaowei Wang, Min Zhang, and Shu-Tao Xia. From verbatim to gist: Distilling pyramidal multimodal memory via semantic information bottleneck for long-horizon video agents. *arXiv preprint arXiv:2603.01455*, 2026.

[21] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

[22] Yueqian Lin, Qinsi Wang, Hancheng Ye, Yuzhe Fu, Hai Li, Yiran Chen, et al. Hippomm: Hippocampal-inspired multimodal memory for long audiovisual event understanding. *arXiv preprint arXiv:2504.10739*, 2025.

[23] 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. *CoRR*, abs/2508.09736, 2025. doi: 10.48550/ARXIV.2508.09736. URL https://doi.org/10.48550/arXiv.2508.09736.

[24] Yihao Lu, Wanru Cheng, Zeyu Zhang, and Hao Tang. Mma: Multimodal memory agent. *arXiv preprint arXiv:2602.16493*, 2026.

[25] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37: 95963–96010, 2024.

[26] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and

Peter Clark. Self-refine: Iterative refinement with self-feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

[27] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, 2024.

[28] Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *CoRR*, abs/2310.08560, 2023. doi: 10.48550/ARXIV.2310.08560. URL https://doi.org/10.48550/arXiv.2310.08560.

[29] Cong Pang, Xuyu Feng, Yujie Yi, Zixuan Chen, Jiawei Hong, Tiankuo Yao, Nang Yuan, Jiapeng Luo, Lewei Lu, and Xin Lou. Ica: Information-aware credit assignment for visually grounded long-horizon information-seeking agents. *arXiv preprint arXiv:2602.10863*, 2026.

[30] German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. doi: 10.1016/J.NEUNET.2019.01.012. URL https://doi.org/10.1016/j.neunet.2019.01.012.

[31] Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In Sean Follmer, Jeff Han, Jürgen Steimle, and Nathalie Henry Riche, editors, *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, pages 2:1–2:22. ACM, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

[32] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[33] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36: 68539–68551, 2023.

[34] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

[35] Zhengliang Shi, Lingyong Yan, Weiwei Sun, Yue Feng, Pengjie Ren, Xinyu Ma, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Direct retrieval-augmented optimization: Synergizing knowledge selection and language models. *ACM Transactions on Information Systems*, 2025.

[36] Zhengliang Shi, Lingyong Yan, Dawei Yin, Suzan Verberne, Maarten de Rijke, and Zhaochun Ren. Iterative self-incentivization empowers large language models as agentic searchers. *CoRR*, abs/2505.20128, 2025. doi: 10.48550/ARXIV.2505.20128. URL https://doi.org/10.48550/arXiv.2505.20128.

[37] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

[38] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[39] Weiting Tan, Xinghua Qu, Ming Tu, Meng Ge, Andy T Liu, Philipp Koehn, and Lu Lu. Process-supervised reinforcement learning for interactive multimodal tool-use agents. *arXiv preprint arXiv:2509.14480*, 2025.

[40] 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.

[41] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

[42] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.510. URL https://doi.org/10.18653/v1/2024.acl-long.510.

[43] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

[44] Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

[45] Xixi Wu, Qianguo Sun, Ruiyang Zhang, Chao Song, Junlong Wu, Yiyan Qi, and Hong Cheng. Demystifying reinforcement learning for long-horizon tool-using agents: A comprehensive recipe. *arXiv preprint arXiv:2603.21972*, 2026.

[46] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

[47] B. Y. Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research. *CoRR*, abs/2511.18423, 2025. doi: 10.48550/ARXIV.2511.18423. URL https://doi.org/10.48550/arXiv.2511.18423.

[48] Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025.

[49] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[50] Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. Egolife: Towards egocentric life assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28885–28900, 2025.

[51] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

[52] Woongyeong Yeo, Kangsan Kim, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang. Universalrag: Retrieval-augmented generation over corpora of diverse modalities and granularities. *arXiv preprint arXiv:2504.20734*, 2025.

[53] Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. *CoRR*, abs/2512.02425, 2025. doi: 10.48550/ARXIV.2512.02425. URL https://doi.org/10.48550/arXiv.2512.02425.

[54] Yanwei Yue, Boci Peng, Xuanbo Fan, Jiaxin Guo, Qiankun Li, and Yan Zhang. Mem-t: Densifying rewards for long-horizon memory agents. *arXiv preprint arXiv:2601.23014*, 2026.

[55] Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents. *CoRR*, abs/2602.02474, 2026. doi: 10.48550/ARXIV.2602.02474. URL https://doi.org/10.48550/arXiv.2602.02474.

[56] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. GME: improving universal multimodal retrieval by multimodal llms. *CoRR*, abs/2412.16855, 2024. doi: 10.48550/ARXIV.2412.16855. URL https://doi.org/10.48550/arXiv.2412.16855.

[57] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: LLM agents are experiential learners. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19632–19642. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29936. URL https://doi.org/10.1609/aaai.v38i17.29936.

[58] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs, 2024. URL https://arxiv.org/abs/2312.07104.

[59] 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*, volume 38, pages 19724–19731, 2024.

[60] Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

A Limitations

The scope of this paper should be read as the perimeter of what the present study substantiates rather than as a deficiency of the framework itself: EvoMM is evaluated on three established long-horizon benchmarks under a fixed open-source backbone and compute budget that match prior agentic memory work, with the multimodal memory graph and PRR-GRPO hyperparameters chosen once on a held-out development split and reused throughout, so that improvements remain attributable to memory design rather than to evaluation drift. The iterative PLAN→RETRIEVE→REFLECT→ANSWER loop, combined with the structured-output sub-prompts, naturally requires more LLM calls and more decoded tokens per query than single-shot retrieval baselines, so deployments under tight latency budgets should expect a higher per-query inference cost; profiling that cost on representative long-horizon traces and trading it off against accuracy is an orthogonal engineering question we leave to follow-up work. Settings that lie outside this scope, including continually streaming or strictly online deployment, learned graph structures over very long histories, and deployment-time

controls for safety and provenance over on-the-fly memory revision, would primarily exercise system integration around EvoMM, and we view them as natural complementary directions for follow-up work rather than open weaknesses of the framework.

B Implementation Details

This appendix expands the configuration summarized in the main text and lists every hyperparameter that was held constant across Mem-Gallery, MMLongBench, and LoCoMo.

Backbone and retrievers. We use `Qwen3-VL-4B-Instruct` [2] as the policy π_θ and initialize both the reference policy π_{ref} and the training-time reward-mixing judge from the same checkpoint; the judge is queried but its parameters are frozen, and its contribution to $r(\tau)$ is intentionally kept small ($\alpha_{\text{judge}}{=}0.08$, gated by $r_{\text{ans}}{\geq}\eta_{\text{gate}}$ and further attenuated by $(1{-}0.9{\cdot}\text{EM})$) so that the policy cannot meaningfully overfit to it. To remove evaluator-policy alignment in the most prominently reported numbers, the test-time Mem-Gallery LLM-J metric is computed by an independent, frozen `Qwen3-32B` [49] backbone that is never used during training. Dense multimodal retrieval uses `GME-Qwen2-VL-2B` [56], and a lightweight keyword-and-entity inverted index provides complementary lexical coverage. Each `[RETRIEVE]` action returns up to 10 pages from the base retriever and is capped at 10 pages after graph merging.

Multimodal memory graph. Semantic edges use `all-MiniLM-L6-v2` page embeddings computed on CPU to avoid interfering with training GPUs; we keep the top $K_g{=}4$ neighbors per page with cosine similarity $\theta_{\text{sem}}{\geq}0.2$, and fall back to a lexical Jaccard builder over content lemmas when embeddings are unavailable. Temporal edges connect consecutive turns within a session, image-co-reference edges connect pages sharing the same image identifier, and keyword hyperedges are materialized from a stopword-filtered inverted index. At retrieval time, expansion adds up to $M{=}4$ extra pages beyond the seed set using a single fixed set of edge weights shared across all queries.

Graph expansion scoring. The expansion operator $\text{Expand}_{\mathcal{G}}$ in Equation (3) is realized as a single linear scoring rule shared across queries. For a seed set S_k from the base retriever, every candidate page p accumulates a score

$$s(p) \;=\; \underbrace{w_0 \cdot \mathbb{1}[p \in S_k]}_{\text{seed prior}} \;+\; \alpha_{\text{tmp}}\, s_{\text{tmp}}(p) \;+\; \alpha_{\text{sem}}\, s_{\text{sem}}(p) \;+\; \alpha_{\text{kw}}\, s_{\text{kw}}(p) \;+\; \alpha_{\text{img}}\, s_{\text{img}}(p), \tag{11}$$

where the four edge-type contributions s_{tmp}, s_{sem}, s_{kw}, s_{img} aggregate, respectively, the number of in-session temporal neighbors, the cosine-weighted semantic neighbors, the inverted-index hits over query terms, and matches on the image-identifier pattern; we use $w_0{=}3.0$, $\alpha_{\text{tmp}}{=}1.2$, $\alpha_{\text{sem}}{=}1.0$, $\alpha_{\text{kw}}{=}0.35$, $\alpha_{\text{img}}{=}1.1$ chosen once on a held-out development split. Three implementation choices apply uniformly: (i) within s_{tmp}, backward neighbors carry a $0.6\times$ multiplier relative to forward neighbors so the score respects discourse order; (ii) the keyword contribution is computed over the top 16 stopword-filtered query terms, and each term saturates at a fixed number of postings to prevent head terms from dominating expansion; and (iii) the image term is non-zero only when the query literally references an image identifier, so it is inactive on text-only queries by construction. Finally, candidates are sorted by $(-s(p), -p)$ and the top $M{=}4$ non-seed pages are merged into E_k.

Agentic reasoning loop. The agent interleaves `PLAN`, `RETRIEVE`, and `REFLECT` actions for at most $K_{\max}{=}3$ iterations before emitting `[ANSWER]`. Each intermediate step is limited to 256 new tokens and the forced answer step to 160 new tokens. Training rollouts use a 1024-token prompt budget and a 1024-token response budget, while evaluation rollouts use a 512-token response budget. The experience store retrieves the top $P{=}3$ entries by stopword-filtered keyword overlap with the question, each carrying at most 3 distilled lessons, and admits new entries only when the trajectory's answer F1 exceeds 0.35.

Raw memory construction. Both the entity list K_i and the extractive summary σ_i in $m_i = (d_i, I_i, K_i, \sigma_i, p_i)$ are produced by lightweight rules so that memory construction stays offline and reproducible: K_i is obtained by a capitalized-span regex (`[A-Z][a-z]+(?:\s+[A-Z][a-z]+)*`, after stripping leading sentence-final positions and stopwords; deduplicated and truncated to 15

entries), and σ_i is a 200-character extractive summary that drops image placeholders and is cut at the last whitespace boundary. Image references I_i are read directly from the dialogue's image fields; if absent, `<image>`/`<ImageHere>` occurrences in d_i are mapped to placeholder identifiers. Long-memory entries l_s are the only summaries produced by π_θ itself, via the MEMORYAGENT prompt (Appendix E.7).

Reward configuration. Outcome and process reward weights are listed in Table 6. The process gate on answer F1 is η_{gate}=0.2: below this threshold all five process rewards (evidence, plan, reflection, format, experience) are zeroed out to prevent rewarding well-structured but incorrect trajectories. The training-time judge (a frozen copy of `Qwen3-VL-4B-Instruct`) produces a scalar score per answer that is mixed into the outcome reward with coefficient α_{judge}=0.08, further attenuated by $(1 - 0.9 \cdot \text{EM})$ when exact match is already high to avoid double counting. Conciseness uses a length-adaptive soft cap on the length ratio ρ with $\rho_{\text{soft}} \in \{1.10, 1.16, 1.20, 1.24\}$ for reference-answer word counts of ≤ 8, ≤ 20, ≤ 50, and > 50, respectively. The piecewise-linear decay f_{conc} in Eq. (7)'s conciseness term is 1 for $\rho \leq \rho_{\text{soft}}$, $\max(0,\, 1 - 0.95(\rho - \rho_{\text{soft}}))$ for $\rho_{\text{soft}} < \rho \leq 2$, and $\max(0,\, (1 - 0.95(2 - \rho_{\text{soft}})) - 0.42(\rho - 2))$ for $\rho > 2$, minus an absolute-excess penalty $\min(0.62,\, 0.042 \cdot \max(0,\, |\hat{y}| - |y^\star| - \max(5,\, \lfloor 0.12|y^\star| \rfloor)))$. The tail penalty in Eq. (7) uses α_{tail}=0.038, κ_{tail}=0.42.

Table 6: Reward coefficients and tree-search hyperparameters used in `PRR-GRPO`. The five process weights $\{\lambda_c\}_{c \in \mathcal{P}}$ are intentionally small relative to λ_{ans} and λ_{conc} so they act as auxiliary shaping signals on top of the outcome reward. All coefficients are held constant across Mem-Gallery, MMLongBench, and LoCoMo.

Outcome term (r_{ans})		Trajectory weights		Tree search	
w_1 (token F1)	0.65	λ_{ans}	0.62	$K_{\max}$ (PSR loops)	3
w_2 (EM)	0.25	λ_{conc}	0.23	B (branch factor)	3
w_3 (SoftEM)	0.10	λ_{evd}	0.05	$D_{\max}$ (branch depth)	2
η_{gate}	0.20	λ_{pln}	0.03	γ (tree discount)	0.88
α_{judge}	0.08	λ_{ref}	0.03	G (rollouts/prompt)	8
		λ_{fmt}	0.02	P (top-P experiences)	3
		λ_{exp}	0.02		

Process reward scoring. The five process rewards in Eq. (7) are computed deterministically from the trajectory log (no extra LLM call):

- *Evidence relevance* $r_{\text{evd}} = |\text{tok}(y^\star) \cap \text{tok}(\bigcup_k E_k)|/|\text{tok}(y^\star)|$ over stopword-filtered tokens.
- *Plan specificity* r_{pln} averages, over all `[PLAN]` nodes, $0.5 \cdot \min(|c_k|/15,\, 1) + 0.5 \cdot \mathbb{1}[\, c_k$ contains any of {*need, find, check, verify, look, search, determine, confirm, recall, remember, information, evidence*}$\,]$, capped at 1.0 per trajectory.
- *Reflection validity* r_{ref} is 0.5(fraction of update lines whose op tag is in $\{\text{ADD}, \text{MODIFY}, \text{DELETE}, \text{KEEP}\}$)$+$ $0.5 \cdot$(fraction of update lines whose detail span ≥ 3 words), with a floor of 0.1 when `[REFLECT]` fired but produced no parseable update.
- *Format compliance* $r_{\text{fmt}} = 0.4 \cdot$(fraction of well-formed action tags) $+ 0.15 \cdot \mathbb{1}$[`[PLAN]` present] $+$ $0.15 \cdot \mathbb{1}$[`[ANSWER]` present] $+ 0.15 \cdot \mathbb{1}$[starts with `[PLAN]`] $+ 0.15 \cdot \mathbb{1}$[ends with `[ANSWER]`], capped at 1.0.
- *Experience quality* r_{exp} inherits the `quality_score` returned by the experience-extraction step (Appendix E.8), bounded to $[0, 1]$.

All five terms are normalized to $[0, 1]$ before the gate and weighted aggregation in Eq. (7), so they remain interpretable across rollouts of varying length.

Reinforcement learning. `PRR-GRPO` is trained for 200 optimization steps. At each step we sample G=8 rollouts per prompt with rollout batch size 2 and global batch size 4, using temperature 0.6; evaluation is interleaved every 10 steps. The clipped surrogate uses asymmetric bounds ϵ=0.2 and ϵ'=0.28. Auxiliary objectives include a low-variance KL penalty against π_{ref} with coefficient

β_{kl}=0.02 and an entropy bonus with coefficient β_{ent}=5×10^{-4}. The reward discount is γ=0.9, and advantages are dual-scale normalized (intra-group + inter-batch) and clipped to $[-10, 10]$. Optimization uses Adam with β_1=0.9, β_2=0.98, weight decay 0.1, and a constant learning rate of 5×10^{-7}.

Infrastructure. Training is implemented on `Slime` [60] with a `Megatron-LM` [38] backend and runs on 8×NVIDIA A800 GPUs. The actor occupies 4 GPUs (tensor-model-parallel size 4, pipeline size 1, context parallel 1), and the remaining 4 GPUs serve rollouts via `SGLang` [58], with 2 GPUs dedicated to actor-policy rollouts and 2 GPU hosting the frozen training-time judge model. The independent `Qwen3-32B` LLM-J evaluator is invoked offline at evaluation time on a separate node and is not part of the training loop. We enable precision-aware Adam CPU offloading, dynamic batching with a 768-token per-GPU budget, full-granularity activation recomputation, and `FlashAttention`. Unless otherwise noted, this configuration is shared across Mem-Gallery, MMLongBench, and LoCoMo.

C Detailed Baseline Descriptions

This appendix describes every baseline used in our main results, together with the concrete configuration we adopt to ensure a fair comparison with `EvoMM`. Unless otherwise noted, all memory baselines are reproduced under our unified evaluation harness using the same backbone (`Qwen3-VL-4B-Instruct` [2] on Mem-Gallery and MMLongBench, and an off-the-shelf `Qwen2.5-14B-Instruct` [32] on LoCoMo), the same dialogue history, and the same per-query token budget.

C.1 Text-only Memory Baselines (Mem-Gallery)

Full Memory (Text). A long-context baseline that linearly concatenates the entire dialogue history into a single textual context, dropping all visual content, and prompts the backbone for the answer. It serves as the no-memory upper bound on context utilization but is bounded by the backbone's context window. Following the Mem-Gallery default, we truncate from the front when the budget is exceeded.

FIFO. A short-term memory baseline that retains only the most recent turns within a fixed token budget and concatenates them in temporal order. It captures recency but discards distant evidence, and corresponds to a sliding-window controller without retrieval.

NaiveRAG. A standard retrieval-augmented baseline that embeds each dialogue turn with a sentence encoder and recalls top-k records by cosine similarity at query time. There is no graph structure, no reflection, and no memory revision; this isolates the contribution of plain dense retrieval.

Reflexion [37]. A self-reflection baseline that, after each query, prompts the model to verbalize a global insight, which is then prepended to future contexts. Memory updates are entirely prompt-driven and unstructured. We follow the original Reflexion prompts shipped with Mem-Gallery and run a single reflection pass per query.

HippoRAG [13]. A neurobiologically inspired retrieval system that builds an entity-relation knowledge graph from the corpus and applies personalized PageRank for multi-hop retrieval. We use the official HippoRAG library, indexed once over the full dialogue history, and retrieve the top-k documents with the recommended hyperparameters.

MemGPT [28]. A hierarchical memory baseline that emulates an operating-system view of LLM memory, separating a small in-context working memory, a recall storage of recent turns, and an archival storage queried via embedding lookup. Recursive summarization is invoked when the working memory overflows.

MemoryOS [16]. A recent agentic memory system that organizes memory into short-term, mid-term, and long-personal stores with explicit promotion / eviction rules. We adopt the authors' default thresholds and use the same embedding backbone as our other RAG baselines.

C.2 Multimodal Memory Baselines (Mem-Gallery)

Full Memory (MM). A multimodal long-context baseline that concatenates both text and images and applies a joint text+image token budget, $N_{\text{text}} + N_{\text{image}} \cdot T_{\text{img}} \leq L_{\max}$, keeping the most recent items when truncating. Unlike *Full Memory (Text)*, raw images are preserved rather than captioned.

UniversalRAG [52]. A modality-aware retrieval pipeline that learns to route a query to one of {NO-RETRIEVE, DOCUMENT, IMAGE} and then performs the corresponding text or image retrieval. We use `GME-Qwen2-VL-2B` as the shared encoder for fair comparison with `EvoMM`.

NGM [9]. A neural graph memory that stores observations as nodes in a graph storage and retrieves with multimodal embeddings followed by graph expansion. We use the default Mem-Gallery configuration with the same multimodal encoder as UniversalRAG.

AUGUSTUS [14]. A concept-tagged contextual memory that maintains a tag graph over multimodal entries and retrieves by combining concept-based and embedding-based scoring. Concepts are extracted with the LLM-based extractor, matching the original paper.

MuRAG [5]. A multimodal RAG baseline that jointly embeds text and image entries with the same multimodal encoder and retrieves top-k items by similarity. It serves as the canonical multimodal RAG anchor.

C.3 Agentic Memory Baselines (Mem-Gallery)

Generative Agent [31]. The pioneer agentic memory model with weighted retrieval (recency, importance, similarity) and periodic self-reflection that summarizes recent observations into higher-level memories. We use the default importance-judging prompt and reflection cadence.

GAM [47]. A general agentic memory framework that decouples memory writing, indexing, and answering into specialized agents and produces precomputed answers per query.

A-Mem [46]. An adaptive agentic memory that constructs Zettelkasten-style notes with automatically generated links and updates them online. We use the authors' open-source implementation with the recommended encoder.

M3-Agent [23]. A multimodal long-term memory-augmented agent that interleaves perception, memory writing, and reasoning over dialogue and visual streams. Memory is organized into episodic and semantic stores, both queried at answer time.

WorldMM [53]. A dynamic multimodal memory-augmented agent with three coupled stores—episodic memory retrieved with HippoRAG-style PPR, semantic memory retrieved with personalized PageRank over an entity graph, and visual memory retrieved with multimodal embeddings. We integrate WorldMM into our harness using the authors' source and share the same VLLM-served backbone for answering.

C.4 MMLongBench Baselines

Direct (raw context). The official long-context baseline of MMLongBench [25]: for each query, all K document pages (interleaved text and images) are placed in the prompt verbatim under the official user / system templates, and the backbone is asked to answer with at most 128 generation tokens. We sweep the per-query page budget $K \in \{8, 16, 32, 64, 128\}$ over the LongDocURL, MMLongDoc, and SlideVQA subsets, exactly mirroring the official evaluation grid. Scoring follows the official `eval_docqa_score` routine, which mixes type-aware exact match, numeric equality, soft containment, and token-F1, then averages over the answer types.

MuRAG [5] on MMLongBench. A canonical multimodal RAG baseline ported to the document-QA setting. The same K pages constitute the per-query corpus; we embed each page (interleaved text and image) with `GME-Qwen2-VL-2B`, retrieve the top-10 pages by cosine similarity to the question,

and prompt the backbone with this fixed-size top-10 context under the official MMLongBench answer template. There is no graph expansion, no reflection, and no memory revision; this isolates the gain attributable to plain dense retrieval at the same retrieval budget as EvoMM.

WorldMM [53] on MMLongBench. A dynamic multimodal memory-augmented agent that maintains episodic, semantic, and visual stores and combines HippoRAG-style PPR over the entity graph with multimodal retrieval. We adapt it to the document-QA setting by treating each page as one observation, and cap the recall list at top-10 to match the retrieval budget of EvoMM; everything else uses the authors' default thresholds and the same backbone served via vLLM. WorldMM thus serves as a strong agentic-memory baseline with comparable retrieval budget but without the joint planning–reflection loop and PRR-GRPO-trained policy of EvoMM.

EvoMM **on MMLongBench.** For EvoMM, the same K pages serve as the dialogue history; pages are converted into raw memory entries with one page per turn and indexed by our heterogeneous memory graph (Section B). The agent then runs the same PLAN→RETRIEVE→REFLECT→ANSWER loop, retrieving up to ten pages per [RETRIEVE] step. The answer template is byte-identical to MMLongBench's load_doc_qa template, ensuring that any difference against *Direct* / MuRAG / WorldMM reflects the memory mechanism and not the prompt.

C.5 LoCoMo Baselines

Long-LLM and RAG. *Long-LLM* concatenates the whole multi-session dialogue into a single context and asks the backbone to answer; it serves as a no-memory upper bound subject to the context window. *RAG* replaces the full context with top-k utterances retrieved by sentence embeddings (we use the same encoder as our NaiveRAG on Mem-Gallery).

A-Mem, Mem0, MemoryOS, HippoRAG, LightMem, GAM. We reproduce the recent memory baselines under a shared protocol. A-Mem [46], MemoryOS [16], HippoRAG [13], and GAM [47] reuse the same configurations as on Mem-Gallery. Mem0 [6] is a production-oriented memory system that extracts atomic factual statements and maintains a deduplicated key–value store with periodic compaction; we use the official mem0 library with default extraction prompts. LightMem [8] is a lightweight memory-augmented generation framework that compresses dialogue into a small set of compressed memory tokens; we follow the authors' recommended compression ratio.

Question types and metrics. LoCoMo provides four question types: *Single-Hop*, *Multi-Hop*, *Temporal*, and *Open-Domain*. For every method we report token F1 and BLEU-1 against the reference answer following the original protocol, evaluating each question conditioned on the corresponding multi-session conversation.

D Evaluation Protocols and Scoring Details

Mem-Gallery. We use the official Mem-Gallery [3] test split with an 8:1:1 train/val/test partition. Each query is annotated with one of nine categories: AR (answer refusal), CD (contradiction detection), FR (factual recall), KR (knowledge retrieval), MR (multi-hop reasoning), TR (temporal reasoning), VR (visual recall), VS (visual search), and TTL (long-range titling). Per-category metrics are computed by the official Mem-Gallery evaluator's per-category answer parsers (e.g., AR uses a refusal classifier, CD uses a contradiction template, VS uses image-grounded matching); these parsers belong to the benchmark's own scoring logic and are not consumed by EvoMM's policy or retrieval. The *Overall* columns report token F1, exact match (EM) after light normalization, and an LLM-judge score (LLM-J) produced by an independent, frozen Qwen3-32B [49] backbone that is never updated by PRR-GRPO and never shares parameters with the policy, so LLM-J is not subject to the reward-hacking concern that would arise if the policy and the evaluator coincided.

MMLongBench DocQA. For each $K \in \{8, 16, 32, 64, 128\}$ and each subset in {LongDocURL, MMLongDoc, SlideVQA}, the evaluator processes one row at a time. Each row contains a list of page images and the question. For *Direct* we use the unmodified MMLongBench prompt template; for memory-based methods (including EvoMM) we (a) build a per-query page store, (b) call the memory's recall to select up to top_k=10 candidate pages, then (c) reuse the official MMLongBench

parse_output and eval_docqa_score for scoring. Caches and score files are written under results so that runs are fully resumable.

LoCoMo. LoCoMo [27] consists of long multi-session dialogues with question-answer pairs of four types. We follow the standard split and evaluate with token F1 and BLEU-1 per type. For `EvoMM` on LoCoMo, raw memory pages are constructed at the granularity of single turns within each session, long memories are produced by summarizing each session, and the experience store is shared across questions of the same dialogue, allowing skill transfer between related queries.

Reproducibility notes. All evaluations use a single backbone served via vLLM with deterministic decoding (`temperature=0.0`, `top_p=1.0`) for the answer step; only `[PLAN]`, `[RETRIEVE]`, and `[REFLECT]` steps use the rollout temperature of 0.6. Random seeds and the per-query page caches are persisted alongside results. Wherever a baseline ships with its own backbone-specific prompts, we keep those prompts unchanged but route generation through the same vLLM endpoint, so that comparisons reflect memory design rather than prompt engineering or serving-stack differences.

E Inference-Time Prompt Templates

This appendix lists every prompt that drives the inference-time `EvoMM` agentic loop together with the structured JSON schema each prompt produces. All sub-prompts are issued to the same policy π_θ (`Qwen3-VL-4B-Instruct` on Mem-Gallery / MMLongBench, `Qwen2.5-14B-Instruct` on LoCoMo); no external models are involved. To avoid repetition, we factor out the conventions shared by all prompts and only show, for each prompt, the role description and the parts that are unique to that step. Placeholders in curly braces (e.g. `{request}`) are filled in at runtime from the current trajectory state.

Common conventions. All sub-prompts share the following contract and we omit it from the individual templates below: (i) the model may reason inside an optional `<think>...</think>` block, but the system only consumes the JSON object emitted after `</think>`; (ii) the JSON keys listed in Table 7 are required and any extra keys are rejected; (iii) the model must not invent facts that do not appear in the inputs and must avoid meta phrases such as "the evidence says" or "the user said"; (iv) malformed outputs trigger one re-prompt before the system falls back to the previous step's content.

E.1 Output JSON Schemas

Table 7 summarizes the JSON contract for each prompt. INFOCHECK, GENERATEREQUESTS, and SELFCORRECTION jointly implement the `[REFLECT]` action: the first decides whether to re-enter the loop, the second formulates the next `[RETRIEVE]` query if the loop continues, and the third realizes in-place revision of $\mathcal{M}^{\text{long}}$ when evidence contradicts or critically extends the current draft.

Table 7: JSON output contract for each prompt in the `EvoMM` pipeline.

Prompt	Action role	Required JSON keys
PLANNING	`[PLAN]`	`info_needs`, `tools`, `keyword_collection`, `vector_queries`, `page_index`
INTEGRATE	`[RETRIEVE]` merge	`content`, `sources`
INFOCHECK	`[REFLECT]` sufficiency	`enough` (bool)
GENERATEREQUESTS	`[REFLECT]` follow-up	`new_requests` (array, ≤ 5)
SELFCORRECTION	`[REFLECT]` memory edit	`critique`, `is_correct`, `corrected_answer`
MEMORYAGENT	long-memory writer ($\mathcal{M}^{\text{long}}$)	single paragraph (no JSON)
EXPERIENCEEXTRACT	cross-query distill ($\mathcal{X}$)	`effective_queries`, `key_insight`, `self_assessment`, `lessons`

E.2 Planning Prompt (`[PLAN]`)

The `[PLAN]` action turns the current `{request}` and long-memory snapshot `{memory}` into a concrete retrieval plan over the three available retrieval tools.

```
You are the PlanningAgent. Your job is to generate a concrete retrieval
plan for how to gather information needed to answer the QUESTION, using
```

```
QUESTION and the current MEMORY (abstracts of all messages so far).

PROCEDURE
1. Interpret QUESTION using MEMORY. Identify what is needed to satisfy
   QUESTION.
2. Break that need into concrete "info needs": specific sub-questions
   you must answer to fully respond to QUESTION.
3. For each info need, decide which retrieval tools are useful. You may
   assign multiple tools to the same info need:
   - "keyword"    for exact entities / functions / key attributes;
   - "vector"     for conceptual / "how" / "why" questions;
   - "page_index" only when MEMORY already names relevant page IDs.
4. Build the final plan:
   - "info_needs":         all specific sub-questions / missing facts;
   - "tools":              subset of {keyword, vector, page_index} you
                           will actually use (more than one allowed);
   - "keyword_collection": short keyword-style queries you will issue;
   - "vector_queries":     short natural-language queries you will issue;
   - "page_index":         integer page indices to read fully (<= 5).

RULES
- Be specific. Avoid vague items like "get more details".
- Make keyword and vector queries independent rather than duplicated.
- Do NOT invent tools or page indices.
- You are only planning retrieval; do NOT answer QUESTION here.
```

The runtime then invokes the graph-augmented retriever (Eq. 3; cf. Appendix B) on each `keyword`/`vector`/`page_index` payload and forwards the merged hits to INTEGRATE.

E.3 Integrate Prompt ([RETRIEVE] merge)

After each [RETRIEVE] the agent merges the newly retrieved evidence into a working summary; this output is *not* the final answer but a consolidated factual summary used by later loop iterations and by [ANSWER].

```
You are the IntegrateAgent. Build an UPDATED_RESULT that is a consolidated
factual summary relevant to QUESTION, using:
  RESULT           - the current working notes about QUESTION,
  EVIDENCE_CONTEXT - newly retrieved supporting evidence.

UPDATED_RESULT must:
1. Keep on-topic, correct information from RESULT.
2. Add new well-supported facts from EVIDENCE_CONTEXT (entities, numbers,
   versions, decisions, timelines, outcomes, constraints).
3. Drop content that is off-topic for QUESTION.

This is NOT a final answer. Do NOT add interpretation or recommendations
beyond what is explicitly stated. "sources" lists the page_ids that
support the included facts.
```

E.4 Information-Sufficiency Check ([REFLECT] part 1)

Decides whether the PLAN→RETRIEVE→REFLECT loop should terminate; only judges completeness, never answers.

```
You are the InfoCheckAgent. Decompose REQUEST into the key pieces of
information required, then check whether RESULT covers each piece
clearly and specifically enough that a final answer could be written
directly from it without further retrieval.
```

```
Return "enough"=true ONLY IF RESULT covers all required pieces with
sufficient clarity and specificity; otherwise return "enough"=false.
Do NOT answer REQUEST. Do NOT invent facts.
```

E.5 Follow-up Request Prompt ([REFLECT] part 2)

When INFOCHECK returns `false`, this prompt produces focused retrieval questions that drive the next [PLAN].

```
You are the FollowUpRequestAgent. Identify pieces of information that
REQUEST needs but are missing or unclear in RESULT. For each missing
piece emit ONE standalone retrieval question that:
  - names concrete entities / modules / datasets / events when known,
  - asks for factual information that retrieval can provide
    (not "analyze", "infer", or "judge").

Rank the questions from most critical to least critical. Produce at
most 5 questions. Do NOT answer REQUEST yourself. Do NOT generate
vague requests like "Get more info".
```

E.6 Self-Correction Prompt ([REFLECT] part 3)

This prompt operationalizes the in-place memory revision aspect of [REFLECT] described in Section 4.1: it inspects the current draft summary against the retrieved evidence and decides whether to issue an ADD/MODIFY/DELETE edit to long memory. The five-clause *conservative policy* (A–E) is what actually controls when revision fires; in particular, clause (A) guarantees that absence of mention in the evidence is *never* treated as a deletion signal, which is what keeps the revision step safe under partial retrieval.

```
You are a strict but pragmatic fact-checker and editor for long-conversation
memory. Inputs: QUESTION, numbered EVIDENCE, DRAFT (current memory summary).

GOAL: produce a memory summary that is not contradicted by EVIDENCE and is
most useful for answering QUESTION.

CONSERVATIVE POLICY
A) Do NOT delete or rewrite statements solely because they are absent from
   EVIDENCE. Lack of mention is not a reason to change.
B) MUST edit if a statement CONTRADICTS EVIDENCE.
C) SHOULD edit on a CRITICAL OMISSION: EVIDENCE contains a fact essential
   for answering QUESTION whose omission would be misleading.
D) MAY make small HELPFUL edits even without (B)/(C), but ONLY if they
   improve answerability AND are fully evidence-grounded (reorder/condense
   to highlight question-relevant facts, add evidence-supported details,
   remove hedging without losing meaning).
E) Never add speculation; use only what EVIDENCE explicitly supports.

JUSTIFICATION
Every change must cite one or more EVIDENCE item numbers; if no change is
made, say so.

OUTPUT
"critique"        - "No change needed" OR which of (B)/(C)/(D) applied
                    and the EVIDENCE item numbers used;
"is_correct"      - true iff no edits were made;
"corrected_answer"- if is_correct, byte-identical to DRAFT; otherwise the
                    minimally edited summary (no critique text inside).
```

The system converts the diff between `DRAFT` and `corrected_answer` into the discrete operation set $\{(\mathrm{op}_j, \delta_j)\}$ ($\mathrm{op}_j \in \{\text{ADD}, \text{MODIFY}, \text{DELETE}\}$) used in Section 4.1: added bullets become ADD,

replaced bullets become MODIFY, removed bullets become DELETE, and bullets that are byte-identical are implicitly KEEP.

E.7 Memory Writer Prompt (MEMORYAGENT)

Used to construct each session-level entry of $\mathcal{M}^{\text{long}}$ before reasoning starts; this is invoked once per session during memory construction, not inside the agentic loop.

```
You are the MemoryAgent. Write ONE concise paragraph that can be stored as
long-term memory. Inputs: MEMORY_CONTEXT (prior abstracts, used only for
disambiguation and consistent terminology) and INPUT_MESSAGE (the new
turn(s) to summarise).

Extract from INPUT_MESSAGE every specific, memory-relevant item: plans,
goals, decisions, requests, preferences; actions taken, next steps,
assignments, responsibilities; problems, blockers, open questions; and
specific facts (names, dates, numbers, locations).

Use MEMORY_CONTEXT only to keep terminology consistent and to disambiguate
entities; do NOT invent or add information that appears only in
MEMORY_CONTEXT and is not implied by INPUT_MESSAGE.

Output exactly ONE concise paragraph. No bullet points, no meta phrases
("the user said...", "the conversation is about..."), no advice or
opinions, no questions. The abstract must be self-contained, factual,
and specific.
```

E.8 Experience Extraction Prompt (EXPERIENCEEXTRACT)

After each completed trajectory, this prompt distills one entry into the cross-query experience store $\mathcal{X}$ in Section 4.1. The trace inputs (search rounds, tools used, queries issued, abbreviated integrated result) are deterministically assembled from the trajectory log.

```
You are an experience analyst. Given the research trace for a QA task -
QUESTION, search_rounds, tools_used, queries_issued, and an abbreviated
integrated_result - distill reusable knowledge:

- "effective_queries": the queries from the trace that retrieved the most
  useful evidence;
- "key_insight":       the single most important finding (one sentence);
- "self_assessment":   brief judgement on whether the integrated result is
  complete and accurate;
- "lessons":           2-4 short, actionable lessons that would help a
  future agent answer a similar question faster or more accurately
  (which retrieval tool worked best, what was hardest to find, pitfalls
  to avoid).
```

Mapping to the experience tuple. Each accepted entry $x = (q, \{r_k^\star\}, \kappa, \xi, \mathcal{L}_x)$ in Section 4.1 corresponds, in the same order, to the JSON keys `question`, `effective_queries`, `key_insight`, `self_assessment`, and `lessons`. The quality gate $\text{quality}(x) \geq \eta$ at evaluation time is a reference-free proxy: it requires (i) all four JSON content fields to be present and non-empty and (ii) the self-assessment heuristic to be non-negative, so the store is never gated by ground-truth labels during testing.

F Case Study: Self-Evolving Memory on a Major-Choice Question

To illustrate how on-the-fly memory revision changes what the policy retrieves and answers, we walk through one Mem-Gallery query whose ground-truth answer cannot be recovered from any single dialogue turn, but only from a state that evolves across the conversation.

Case study at a glance

Domain	`Education_Career_Research_Lifestyle`
Character	Lin, session `D1`
Category	KR (knowledge retrieval)
Backbone	`Qwen3-VL-4B` (`PRR-GRPO`, step 199)

Query. *"Did Lin ultimately switch majors or stay in their current one?"*
Reference. Lin ultimately stayed in their current major and declared a minor in psychology.
`EvoMM` **answer.** Lin stayed in their current major but declared a minor in psychology. **F1** $= 0.91$
LLM-J $= 1.0$

Why this query is hard for static memory. The question targets a *final, evolved state*, yet the dialogue contains earlier statements that are locally consistent but globally outdated: Lin first expresses interest in *switching* to psychology and only later commits to a *minor*. A static index that surfaces an early turn (e.g. `D1:2`, *"I'm interested in psychology, but I'm worried about delaying my graduation if I switch now"*) without reconciling later evidence can plausibly answer either "switch" or "undecided", neither of which matches the ground truth.

Trajectory of long-memory revisions. Table 8 traces the seven `[REFLECT]` edits that progressively stabilize the long-memory entry for *Lin's major*. The two highlighted rows are pivotal: `D1:8` MODIFY-rewrites the binary "switch vs. stay" framing into the minor-as-compromise state, and `D1:13` MODIFY-commits the final state while DELETE-ing earlier contradicting bullets rather than appending alongside them.

Table 8: Long-memory evolution over the seven turns relevant to Lin's major-choice question. Highlighted rows mark the two pivotal `[REFLECT]` edits. Image columns reference pages reachable via the image-co-reference edges $\mathcal{E}_{\text{img}}$ in the memory graph; clause A of the conservative policy (Appendix E.6) prevents `D1:20` from deleting the prior commitment merely because it is not re-stated.

Turn	Image	Dialogue excerpt	Op	Long-memory state after `[REFLECT]`
`D1:2`	–	"Interested in psychology, but worried about delaying graduation if I switch now."	ADD	Real interest in psychology; switching risks graduation delay.
`D1:3`	–	"My family thinks I should stick with my current major for stability."	ADD	+ external constraint: family preference for stability.
`D1:6`	`D1_IMG_002`	"I'll set up a meeting with my academic advisor to discuss my options."	ADD	Decision not yet final; will be revised after advisor input.
`D1:8`	–	"Advisor mentioned minoring in psychology *instead of* switching majors."	MODIFY	**Pivot.** Replaces "switch vs. stay" framing with "minor as compromise".
`D1:12`	`D1_IMG_003`	"I started searching the library for psychology materials to prepare."	MODIFY	Interest becomes action: preparing for the minor.
`D1:13`	–	"Decided to declare a minor in psychology while keeping my current major."	MODIFY	**Commit.** Final state; earlier contradicting bullets removed via DELETE.
`D1:20`	`D1_IMG_004`	"I'm satisfied with the minor; it complements my major well."	ADD	Reinforces final state with later validation.

Evolved memory capsule. After all seven edits, the long-memory entry for Lin's major consolidates into a single self-contained paragraph that subsequent queries can plan against:

> *Lin ultimately chose to declare a psychology minor while keeping her current major. This decision evolved from her initial interest in psychology, concerns about delayed graduation, family preference for stability, and advisor guidance suggesting a minor as a compromise. Later, Lin expressed satisfaction with the choice and said the minor complements her major well.*

Query-time retrieval and answer. At inference, the PLANNING prompt issues two retrieval queries (one keyword, one vector) targeting "major", "minor", "psychology", and "advisor". The graph-augmented retriever returns the merged top-10 pages, with the three pivotal turns surfaced alongside the earlier, now-superseded statements:

$$E = \Big\{ \underbrace{\texttt{D1:13}}_{\text{commit}}, \texttt{D1:1}, \underbrace{\texttt{D1:8}}_{\text{pivot}}, \texttt{D1:2}, \texttt{D1:3}, \texttt{D1:10}, \texttt{D1:7}, \texttt{D5:24}, \underbrace{\texttt{D1:20}}_{\text{validation}}, \texttt{D1:9} \Big\}.$$

INTEGRATE consolidates them, INFOCHECK accepts the result as sufficient (no follow-up retrieval triggered), and `[ANSWER]` emits *"Lin stayed in her current major and declared a minor in psychology."* - byte-identical to the capsule's terminal claim and aligned with the reference answer up to surface phrasing.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state three contributions - identifying the limitations of decoupled multimodal memory-augmented agents, proposing `EvoMM` with a heterogeneous memory graph, on-the-fly memory revision, and a cross-query experience store, and introducing `PRR-GRPO` for dense process supervision - and these match the formal description in Section 4.1–4.2 and the empirical findings in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix A delineates the empirical scope of the study, including the single open-source backbone scale, fixed compute budget, choice of three established long-horizon benchmarks, automated evaluation metrics, a single conservative `PRR-GRPO` configuration tuned once on a held-out development split, and the higher per-query inference cost induced by the iterative `PLAN`→`RETRIEVE`→`REFLECT`→`ANSWER` loop and its structured-output sub-prompts; it also identifies natural extensions left to follow-up work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper proposes an empirical agentic memory framework and a process-supervised reinforcement learning training scheme; it does not state formal theorems or rely on proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix B specifies the backbones (policy / reference / training-time judge / independent LLM-J evaluator), dense and lexical retrievers, multimodal memory graph construction with the explicit linear scoring rule and its weights, raw memory construction, agentic loop budgets, reward configuration with all coefficients listed in Table 6, the deterministic process-reward scoring rules, the `PRR-GRPO` optimization recipe, and the training infrastructure; Appendix E provides every inference-time prompt template together with its JSON schema (Table 7); Appendix D details the evaluators and scoring routines used for each benchmark.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All benchmarks (Mem-Gallery, MMLongBench, LoCoMo) and models (`Qwen3-VL-4B-Instruct`, `Qwen2.5-14B-Instruct`, `Qwen3-32B`, `GME-Qwen2-VL-2B`, `all-MiniLM-L6-v2`) are publicly available and cited in Section 5 and Appendix C; we release the `EvoMM` training and evaluation code, prompt templates, and configuration files at an anonymous repository linked from the abstract footnote, and will release the `PRR-GRPO`-trained policy checkpoint upon acceptance. Appendix B and Appendix E document the exact hyperparameters and prompts. Our code is also available at an anonymous repository.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 summarizes datasets, splits, baselines, backbones, and the high-level training setup; Appendix B additionally specifies the graph expansion weights, agentic loop budgets, reward weights and gates (Table 6), the Adam optimizer settings, learning rate, KL and entropy coefficients, advantage clipping, and dual-scale normalization, all kept fixed across the three benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the substantial cost of full PRR-GRPO rollouts over multimodal long-horizon trajectories on 8×NVIDIA A800 GPUs (Appendix B), we report single-seed results following the protocol of prior agentic memory work; we keep all hyperparameters, seeds, decoding settings, and the served backbone identical across baselines and EvoMM variants so that comparisons remain controlled.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B (Infrastructure paragraph) reports the hardware (8×NVIDIA A800 GPUs), the actor / rollout / judge GPU split, parallelism strategy (tensor-model-parallel size 4, pipeline size 1), dynamic batching budget, and serving stack (Slime + Megatron-LM with SGLang rollouts).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All experiments are conducted on publicly released benchmarks (Mem-Gallery, MMLongBench, LoCoMo) under their original protocols, without involving human subjects, crowdsourcing, or sensitive personal data, and the paper preserves anonymity in line with the double-blind submission policy.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Multimodal long-horizon memory-augmented agents can support extended assistive interactions over heterogeneous user histories, but raise concerns about the privacy of accumulated user data and the integrity of mutable memory under continual updates; we explicitly note in Appendix A that productionizing on-the-fly memory revision should be paired with deployment-time controls for safety, privacy, and provenance over ADD/MODIFY/DELETE operations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release any new pre-trained generative model, scraped dataset, or other high-risk artifact; it builds on publicly released backbones, retrievers, and benchmarks under their original licenses, and only releases training and evaluation code that orchestrates these components.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Backbones (`Qwen3-VL`, `Qwen2.5`, `Qwen3-32B`), retrievers (`GME-Qwen2-VL-2B`, `all-MiniLM-L6-v2`), benchmarks (Mem-Gallery, MMLong-Bench, LoCoMo), training infrastructure (`Slime`, `Megatron-LM`, `SGLang`), and every baseline are credited via citation in Section 5 and Appendix C; each asset is used in its publicly released form under its original license without modification.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce a single new asset: the `PRR-GRPO`-trained `EvoMM` policy checkpoint built on top of the publicly released `Qwen3-VL-4B-Instruct` backbone, which we plan to release upon acceptance under the same license as the upstream backbone, together with the training and evaluation code; the checkpoint is documented end-to-end by

Appendix B (training data composition, optimizer, learning rate, KL/entropy coefficients, reward configuration, and infrastructure), Appendix E (prompts and JSON schemas defining the action space the checkpoint is trained to follow), and Appendix A (intended use, scope, and remaining limitations).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: All experiments use publicly released benchmarks under their original evaluation protocols; no crowdsourcing or research with human subjects was conducted as part of this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or any data collection requiring IRB approval; all data come from publicly released benchmarks and follow their original release terms.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: `Qwen3-VL-4B-Instruct` (Mem-Gallery, MMLongBench) and `Qwen2.5-14B-Instruct` (LoCoMo) serve as the policy, reference, and training-time judge in our agentic loop. To avoid evaluator-policy alignment in the reported scores, the test-time LLM-J metric on Mem-Gallery is additionally computed by an independent, frozen `Qwen3-32B` that is never used during training. All four models are used in their publicly released, off-the-shelf form following the inference setup specified in Appendix B, and only the policy is modified by `PRR-GRPO` fine-tuning (Section 4.2).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.