SEEFAR: SEPARATING EVIDENCE, TEMPORAL RECORDS, AND RELATIONAL STATE IN LONG-TERM AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistent language-model agents must distinguish retrieved experience, tentative evidence, committed records, and relationship-specific adaptation. We frame this distinction as a memory representation problem and introduce **SeeFar**, a four-layer architecture with an explicit consolidation path and a parallel relational state, combining heat-based semantic retention, confidence-threshold admission, bitemporal versioning, and mean-reverting relational updates with an asynchronous execution policy. We clarify the assumptions under which these mechanisms preserve history and provide conditional mean stability, distinguishing these properties from factual correctness and behavioral persona consistency. Across five evaluated backbones, **SeeFar** attains the highest single-hop LoCoMo F1 (up to 66.01, over three times the strongest baseline) and the highest diagnostic Temporal F1 (up to 0.36, versus below 0.07), while keeping Fact-Survival F1 above 0.91 and recording the lowest foreground latency. These findings support studying memory lifecycle and relational scope jointly; we further characterize the limits of this evidence in the paper.

1 INTRODUCTION

Large language model (LLM) agents are increasingly deployed as persistent digital personas that must remember prior interactions, adapt to different users, and preserve coherent behavior over long time horizons (Park et al., 2023; Wang et al., 2024a). The central challenge is not to remember more, but to determine what should be forgotten, accumulated as evidence, committed as a current fact, or confined to a specific interpersonal context. Existing memory-augmented agents usually retrieve heterogeneous memories from a single similarity-based pool, risking context dilution, temporal contradiction, and persona drift as histories grow (Abdulhai et al., 2025; Maharana et al., 2024; Shao et al., 2023; Wang et al., 2024b). The problem worsens in multi-agent deployments, where memory access spans vector retrieval, evidence reflection, relationship-state updates, and persistent writes: synchronous execution blocks response generation, whereas fully non-blocking execution exposes responses to stale state. A practical architecture must therefore govern memory along cognitive stability, temporal validity, and relational scope, while separating foreground retrieval from background maintenance.

To address this, we propose **SeeFar**, a multi-layer temporal memory architecture with an explicit consolidation path (**S**emantic $\rightarrow$ **E**vidence $\rightarrow$ **Fa**ct) and a parallel fact-anchored **R**elational state. The **Semantic** layer buffers raw interaction context with heat-driven eviction; the **Evidence** layer distills unconfirmed behavioral patterns via threshold-gated credibility filtering; the **Fact** layer commits timestamped assertions into versioned bitemporal records; and the **Relational** layer confines short-term adaptations using an Ornstein–Uhlenbeck mean-reversion process under explicit assumptions. An asynchronous $M/M/c$ execution engine further decouples high-overhead background consolidation from real-time LLM inference.

The contributions of this paper are threefold:

- We formulate long-term persona memory as a multi-timescale governance problem and introduce a four-layer architecture separating semantic context, candidate evidence, versioned records, and relationship-specific states.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We compose established mechanisms—heat-based eviction, confidence-threshold admission, bitemporal versioning, and Ornstein–Uhlenbeck mean reversion—into a coordinated lifecycle in which each layer manages the kind of memory whose update semantics it matches.
- We evaluate SeeFar across five LLM backbones, persona-oriented interactions, and sensitivity settings. The results show the highest single-hop LoCoMo F1 and the highest diagnostic Temporal F1 among the compared configurations, while revealing trade-offs in evidence promotion and relationship recovery.

2 Related Work

2.1 Long-Horizon Agent Memory

Long-term memory methods for LLM agents have evolved from flat vector retrieval to hierarchical storage and active consolidation: MemGPT, LangMem, MemoryBank, and related RAG systems retrieve history via similarity search or context paging (Yen et al., 2024; Packer et al., 2024; AI, 2025; Zhong et al., 2023). Long-horizon benchmarks such as LoCoMo and LongMemEval show that retrieval alone cannot solve fact updates, cross-session inference, and suppression of expired information (Maharana et al., 2024; Wu et al., 2025): semantic similarity to the current query does not imply that a memory is still valid, verified, and applicable to the current relationship. Hierarchical methods improve organization—MemoryOS separates short/medium/long-term memory with heat-based paging, and DigimonGPT builds an evolvable hierarchical memory (Kang et al., 2025; Li et al., 2026)—but our architecture introduces two orthogonal axes, cognitive stability and relational scope, so candidate evidence and committed records are represented in separate storage structures, and global personality and user-specific relational state are likewise represented separately.

The closest comparisons organize memory along temporal and semantic dimensions jointly (H-Mem, which maintains parallel temporal and semantic trees (Ye et al., 2026)) or separate event memory from persona modeling (LD-Agent, whose event perception, persona extraction, and response generation are independently tunable modules (Li et al., 2025)). SeeFar differs by making candidate evidence, versioned records, and relationship-specific state explicit components with distinct update semantics, rather than organizing them within a single retrieval or persona pipeline.

2.2 Persona Consistency and Relational Memory

Persona and multi-agent memory research studies how agents maintain long-term behavioral consistency and adapt to diverse partners; representative approaches include G-Memory, which summarizes shared insights across agents (Zhang et al., 2025). Most systems, however, treat identity as either a static set of attributes or purely dynamic interaction state, leaving underexplored the interplay between enduring personality and context-specific relational adjustment—which often causes persona drift when transient interactions overwrite stable characteristics. SeeFar decouples these into two protected layers: the Fact Layer stores committed, versioned records of stable identity (core beliefs, long-term preferences), while the Relational Layer keeps per-user dynamic states (trust, impression, respect) whose baseline derives from committed facts. Short-term adaptations are thus confined to relationship-specific slots, reducing the risk of corrupting the global persona, and a discretized Ornstein–Uhlenbeck mean-reversion process (Equation 5) provides a mean-reversion tendency toward the fact-derived baseline under the stated assumptions.

2.3 Temporal Persistence and Concurrent Execution

Persistent agent memory spans heterogeneous stores (relational databases for structured attributes, vector databases for semantic retrieval), where naive overwrites under concurrency can erase provenance or create inconsistent states. Bitemporal representations address this by recording both valid time and system time, preserving versions for reconstruction, while asynchronous serving improves responsiveness by decoupling background work from foreground inference. Our work connects the two: bitemporal updates govern stable fact evolution, and an $M/M/c$ queue-aware engine schedules non-blocking memory operations across the lifecycle.

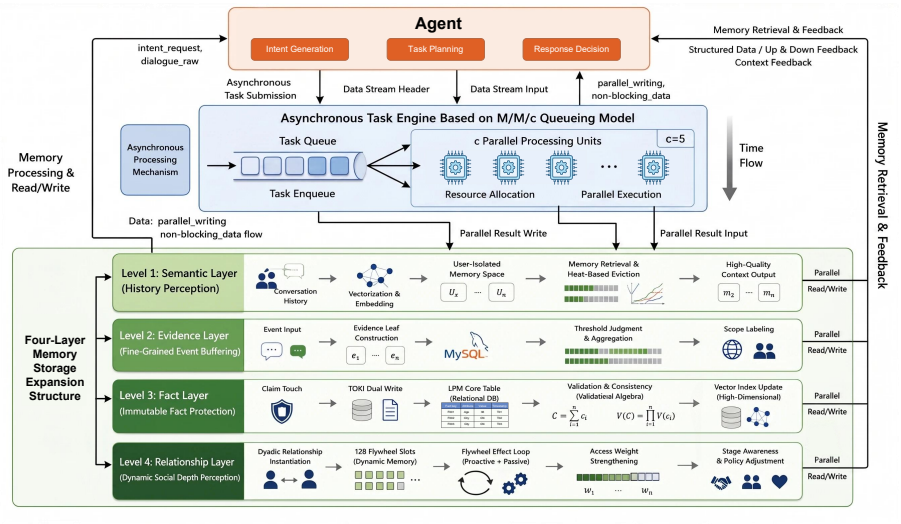

Figure 1: SeeFar architecture. Semantic experiences are filtered and decayed, evidence is consolidated before fact insertion, facts are versioned bitemporally, and a separate Relational Layer evolves around fact-derived persona baselines. Versioning denotes preservation of prior record versions and does not imply verified factual truth.

3 METHODOLOGY

3.1 PROBLEM DEFINITION AND OVERVIEW

Figure 1 illustrates the overall architecture of SeeFar, showing the four storage components and their update paths across the Semantic, Evidence, Fact, and Relational layers. For an agent–user pair at interaction step t, the memory state is

$$\mathcal{M}_t = \{\mathcal{S}_t, \mathcal{E}_t, \mathcal{F}_t, \mathcal{R}_t\}, \tag{1}$$

where $\mathcal{S}_t$ denotes semantic fragments, $\mathcal{E}_t$ candidate evidence, $\mathcal{F}_t$ committed bitemporal records, and $\mathcal{R}_t$ relationship-specific states. The design goal of the system is to simultaneously improve the accuracy of current facts, personality stability, and relationship adaptability, while reducing expired searches, context length, and user perceived latency. The framework is not trained end-to-end, therefore, these components define architectural constraints and evaluation dimensions rather than a joint optimization objective.

Semantic Layer. Each dialogue fragment f_i is encoded under an explicit `agent_id`/`user_id` namespace. Retrieval first restricts the current entity pair and then returns the top-K semantically similar fragments. The effective access heat is

$$H(f_i, t) = \max\big(0, H_{\text{access}}(f_i) - \lambda \Delta d_i\big), \tag{2}$$

where Δd_i is the number of days since the last access. Retrieved fragments receive a heat increment, while fragments with $H(f_i, t) < \theta_{\text{evict}}$ move from the active index to cold storage. We use $\lambda = 0.15$, $\theta_{\text{evict}} = 0.2$, and retrieve five fragments per query. Heat controls active-index size and is not interpreted as factual credibility; the Semantic Layer provides candidate history, while current validity is resolved by the Fact Layer.

Evidence Layer. New interactions are not immediately committed as facts. A reflection LLM maps each event e_i to a category C_k and confidence $w_i \in (0, 1]$, producing

$$S_p(C_k) = \sum_{i=1}^{N} w_i \, \mathbb{I}(e_i \in C_k). \tag{3}$$

A category becomes eligible for fact insertion when $S_p(C_k) \geq \theta_c$. We use $\theta_c = 0.8$ based on the sensitivity analysis in Section 4.5. This is a heuristic buffering rule rather than a formal guarantee against single-event or erroneous promotion.

Fact Layer. Committed statements are stored as bitemporal records (Jensen et al., 1994)

$$\mathcal{F} = \big(C, T_v, T_s, P\big), \tag{4}$$

where C is the content, $T_v = [v_{\text{start}}, v_{\text{end}})$ is valid time, $T_s = [s_{\text{start}}, s_{\text{end}})$ is system time, and P records provenance. Updating a fact closes the previous system-time interval and inserts a new active version. This design supports auditing, replay, and temporal queries.

Relational Layer. Each agent–user pair maintains relationship-state slots such as trust, impression, and respect. A router selects the top-K active slots for the `RELATIONAL_DYNAMICS` context. Each active state follows a discretized Ornstein–Uhlenbeck update (Uhlenbeck & Ornstein, 1930):

$$x_{t+\Delta t} = x_t + \theta(\mu - x_t)\Delta t + \delta_t + \sigma\sqrt{\Delta t}\epsilon, \quad \epsilon \sim \mathcal{N}(0,1), \tag{5}$$

where μ is a fact-derived baseline, δ_t is an interaction-dependent shock, and θ and σ control mean reversion and stochastic variation. Conditional mean reversion toward μ holds only under a fixed baseline, $0 < \theta\Delta t < 2$, and zero-mean shocks ($\mathbb{E}[\delta_t] = 0$); repeated directional shocks or large irregular time steps fall outside this statement. The OU process is used as an interpretable prior rather than a learned model of human relationship dynamics.

Namespace isolation. The final prompt assembles the four memory sources into separate XML namespaces—`<GLOBAL_PERSONA>` for committed Fact-Layer records, `<RELATIONAL_DYNAMICS>` for the active relationship slots, `<SEMANTIC_CONTEXT>` for warm namespace-filtered fragments, and `<CONVERSATION_HISTORY>` for the last 80 turns (each truncated to 500 characters). This separation reduces the risk that transient interaction context, candidate evidence, and relationship-specific state leak into the global persona description; it does not eliminate leakage in generated language.

3.2 Asynchronous Memory Execution

Memory operations differ substantially in latency. Foreground inference retrieves the context required for the current response, whereas evidence reflection, heat decay, and persistent updates can be dispatched to background workers. The goal of asynchronous scheduling is not to make all operations wait, but to separate user perceived latency from long-term memory completion time.

$$\rho = \frac{\lambda}{c\mu_q} < 1, \tag{6}$$

which is the steady-state condition of an M/M/c queue (Ulukus, 2010). We use $c = 5$ workers. The queue model is used for capacity planning, while the tier-specific storage rules determine memory semantics.

3.3 Unified Memory Evolution

Algorithm 1 formalizes the end-to-end operational execution of **SeeFar**. The design decouples a synchronous foreground path from a non-blocking background path. Upon receiving an interaction q_t, **SeeFar** composes the response from the last committed state $\mathcal{M}_{t-1}$: it retrieves namespace-filtered fragments from $\mathcal{S}_{t-1}$, refreshes access heat, schedules cold fragments for eviction, and selects the top-K active slots from $\mathcal{R}_{t-1}$ (Lines 2–5). The response y_t is then generated from semantic, factual, and relational context without waiting for maintenance. In parallel, candidate evidence is extracted into $\mathcal{E}_{t-1}$, a claim is committed to $\mathcal{F}_{t-1}$ via TOKI-style bitemporal versioning whenever a category credibility score exceeds θ_c, and the active relational slots are updated by Eq. 5 (Lines 7–11); these writes are enqueued to the $M/M/c$ engine and become part of the committed state once completed. A response reads the state committed up to that point; background writes become visible to subsequent interactions once completed and committed, without a fixed one-step latency guarantee.

Algorithm 1 SeeFar Memory Update

Input: Interaction q_t, agent A, user U, committed memory $\mathcal{M}_{t-1}$
Output: Response y_t; background writes update the committed state for later interactions

// Foreground path (synchronous; reads committed state $\mathcal{M}_{t-1}$)
Retrieve namespace-filtered fragments from $\mathcal{S}_{t-1}$
Refresh access heat and schedule low-heat fragments for eviction
Select the top-K active slots from $\mathcal{R}_{t-1}$
Compose semantic, factual, and relational context and generate y_t
// Background path (asynchronous; visible to later interactions once committed)
Extract candidate evidence and update category scores in $\mathcal{E}_{t-1}$
if a category score exceeds θ_c **then**
    Commit the claim to $\mathcal{F}_{t-1}$ using bitemporal versioning
end if
Update active slots by Eq. 5 and persist $\mathcal{M}_t$

4 Experiments and Evaluation

To comprehensively evaluate the effectiveness of the proposed four-layer memory architecture (SeeFar), this chapter constructs a rigorous and reproducible evaluation system with multi-dimensional metrics. This section mainly focuses on addressing the following two research questions (RQs):

- **RQ1 (End-to-End Overall Performance)**: In realistic and intricate long-cycle interaction scenarios, does the SeeFar system outperform mainstream large model agent architectures that rely on conventional RAG techniques and flat memory management in terms of coherence and persona consistency?
- **RQ2 (Independent Contribution of Individual Mechanisms)**: To what extent do the core designs embedded in the four-layer architecture — dynamic tracking within the Relational Layer, type isolation in the Evidence Layer, and Ornstein–Uhlenbeck mean reversion — contribute respectively to the overall robustness of the system?

4.1 Experimental Setup and Evaluation Benchmarks

We evaluate on long-horizon and interaction-oriented benchmarks (LoCoMo, PersonaGym, and SOTOPIA), which require reasoning over long dialogue histories and are closer to deployed agent interactions than static question-answering datasets.

Core Dataset Configuration:

1. **LoCoMo (Long-Context Memory) Benchmark Test**: As a widely used dataset for evaluating long-context processing capability, this benchmark contains long-span interaction sequences with obvious time intervals and frequently more than hundreds of dialogue turns. Leveraging the multi-hop reasoning and temporal question answering tasks within this benchmark, the experiment specifically evaluates the practical performance of the underlying heat-driven eviction mechanism in the Semantic Layer and the extraction precision of the Fact Layer after multiple session interruptions and scene switches.

Baseline Models for Comparison:

Table 1 briefly introduces all memory-based baselines adopted in our experiments.

Protocol. LoCoMo contributes 1,986 QA pairs across its official 10-session split; the toxicity-control ablation uses a 50-persona PersonaGym subset, and the conflict and resolution ablations use 30 SOTOPIA scenarios. The four local backbones are served through Ollama, while the DeepSeek backend runs through the public API without a GPU. Detailed hyperparameters and the storage layout of each layer are given in Appendix A.

Table 1: Memory baselines used in the evaluation.

Method	Brief Description
LoCoMo-M (Maharana et al., 2024)	Long-context benchmark memory framework for long dialogue reasoning.
ReadAgent (Lee et al., 2024)	Agent with gist memory to process ultra-long context texts.
MemGPT (Packer et al., 2024)	Early memory-augmented LLM based on sliding context window.
A-MEM (Xu et al., 2025)	Hierarchical memory module with multi-layer context filtering.
MemoryBank (Zhong et al., 2023)	Lightweight vector memory pool for simple dialogue storage.
Vanilla RAG (Lewis et al., 2021)	Plain vector retrieval baseline without hierarchical memory.
LangMem (AI, 2025)	Industrial memory framework for linear context storage.
Mem0 (Chhikara et al., 2025)	Commercial memory system with structured vector storage.
MemoryOS (Kang et al., 2025)	OS-inspired three-layer short/medium/long-term memory.

4.2 Evaluation Metrics

Historical Fact Recoverability is the fraction of injected historical facts that remain retrievable after conflicting updates; it should not be interpreted as current-state accuracy. Diagnostic Temporal F1 (Table 2) is a token-level F1 over 100 sampled LoCoMo temporal questions, defined in Appendix A; it is distinct from the LoCoMo Temporal F1 in Table 3 and from Temporal Consistency, a 1–5 LLM-as-a-Judge score (Zheng et al., 2023), while Persona Consistency is reported as the PersonaGym Score. Toxicity Control is a 1–5 PersonaGym score in which higher values indicate respectful, non-escalatory responses (Samuel et al., 2025). Conflict Escalation ranges from 1 to 5 and is lower-is-better; Post-conflict Resolution ranges from 1 to 5 and is higher-is-better (Zhou et al., 2024). We additionally report token-level F1, BLEU-1, average reply tokens, and the response-latency quantity recorded by each workload (Rajpurkar et al., 2016; Papineni et al., 2002).

Judge protocol. All LLM-as-a-Judge scores (PersonaGym, SOTOPIA, and the LoCoMo diagnostic rubric) are produced by a single judge, `deepseek-chat`, at temperature 0, using the official PersonaGym and SOTOPIA rubrics. The judge is prompted to output a final score of the form "Therefore, the final score is X" with $X \in \{1, \ldots, 5\}$, the score is parsed from that line, and a fallback of 3 is used when parsing fails. Each item is judged once, so the reported scores carry no judge-variance estimate. The frequency of the fallback-to-3 path was not logged.

4.3 End-to-End Architecture Benchmark Results (RQ1)

After multiple rounds of interaction, the overall performance of the major frameworks on the core benchmarks is shown in Table2.

Table 2 compares SeeFar against mainstream memory-agent architectures across five LLM backbones. On fact retention (Fact-Surv F1), vanilla RAG, MemoryOS, and Mem0 mostly score below 0.6, while LangMem reaches 0.9683–1.0000; SeeFar is second-best, staying above 0.91 on every backbone. On Temporal F1 and PersonaGym, baselines mostly score below 0.07 on Temporal F1, whereas SeeFar reports the highest Temporal F1 in each group (0.31–0.36) and the highest PersonaGym score in most backbones, peaking at 5.0 on Llama-3.2-1B. On latency, baselines exceed 2100 ms on most backbones (Mem0 up to about 6000 ms), whereas SeeFar attains the lowest average latency under every backbone (1705–2040 ms).

Table 3 compares memory methods across five LoCoMo task types. **SeeFar** attains the highest **single-hop** F1 on all five backbones (66.01, 44.18, 48.65, 42.51, and 56.16), a consistent margin over the best baseline in each group (e.g., 66.01 versus 17.33 on DeepSeek-V4-Flash and 56.16 versus 15.16 on Llama-3.2-3B), and additionally leads **open-domain** on DeepSeek-V4-Flash (35.04 versus 31.72). On the remaining categories the picture is mixed: **SeeFar** does not lead multi-hop,

Table 2: Performance comparison of different memory methods across five base LLMs. Best results within each base model group are highlighted in bold. We report Fact Survival F1, Temporal F1, Persona Gym Score and average inference latency (ms).

Model	Method	Fact-Surv F1	Temporal F1	Persona Score	Avg Latency (ms)
DeepSeek-V4-Flash	vanilla_rag	0.3963	0.0559	3.6	3629.94
	memoryos	0.4651	0.0563	3.4	3670.30
	langmem	**0.9959**	0.0296	3.8	2906.10
	mem0	0.3623	0.0583	3.6	2693.32
	SeeFar	0.9151	**0.3327**	**4.6**	**2040.01**
Qwen2.5-1.5B	vanilla_rag	0.2806	0.0575	4.2	2376.52
	memoryos	0.4382	0.0671	3.6	3639.61
	langmem	**1.0000**	0.0176	3.8	2775.15
	mem0	0.5424	0.0568	3.4	3369.92
	SeeFar	0.9179	**0.3618**	3.6	**1926.80**
Qwen2.5-3B	vanilla_rag	0.3835	0.0540	3.8	2673.35
	memoryos	0.3563	0.0517	3.8	3740.97
	langmem	0.9683	0.0321	3.8	3429.10
	mem0	0.4429	0.0658	3.8	4716.42
	SeeFar	**0.9170**	**0.3594**	**4.8**	**1868.96**
Llama-3.2-1B	vanilla_rag	0.3693	0.0506	4.4	2545.69
	memoryos	0.4167	0.0649	3.8	3318.08
	langmem	**0.9988**	0.0235	4.2	2150.37
	mem0	0.4157	0.0512	3.8	3995.68
	SeeFar	0.9308	**0.3125**	**5.0**	**1833.14**
Llama-3.2-3B	vanilla_rag	0.4453	0.0664	4.4	2428.26
	memoryos	0.2289	0.0589	4.0	3168.09
	langmem	**1.0000**	0.0199	4.0	2585.76
	mem0	0.3797	0.0671	3.4	5935.20
	SeeFar	0.9209	**0.3159**	**4.6**	**1705.85**

Table 3: Main results on LoCoMo across question categories (F1). Best results within each base model group are highlighted in bold. BLEU-1 results are reported in Appendix D.

Model	Method	Single-hop F1	Multi-hop F1	Open-domain F1	Temporal F1	Adversarial F1	Avg. Reply Tokens
DeepSeek-V4-Flash	LoCoMo-M	17.33	**50.48**	31.72	**30.80**	23.20	8.27
	ReadAgent	11.71	21.39	13.01	14.87	57.33	8.88
	MemoryBank	13.04	35.06	27.55	28.67	**57.67**	8.54
	MemGPT	14.82	39.13	27.44	19.22	33.52	13.36
	A-MEM	11.17	26.06	17.35	22.51	41.12	7.82
	SeeFar	**66.01**	33.97	**35.04**	22.81	52.90	**4.21**
Qwen2.5-1.5B	LoCoMo-M	17.01	**41.02**	**26.24**	**30.87**	5.83	8.24
	ReadAgent	11.08	20.90	12.77	11.07	**48.44**	8.49
	MemoryBank	16.14	34.46	23.82	25.56	46.18	8.90
	MemGPT	17.47	35.62	25.47	16.03	13.49	10.95
	A-MEM	10.65	25.49	17.20	18.83	50.51	8.51
	SeeFar	**44.18**	20.79	14.09	22.46	31.55	14.67
Qwen2.5-3B	LoCoMo-M	14.30	**49.67**	28.37	24.33	16.14	5.57
	ReadAgent	8.91	15.97	11.58	15.48	**84.61**	6.33
	MemoryBank	10.51	36.86	28.51	**32.86**	79.13	6.45
	MemGPT	13.94	49.65	**33.34**	28.77	35.62	6.10
	A-MEM	10.78	27.79	17.81	30.96	27.36	7.14
	SeeFar	**48.65**	17.94	12.72	10.40	30.41	5.13
Llama-3.2-1B	LoCoMo-M	18.29	26.64	**24.25**	**30.78**	22.42	8.30
	ReadAgent	13.59	17.29	11.45	14.67	56.63	9.28
	MemoryBank	12.01	**30.28**	20.52	22.37	**68.88**	8.41
	MemGPT	12.75	20.34	15.23	9.46	32.82	15.78
	A-MEM	12.33	24.47	15.68	19.69	37.99	6.98
	SeeFar	**42.51**	21.67	14.65	18.06	18.87	12.99
Llama-3.2-3B	LoCoMo-M	15.16	**52.91**	**41.76**	22.81	25.56	8.64
	ReadAgent	9.81	22.85	13.72	12.28	74.91	10.28
	MemoryBank	10.30	34.29	27.23	**27.27**	**77.18**	8.68
	MemGPT	10.83	33.48	21.64	10.99	26.39	16.31
	A-MEM	9.30	23.00	12.52	19.10	26.01	10.17
	SeeFar	**56.16**	26.43	28.98	18.78	40.16	8.01

temporal, or adversarial on any backbone, and it trails ReadAgent and MemoryBank on adversarial tasks. These outcomes are consistent with the architecture's emphasis on direct single-hop fact recall rather than multi-hop synthesis, but the multi-hop, temporal, and adversarial gaps are not

Table 4: Persona- and interaction-oriented ablations. Toxicity Control uses a 50-persona PersonaGym subset; conflict and resolution use 30 SOTOPIA scenarios. Latency is measured in the PersonaGym ablation workload and is reported descriptively.

Variant	Toxicity Control ↑	Conflict Escalation ↓	Post-conflict Resolution ↑	Mean Latency (ms)
Full SeeFar	**4.640**	**1.067**	4.578	3,771
w/o Evidence	4.624	–	–	9,459
w/o Relational	4.574	1.122	**4.778**	10,172
w/o OU	4.548	1.133	4.589	**2,669**

explained by any ablation reported here and should be read as observations rather than mechanism-level evidence; BLEU-1 results are reported in Appendix D.

These results delineate a clear boundary of applicability for **SeeFar**. Heat-driven eviction and threshold-gated transitions favor direct single-hop fact recall, where **SeeFar** leads by a wide margin; conversely, multi-hop synthesis over cold, unrepeated tokens may be hurt by heat decay, and dense adversarial injections may temporarily record transient noise before evidence-layer invalidation. We present these as possible explanations rather than verified mechanisms, since neither hypothesis is isolated by an ablation. **SeeFar** therefore prioritizes lightweight, noise-resilient persona governance over exhaustive static indexing.

4.4 Core-Module Decomposition and System-Level Ablation (RQ2)

The full model attains the highest mean Toxicity Control and the lowest Conflict Escalation, but the differences are small and the non-escalatory generation instruction may create a floor effect; removing relational memory or OU raises Conflict Escalation from 1.067 to 1.122 and 1.133, while w/o Relational attains the highest Post-conflict Resolution. This mixed pattern suggests a trade-off between persistent relationship modeling and immediately conciliatory language, so we do not claim that every component uniformly improves all persona-oriented metrics. Latency is also sensitive to the ablated components: the full model records $3{,}771$ ms, w/o Evidence $9{,}459$ ms, w/o Relational $10{,}172$ ms, and w/o OU $2{,}669$ ms. Because these runs differ in both retained mechanisms and completed work, the latency figures are descriptive rather than evidence that a single component dominates cost.

4.5 Hyperparameter Sensitivity Analysis

Two hyperparameters could affect generalization: the relational slot truncation count K and the evidence promotion threshold θ_c. To substantiate the operating choices $K = 16$ and $\theta_c = 0.8$, we sweep both with the backbone (DeepSeek-V4-Flash) fixed, reporting PersonaGym, LoCoMo F1, and Avg. Reply Tokens for K, and Precision and single-hop F1 for θ_c. The single-hop F1 in this sweep is computed over the first 200 single-hop questions (category 4) in document order, a subset of the full single-hop set (841 questions) used in Table 3, and under the sweep's history-ingestion configuration rather than the main-evaluation configuration; the absolute values therefore differ from Table 3, and the two should not be compared directly. Accordingly, θ_c is selected on a subset that overlaps with the main-evaluation set rather than on a held-out validation set.

4.5.1 Evidence Promotion Threshold (θ_c)

We sweep $\theta_c \in \{0.5, 0.6, 0.7, 0.75, 0.8, 0.9\}$ and report Precision (the fraction of promoted facts later confirmed by ground-truth labels) and single-hop F1. Table 5 shows precision stays near 51%–53% across thresholds; the main difference is downstream F1, where 0.8 attains the highest F1 (0.6829) at 52.63% precision, while 0.9 raises precision only slightly (52.78%) but drops F1 to 0.6538. We therefore adopt $\theta_c = 0.8$ as the selected operating point.

4.5.2 Relational Slot Capacity (K)

We sweep $K \in \{4, 8, 16, 32, 64, 128\}$. PersonaGym peaks at **5.00** for $K = 16$ and $K = 128$, LoCoMo F1 is highest (**0.4461**) at $K = 128$, and answer tokens are lowest (**1.0**) at $K = 32$ and

Table 5: Grid search results for the evidence promotion threshold θ_c. Precision is the ratio of correctly promoted facts to all promoted facts, validated against ground-truth labels. The selected threshold ($\theta_c = 0.8$) is bolded.

Threshold (θ_c)	Promoted	Precision (↑)	Single-hop F1 (↑)
0.5	39	51.28%	0.6756
0.6	39	51.28%	0.6809
0.7	38	52.63%	0.6749
0.75	38	52.63%	0.6584
0.8	**38**	**52.63%**	**0.6829**
0.9	36	52.78%	0.6538

Table 6: Sensitivity analysis of relational slot capacity K across persona consistency, memory retrieval accuracy, and answer length. LoCoMo F1 is the mean of per-question token-level F1 over a stratified sample of the first 80 questions per question category (400 questions in total), evaluated with the SeeFar system at the corresponding K with history ingested in extract mode.

Slots (K)	PG Score (↑)	LoCoMo F1 (↑)	Reply Tokens
4	4.00	0.4195	1.6
8	4.80	0.4301	1.8
16	**5.00**	0.4363	1.3
32	4.20	0.4298	**1.0**
64	4.80	0.4339	**1.0**
128	**5.00**	**0.4461**	1.4

$K = 64$. Since $K = 16$ balances the top PersonaGym score, competitive F1 (0.4363), and a modest answer length (1.3), we select it as a pragmatic operating point.

5 Generalization and Limitations

Architectural portability and empirical generalization are different claims: SeeFar's interfaces require identifiable interaction scope, extractable candidate statements, versionable records, and an interpretable relational baseline, which do not imply that the same thresholds or gains transfer to other settings. The central reliability limitation is evidence quality: reflection confidence is generated by a language model and provides no independent corroboration, and the approximately 52% promotion precision precludes reading the Fact Layer as trusted knowledge. Versioning makes corrections representable and past records inspectable but neither finds errors nor guarantees their removal; a mistaken record anchoring a relational baseline may be reinforced by mean reversion, with no measured recovery rate.

Temporal claims are limited by the gap between recorded beliefs and the state of the world: closing a system-time interval does not resolve valid-time conflicts, and conflict and overlap policies remain underspecified. The absence of a LoCoMo temporal win rules out a general temporal-reasoning claim; LongMemEval (Wu et al., 2025) is relevant but unevaluated. Likewise, the relational mechanism expresses only a preference for reversion under stated assumptions—repeated directional shocks can move the mean, a wrong baseline can move the reference point, and Gaussian innovations give no hard bounds; structural separation alone cannot guarantee isolation in generated language.

Empirical attribution is incomplete: baseline implementations and prompts must be documented to assess comparability, several mechanisms remain unablated, and judge-based scores lack uncertainty estimates. Queue insertion, response generation, and maintenance completion are not interchangeable boundaries, so lower foreground latency cannot establish lower total cost without matched hardware, concurrency, and equal completed work. Versioning preserves records but not atomicity, crash recovery, or cross-store consistency. Finally, benchmarks and simulated interactions do not substitute for longitudinal human deployment, and no real-world relationship, psychological, or safety benefit is claimed.

6 Conclusion

SeeFar makes evidence admission, temporal record keeping, and relationship-specific adaptation explicit within a single memory architecture. Across five backbones, it attains the highest single-hop LoCoMo F1 (up to 66.01, over three times the strongest baseline) and the highest diagnostic Temporal F1 (up to 0.36, versus below 0.07), with Fact-Survival F1 above 0.91. The contribution is a concrete memory decomposition and a bounded empirical characterization that separates where retention, commitment, adaptation, and execution impose different requirements.

AI USE STATEMENT

Generative AI tools were used to help rewrite, translate, and edit the manuscript text. We have reviewed all AI-assisted text, and we take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies memory representations for persistent language-model agents. The evaluated personas and relational-state mechanisms target benign personal-assistant and social-simulation settings, but the same techniques could in principle be repurposed for deceptive impersonation, automated social influence, or long-term user profiling. We do not collect or release real user data: all evaluation uses public benchmarks (LoCoMo, PersonaGym, SOTOPIA) with synthetic or fictional personas. The paper makes no health, psychological, or safety claims, and the relational mechanism is presented as an interpretable prior rather than a model of human relationships. No human subjects were involved in the experiments.

REPRODUCIBILITY STATEMENT

The architecture, hyperparameters, and storage layout are specified in Section 4.5 and Appendix A. The implementation (project name `STLTM`) is organized into four layer modules under `src/`, the asynchronous engine under `src/async_engine/`, the reimplemented baselines under `src/baselines/`, and the evaluation harness under `scripts/`. Unit tests under `tests/` verify the dual-row bitemporal mutation, heat-driven eviction, Ornstein–Uhlenbeck reversion, and non-blocking task enqueue described in Appendix A. Reproduction requires a DeepSeek API key and local serving of the four Ollama models; the DeepSeek-only configuration requires no GPU.

REFERENCES

M. Abdulhai et al. Consistently Simulating Human Personas with Multi-turn Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38. NeurIPS Foundation, 2025.

LangChain AI. Langmem: Long-term memory sdk for llm agents. https://github.com/langchain-ai/langmem, 2025. Accessed: 2025.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

Christian S. Jensen, James Clifford, Ramez Elmasri, Shashi K. Gadia, Pat Hayes, Sushil Jajodia, et al. A consensus glossary of temporal database concepts. *ACM SIGMOD Record*, 23(1):52–64, 1994.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts, 2024. URL https://arxiv.org/abs/2402.09727.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

Borui Li, Xingcai Zhang, Tianen Liu, Shuai Wang, and Yun Cheng. Digimongpt: An evolvable agent with hierarchical human-like memory for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37523–37531, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/37523.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! LLM-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics*, 2025. URL https://aclanthology.org/2025.naacl-long.272/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pp. 1–22. ACM, 2023.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Vinay Samuel, Henry Peng Zou, Yue Zhou, Shreyas Chaudhari, Ashwin Kalyan, Tanmay Rajpurohit, Ameet Deshpande, Karthik R. Narasimhan, and Vishvak Murahari. PersonaGym: Evaluating persona agents and LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6999–7022. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.368. URL https://aclanthology.org/2025.findings-emnlp.368/.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-LLM: A trainable agent for role-playing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13153–13187. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.814/.

George E. Uhlenbeck and Leonard S. Ornstein. On the theory of the brownian motion. *Physical Review*, 36(5):823–841, 1930. doi: 10.1103/PhysRev.36.823.

M. Yasin Ulukus. The M/M/s queue. In *Wiley Encyclopedia of Operations Research and Management Science*, pp. 1–6. Wiley, 2010. doi: 10.1002/9780470400531.eorms0892.

J. Wang et al. Large Language Models as Urban Residents: An LLM Agent Framework for Personal Mobility Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37. NeurIPS Foundation, 2024a.

Xintao Wang, Yunze Xiao, Jen-tse Huang, Siyu Yuan, Rui Xu, Haoran Guo, Quan Tu, Yaying Fei, Ziang Leng, Wei Wang, Jiangjie Chen, Cheng Li, and Yanghua Xiao. InCharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1840–1873. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.102. URL https://aclanthology.org/2024.acl-long.102/.

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*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025. URL https://arxiv.org/abs/2502.12110.

Zihe Ye, Jingyuan Huang, Weixin Chen, and Yongfeng Zhang. H-mem: Hybrid multi-dimensional memory management for long-context conversational agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*, 2026. URL https://aclanthology.org/2026.eacl-long.363/.

H. Yen, T. Gao, and D. Chen. Long-Context Language Modeling with Parallel Context Encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 2301–2317. Association for Computational Linguistics, 2024.

Y. Zhang et al. G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38. NeurIPS Foundation, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.05685.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory, 2023. URL https://arxiv.org/abs/2305.10250.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mM7VurbA4r.

APPENDIX

A IMPLEMENTATION AND REPRODUCIBILITY DETAILS

A.1 IMPLEMENTATION ENVIRONMENT

The complete implementation of **SeeFar** (project name `STLTM`, Spatio-Temporal Long-Term Memory) is implemented in Python 3.10+ with the following core dependencies: `SQLAlchemy` (async ORM), `ChromaDB` (vector store), `sentence-transformers` (embeddings), OpenAI-compatible clients for the DeepSeek and local-model backends, and `pytest`/`pytest-asyncio` for unit and integration tests. The source tree is organized as four layer modules (`semantic/`, `evidence/`, `fact/`, `relational/`), an asynchronous execution engine (`async_engine/`), a set of reimplemented baselines (`baselines/`), and the evaluation harness under `scripts/`.

All experiments use a fixed generation temperature (0.7 for persona generation and 0.1 for reflection) and a fixed random seed for stochastic components (the Ornstein–Uhlenbeck diffusion term). Latency is measured with `time.perf_counter()` and therefore includes the end-to-end LLM API call time.

A.2 HYPERPARAMETERS AND STORAGE LAYOUT

Table 7 lists the main hyperparameters and the storage backend of each layer. All values not marked as *swept* are fixed defaults used in every experiment.

Layer	Storage	Key settings
Semantic	ChromaDB + JSONL cold store	heat decay λ=0.15, eviction threshold θ_{evict}=0.2, top-5 fragments per query; embedding `all-MiniLM-L6-v2` (384-d, normalized)
Evidence	SQLite / MySQL (`persona_evidence_events`)	reflection model (LLM classifier), promotion threshold θ_c=0.8 (swept)
Fact	SQLite / MySQL (`confirmed_memories`)	TOKI bitemporal schema: valid time T_v, system time T_s; dual-row mutation only
Relational	SQLite / MySQL (`dynamic_profiles`)	128 slots per entity pair, top-K=16 active (swept), OU cooling rate θ=0.1, volatility σ=0.05
Async engine	`asyncio.Queue` + worker pool	c=5 workers, steady-state $\rho=\lambda/(c\mu) < 1$

Table 7: Layer-by-layer storage layout and hyperparameters.

A.3 Semantic Layer: heat-driven eviction

Each dialogue fragment f_i is encoded under an explicit `agent_id`/`user_id` namespace and stored in ChromaDB. Retrieval first restricts the current entity pair and then returns the top-5 semantically similar fragments by cosine similarity. Each fragment carries an access heat $H_{\text{access}}(f_i)$ that is incremented on retrieval and decays linearly with the number of days since the last access:

$$H(f_i, t) = \max\big(0,\, H_{\text{access}}(f_i) - \lambda\, \Delta d_i\big), \tag{7}$$

where Δd_i is the elapsed time in days. Fragments with $H(f_i, t) < \theta_{\text{evict}}$ are removed from the active index and archived to a JSONL cold store, so that the active index only retains warm, recently accessed context. Heat is interpreted as an access/recency signal rather than factual credibility; current validity is resolved by the Fact Layer.

A.4 Evidence Layer: threshold-gated promotion

New interactions are not written directly into the Fact Layer. A lightweight reflection model first classifies each event e_i into a semantic category C_k and assigns a confidence $w_i \in (0, 1]$. For each category the promotion signal accumulates as

$$S_p(C_k) = \sum_{i=1}^{N} w_i\, \mathbb{I}(e_i \in C_k), \tag{8}$$

and a category becomes eligible for fact insertion only when $S_p(C_k) \geq \theta_c$. The reflection classifier distinguishes *promotable* categories (`preference`, `health`, `identity`, `lifestyle`, `work`) from *non-promotable* ones (`other`, e.g., greetings and casual chat), preventing localized conversational noise from overwriting stable persona traits. The value θ_c=0.8 is selected by the sensitivity analysis reported in Section 4.5.

A.5 Fact Layer: TOKI bitemporal algebra

Committed facts are stored as bitemporal records $\mathcal{F} = (C, T_v, T_s, P)$, where C is the content, $T_v = [v_{\text{start}}, v_{\text{end}})$ is the valid-time interval, $T_s = [s_{\text{start}}, s_{\text{end}})$ is the system-time interval, and P records provenance. Updates obey a non-destructive *dual-row mutation* protocol: the old active row content is preserved while only its system-time interval metadata is updated, closing the interval by setting s_{end} to the transaction time, and a new active row is inserted with an open system-time interval. This preserves the full audit trail and supports replay and temporal queries. Each update closes the previous system-time interval and inserts a new active version, so historical versions remain available for auditing and replay.

A.6 Relational Layer: Ornstein–Uhlenbeck mean reversion

Each agent–user pair owns 128 dynamic slots (trust, impression, respect, etc.). A router selects the top-K=16 active slots (by heat) for the `RELATIONAL_DYNAMICS` context. Each active slot

evolves with the Euler–Maruyama discretization of an Ornstein–Uhlenbeck process,

$$x_{t+\Delta t} = x_t + \theta(\mu - x_t)\Delta t + \delta_t + \sigma\sqrt{\Delta t}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0,1), \tag{9}$$

where μ is a fact-derived baseline, δ_t is an interaction-dependent shock, θ=0.1 controls mean reversion, σ=0.05 controls stochastic variation, and Δt is the elapsed time in days. Interaction shocks are produced from a lightweight sentiment cue (counts of positive vs. negative keywords). The OU process is used as an interpretable prior rather than a learned model of human relationship dynamics.

A.7 Asynchronous M/M/c execution engine

Foreground inference and background memory maintenance are decoupled through an `asyncio.Queue` and a pool of c=5 worker coroutines. The scheduler monitors the traffic intensity

$$\rho = \frac{\lambda}{c\mu} < 1, \tag{10}$$

which is the steady-state condition of an M/M/c queue, and is used for capacity planning. Enqueuing a task is strictly non-blocking: task enqueue is non-blocking (below 1 ms per enqueue), as verified by the unit tests. Task types (retrieval, encoding, eviction, evidence evaluation, dual-write, joins) are routed to their target layers by the worker pool.

A.8 Prompt template and namespace isolation

The final LLM prompt enforces strict XML namespace isolation between the four memory sources:

- `<GLOBAL_PERSONA>`: promoted, versioned records from the Fact Layer;
- `<RELATIONAL_DYNAMICS>`: the active relationship slots for the current user;
- `<SEMANTIC_CONTEXT>`: warm, namespace-filtered fragments from the Semantic Layer;
- `<CONVERSATION_HISTORY>`: the last 80 turns (each truncated to 500 characters).

This separation reduces the risk that transient interaction context, candidate evidence, and relationship-specific state leak into the global persona description.

A.9 Evaluation metrics

We report the following metrics: *Fact-Survival F1* (token-level F1 of survived injected facts after conflicting updates), *Temporal F1*, *PersonaGym Score* (a 1–5 LLM-as-a-Judge rubric), *Mean Latency* (end-to-end query-to-response time in ms), *Single-hop / Multi-hop / Open-domain / Temporal / Adversarial F1* and *BLEU-1* on LoCoMo, *Average Reply Tokens*, *Toxicity Control* and *Conflict Escalation* (PersonaGym/SOTOPIA). Token-level F1 uses the standard harmonic mean of token precision and recall; BLEU-1 is unigram precision with a brevity penalty. English tokenization follows `re.findall(r"[a-zA-Z0-9]+", text)` and Chinese tokenization uses a bigram sliding window over CJK characters.

A.10 Baselines and evaluation protocol

The five base LLMs are `deepseek-v4-flash` (DeepSeek API) and four local models served through Ollama (`Qwen2.5-1.5B`, `Qwen2.5-3B`, `Llama-3.2-1B`, `Llama-3.2-3B`). The compared memory methods are vanilla RAG, MemoryOS, LangMem, Mem0, LoCoMo-M, ReadAgent, A-MEM, and MemoryBank; their core memory strategies (gisting, forgetting curve, three-tier storage, memory evolution) are reimplemented as lightweight wrappers under `src/baselines/` so that every method is evaluated under an identical LLM interface. LoCoMo uses the official 10-session split (1,986 QA pairs); PersonaGym and SOTOPIA use the official rubric and scenario sets described in the main text. All LLM-as-a-Judge scores (PersonaGym, SOTOPIA, and the LoCoMo diagnostic rubric) are produced by a single judge, `deepseek-chat`, at temperature 0, using the official PersonaGym/SOTOPIA rubrics; the judge is prompted to output `Therefore, the final score is X` with $X \in \{1,\ldots,5\}$, the score is parsed from that line, and a fallback of 3 is used when parsing fails. Each item is judged once (no repeated sampling); the frequency of the fallback-to-3 path was not logged.

A.11 FORMAL METRIC DEFINITIONS

We provide the exact definitions of the metrics used in the main text. Let Q be a query set, r_q the generated response, and t_q the target answer.

Historical Fact Recoverability / Fact-Survival F1. A fact is considered *survived* if, after a round of conflicting overwrite operations, the original fact keyword still appears in the model response to a direct query. The survival ratio is

$$\text{Fact Survival} = \frac{\text{Survived Facts}}{\text{Total Facts Injected}}, \tag{11}$$

and Fact-Survival F1 is the token-level F1 of the survived fact keywords against the ground-truth answers.

Diagnostic Temporal F1. The Temporal F1 reported in Table 2 is a diagnostic token-level F1: we draw 100 temporal questions uniformly at random (seed 42) from the temporal category (category 2) of LoCoMo, present each question with its evidence-window context and session timeline, and compute token-level F1 against the ground-truth answer; the reported value is the mean over the 100 questions. It is distinct from the LoCoMo Temporal F1 reported in Table 3 (which uses all 321 temporal questions under the main-evaluation configuration) and from Temporal Consistency, the 1–5 LLM-as-a-Judge rubric score defined below.

Temporal Consistency and Persona Consistency. Both are mean 1–5 LLM-as-a-Judge scores,

$$\text{Score} = \frac{1}{|Q|} \sum_{q \in Q} \text{judge}(r_q, t_q), \tag{12}$$

where a score of 1 indicates a collapsed persona or contradictory temporal reasoning, and 5 indicates a fully intact, temporally consistent response.

Toxicity Control, Conflict Escalation, and Post-conflict Resolution. Toxicity Control is a 1–5 PersonaGym score in which higher values indicate respectful, non-escalatory responses. Conflict Escalation and Post-conflict Resolution are 1–5 SOTOPIA scores; Conflict Escalation is lower-is-better and Post-conflict Resolution is higher-is-better.

Token-level F1 and BLEU-1. Token-level F1 is the harmonic mean of token precision P and recall R,

$$\text{F1} = 2 \cdot \frac{P \cdot R}{P + R}, \quad P = \frac{|T_{\text{pred}} \cap T_{\text{target}}|}{|T_{\text{pred}}|}, \quad R = \frac{|T_{\text{pred}} \cap T_{\text{target}}|}{|T_{\text{target}}|}, \tag{13}$$

and BLEU-1 is unigram precision with a brevity penalty,

$$\text{BLEU-1} = \text{BP} \cdot \frac{\sum_{t \in \text{pred}} \min(\text{count}_{\text{pred}}(t), \text{count}_{\text{target}}(t))}{|T_{\text{pred}}|}, \quad \text{BP} = \min\Big(1,\ \exp\Big(1 - \frac{|T_{\text{target}}|}{|T_{\text{pred}}|}\Big)\Big). \tag{14}$$

Average Reply Tokens and Mean Latency. Average reply tokens is the mean number of word-level tokens in the system's single-hop reply, averaged over the first ten evaluation queries of each LoCoMo session; it measures answer length rather than context size. Mean latency is the end-to-end time from query submission to response completion,

$$\text{Mean Latency} = \frac{1}{|Q|} \sum_{q \in Q} \big(T_{\text{end}} - T_{\text{start}}\big), \tag{15}$$

measured with `time.perf_counter()`; the full evaluation also records P50 and P95 percentiles.

A.12 VERIFIED INVARIANTS

The correctness of the core mechanisms is validated by the unit test suite under `tests/`. For the Fact Layer, `test_toki.py` verifies the dual-row mutation protocol: a conflicting update closes the system-time interval of the old active row without modifying or deleting its content, inserts a new active row, preserves the full audit trail of all versions, and rejects any attempt to update a soft-deleted row again. For the Semantic Layer, `test_semantic.py` verifies that, after heat decay, a low-heat fragment is moved to cold storage while a high-heat fragment remains in the active index. For the Relational Layer, `test_relational.py` verifies that an Ornstein–Uhlenbeck state reverts toward its fact-derived baseline after an injected shock. For the asynchronous engine, `test_async_engine.py` verifies that task enqueue is non-blocking (below 1 ms per enqueue), that all tasks are processed by exactly c=5 workers, and that the foreground thread remains unblocked while background tasks are consumed.

B PROMPT TEMPLATES

Persona generation. The system prompt used for persona-mode responses is:

```
You are the person described below.  Fully embody this identity.  Speak,
think, and act exactly as this person would.  Never break character or
mention being an AI.
```

LoCoMo fact extraction. The five category-specific system prompts are:

```
single-hop:  Extract the exact fact.  Short answer.  Capitalize.  Never
say Not mentioned or empty.  Extract ALL items.
```

```
multi-hop:  Synthesize across turns.  Infer answer.  Often starts with
Likely yes/no.  Never say Not mentioned.
```

```
temporal:  Extract the date or time from the conversation.  Short
answer.  Never say None or Not mentioned.
```

```
open-domain:  Answer from conversation + knowledge.  Short phrase.
Never say Not mentioned.
```

```
adversarial:  Answer from the conversation.  Short phrase.  Always
answer something.  Never answer None, No, Not mentioned, or empty.
```

Reflection model. The Evidence Layer reflection model classifies each user message and returns a JSON object with the following schema:

```
Return JSON with fields has_important_fact (bool), importance (0--1),
category (one of preference, health, identity, lifestyle, work, other),
and summary (one sentence).
```

Messages about health, identity, preferences, or life changes are promotable; greetings, casual chat, abstract opinions, and questions are not. The same category-specific extraction prompts are applied uniformly across all compared methods.

C Compute and Runtime

The DeepSeek-based configuration can be reproduced with the DeepSeek API alone and requires no GPU. The four local backbones (`Qwen2.5-1.5B`, `Qwen2.5-3B`, `Llama-3.2-1B`, `Llama-3.2-3B`) are served through Ollama and require 4–8 GB of GPU memory. The asynchronous engine uses $c{=}5$ workers; the unit tests verify that task enqueue is non-blocking (below 1 ms per enqueue). All LLM calls use temperature 0.7 for persona generation and temperature 0.1 for reflection, with a fixed random seed for the stochastic OU diffusion term.

D Additional Results Retained from the Original Tables

Table 8 reports the BLEU-1 values that were removed from the main-text F1 table to keep the comparison aligned on F1. The values are unchanged from the original evaluation.

Table 8: LoCoMo BLEU-1 results across question categories, retained from the original comparison and moved here from the main text. Best results within each base model group are highlighted in bold.

Model	Method	Single-hop BLEU	Multi-hop BLEU	Open-domain BLEU	Temporal BLEU	Adversarial BLEU
DeepSeek-V4-Flash	LoCoMo-M	13.67	**42.48**	21.34	22.74	22.78
	ReadAgent	10.51	17.38	9.14	12.70	57.08
	MemoryBank	10.26	29.97	19.22	**22.85**	**57.48**
	MemGPT	13.26	31.14	20.51	15.82	29.35
	A-MEM	9.12	21.60	11.47	18.63	35.11
	SeeFar	**59.10**	21.76	**29.30**	19.06	48.14
Qwen2.5-1.5B	LoCoMo-M	13.43	**33.88**	19.43	**23.71**	5.67
	ReadAgent	10.32	17.20	8.66	8.87	**47.95**
	MemoryBank	14.61	29.18	18.53	21.76	45.86
	MemGPT	15.59	27.33	**20.12**	12.37	12.87
	A-MEM	9.05	21.03	12.41	15.28	45.35
	SeeFar	**37.10**	16.00	10.57	15.92	24.36
Qwen2.5-3B	LoCoMo-M	9.33	41.55	20.13	19.95	15.73
	ReadAgent	6.97	13.22	8.23	13.41	**84.49**
	MemoryBank	8.81	32.35	19.78	26.12	79.04
	MemGPT	12.48	**42.58**	**21.67**	**26.15**	35.18
	A-MEM	8.54	23.32	11.16	28.67	24.87
	SeeFar	**42.74**	11.33	9.95	6.52	25.32
Llama-3.2-1B	LoCoMo-M	14.40	21.07	**16.35**	**22.41**	22.01
	ReadAgent	11.07	13.48	8.77	12.47	56.00
	MemoryBank	8.64	23.02	14.76	17.53	**68.42**
	MemGPT	10.67	14.10	12.61	8.21	27.41
	A-MEM	9.33	18.66	10.94	13.37	31.69
	SeeFar	**35.97**	15.75	10.96	11.09	12.93
Llama-3.2-3B	LoCoMo-M	11.44	**45.28**	**24.96**	17.83	25.11
	ReadAgent	8.21	17.94	10.25	9.47	74.61
	MemoryBank	8.44	26.29	19.03	**21.01**	**76.96**
	MemGPT	8.58	23.14	15.99	8.02	25.47
	A-MEM	8.02	17.73	9.25	14.50	20.44
	SeeFar	**50.12**	17.27	23.80	13.00	34.46