OrthoMem: Orthogonal Three-View Memory Architecture for LLM-Based Conversational Agents

Anonymous ACL submission

Abstract

Persistent memory is essential for LLM-based conversational agents to maintain coherent, personalized interactions across sessions. Existing memory systems either store flat atomic facts that fail at cross-session reasoning, or adopt hierarchical architectures where higher layers are merely compressed summaries of lower layers—introducing semantic redundancy rather than complementary information. We propose **OrthoMem**, an orthogonal three-view memory architecture where each layer provides a structurally distinct but evidence-linked perspective: (1) L0 Atomic Facts for verbatim factual precision, (2) L1 Entity Index for cross-session per-attribute entity profiles, and (3) L2 Higher-Order Insights for temporal evolution chains and cross-entity reasoning patterns. To ensure faithfulness, we introduce **Cross-Layer Source Tracing**: every L1 and L2 entry maintains explicit provenance links back to its source L0 facts, enabling simultaneous access to higher-order aggregations and their underlying evidence. For efficient retrieval, we propose **Layer-Aware Query Decomposition**, which infers query intent and routes to optimal layer combinations via a single LLM call, with cross-layer redundancy suppression to maximize context information density. Experiments on LoCoMo and LongMemEval benchmarks demonstrate that OrthoMem consistently outperforms all memory-based baselines and achieves competitive performance with Long-Context at 12× lower token cost.

1 Introduction

Large language models (LLMs) have achieved remarkable single-turn capabilities, yet deploying them as persistent conversational agents remains constrained by the absence of effective long-term memory (Park et al., 2023; Packer et al., 2023). Multi-session dialogues spanning weeks or months demand factual recall, entity-level synthesis, temporal reasoning, and cross-session inference (Maharana et al., 2024; Wu et al., 2025)—diverse needs that no single memory representation can optimally serve.

Existing memory systems for LLM agents fall into two paradigms, each with structural limitations. *Flat memory* (Chhikara et al., 2025; Liu et al., 2026; LangChain Team, 2025) extracts atomic facts into vector stores and retrieves via similarity search. While effective for single-hop recall, it struggles with questions requiring cross-session entity aggregation ("What activities has Alice done over the past year?") or temporal reasoning ("How has Bob's career evolved?"), as relevant facts are scattered across sessions without structural links (Chhikara et al., 2025). *Multi-layer* approaches (Fang et al., 2026; Packer et al., 2023; Jimenez Gutierrez et al., 2024) address this by introducing hierarchical structures, but higher layers are information-decreasing compressions of lower layers (e.g., session summaries → topic summaries), creating semantic redundancy rather than complementary signals. Meanwhile, multi-structure methods (Lu et al., 2026; Yan et al., 2026) maintain diverse representations but treat them as alternatives to *select among* per query, rather than complementary views to *compose*—leaving cross-layer synergy unexploited.

We identify the core issue: existing systems lack *structural orthogonality*—the property that each memory layer captures a unique perspective that other layers cannot provide. We argue that effective memory should organize information by orthogonal perspectives rather than decreasing granularity: verbatim facts, entity-level aggregations, and relational patterns serve fundamentally different retrieval needs and should coexist as complementary views. This principle is grounded in Fuzzy-Trace Theory (FTT) (Brainerd and Reyna, 2002; Reyna, 2012), which establishes that humans encode parallel, functionally inde-

pendent verbatim and gist traces serving distinct retrieval functions. We extend FTT's dual-trace to a triple-trace formulation—*verbatim trace*, *entity gist*, and *relational gist*—motivated by the observation that "Alice likes fitness" (entity aggregation) and "Alice's interest evolved from gym to yoga to running" (temporal inference) require structurally different memory representations.

A key challenge in multi-layer memory is *faithfulness*: higher-order representations risk hallucinating or distorting information from the original conversations. Inspired by the Transformation Hypothesis (Winocur and Moscovitch, 2011)—which establishes that schematized memories coexist with rather than replace original traces—we introduce **Cross-Layer Source Tracing**: every L1 and L2 entry maintains explicit provenance links back to its source L0 facts. This ensures that higher-order aggregations always remain verifiable against ground-truth evidence, and enables the retrieval system to simultaneously provide both structured insights and their supporting details.

We propose **OrthoMem**, an orthogonal three-view memory architecture: 1) **L0 Atomic Facts**—verbatim factual extraction serving as high-fidelity ground truth; 2) **L1 Entity Index**—per-attribute structured profiles aggregating facts about the same entity across sessions; 3) **L2 Higher-Order Insights**—temporal evolution chains and cross-entity reasoning patterns capturing relationships *between* facts that are invisible when retrieved individually. For efficient retrieval, we introduce **Layer-Aware Query Decomposition**, which infers query intent and routes to optimal layer combinations via a single LLM call. A cross-layer redundancy suppression mechanism ensures that composed multi-layer contexts remain compact and non-redundant.

Contributions. 1) We propose **OrthoMem**, an orthogonal three-view memory architecture where each layer provides a schema-level complementary perspective rather than progressively compressed summaries. 2) We introduce **Cross-Layer Source Tracing** that maintains provenance links from higher-order representations back to atomic facts, and **Layer-Aware Query Decomposition** with two-stage redundancy elimination for efficient multi-layer retrieval via a single LLM call. 3) Experiments on LoCoMo and LongMemEval across three backbone LLMs show consistent improvements over seven baselines with 12× token reduction compared to Long-Context.

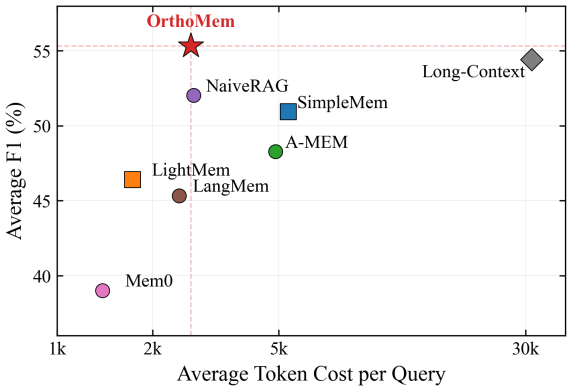

Figure 1: Efficiency–effectiveness tradeoff on LoCoMo (GPT-4.1-mini). OrthoMem achieves the highest F1 (55.3%) with low token cost (∼2.6K), outperforming Long-Context (54.4%, 31K tokens) and all memory-based baselines.

2 Related Work

Full-Context and Retrieval-Based Approaches. Full-context injection concatenates all prior dialogue into the prompt but scales poorly and suffers from middle-context degradation (Liu et al., 2024). ReadAgent (Lee et al., 2024) compresses conversations into gist tokens with on-demand page lookup. SimpleMem (Liu et al., 2026) applies semantic compression with a three-stage pipeline (extraction, online synthesis, intent-aware retrieval). These methods operate on a single representation level without distinguishing the structural roles of different memory types.

Hierarchical Memory. LightMem (Fang et al., 2026) implements sensory→short-term→long-term memory with sleep-time consolidation. MemGPT (Packer et al., 2023) adopts OS-inspired main/archival storage. Think-in-Memory (Liu et al., 2023) introduces hierarchical recall phases. MemoryBank (Zhong et al., 2024) applies forgetting curves, and Chronos (Sen et al., 2026) introduces temporal SVO indexing with dual calendars. These share a critical property: higher layers are information-decreasing compressions of lower layers, creating semantic redundancy. OrthoMem's layers are defined by structural function and jointly composed rather than hierarchically compressed.

Multi-Structure and Adaptive Approaches. FluxMem (Lu et al., 2026) maintains three struc-

tures (linear, graph, hierarchical) with a learned selector. AdaMem (Yan et al., 2026) keeps four memory types with adaptive per-query selection. Memori (Borro et al., 2026) combines semantic triples and conversation summaries. All-Mem (Lv et al., 2026) proposes non-destructive topology evolution with version links. These treat multiple structures as alternatives to *select among*—the selector chooses one structure per query, leaving cross-structure synergy unexploited. OrthoMem layers are designed for joint retrieval: a single query simultaneously draws from atomic facts, entity profiles, and relational insights.

Graph-Based and Agentic Memory. GAAMA (Paul et al., 2026) constructs concept-mediated knowledge graphs with Personalized PageRank. HippoRAG (Jimenez Gutierrez et al., 2024) implements hippocampal-inspired pattern separation. A-Mem (Xu et al., 2025) enables Zettelkasten-style dynamic linking; D-MEM (Song and Xin, 2026) introduces dopamine-inspired RPE routing for admission control; Mem0 (Chhikara et al., 2025) provides graph-enhanced consolidation; MemMA (Lin et al., 2026) coordinates the full memory cycle through multi-agent reasoning. These focus on *when and how* to update or retrieve memory, orthogonal to our contribution on *how to structurally organize* memory for complementary retrieval.

Cognitive Foundations. Prior systems invoke cognitive theory primarily as analogy: Atkinson-Shiffrin multi-store (Atkinson and Shiffrin, 1968), Tulving's episodic/semantic distinction (Tulving, 1972, 1985), and complementary learning systems (McClelland et al., 1995). OrthoMem formalizes two theories as architectural constraints: Fuzzy-Trace Theory (Brainerd and Reyna, 2002; Reyna, 2012) motivates inter-layer orthogonality (parallel independent traces serving distinct retrieval functions), and the Transformation Hypothesis (Winocur and Moscovitch, 2011) motivates non-destructive construction with source tracing.

3 Method

OrthoMem operates in two phases: an offline *memory construction* phase that builds the three-layer memory store from conversation history, and an online *retrieval and generation* phase that answers queries using layer-aware decomposition. Figure 2 provides an overview.

3.1 L0: Atomic Fact Extraction

L0 provides the high-fidelity factual foundation. We process dialogues using a sliding window (20 turns, 5-turn overlap) and extract structured facts via LLM prompting (full prompt in Appendix F). Each entry e_0 contains an id, a *lossless fact restatement*, semantic type ($\in$ {event, preference, attitude, plan, ...}), mentioned persons, entities, and timestamp. The lossless restatement performs two key normalizations: (1) **coreference resolution**—all ambiguous pronouns are resolved to explicit entity names so that each entry is self-contained without requiring surrounding context; (2) **temporal anchoring**—relative time expressions (e.g., "last Tuesday", "two weeks ago") are converted to absolute ISO-8601 timestamps using the conversation's bracketed time as reference. These ensure that every L0 entry can be independently retrieved and interpreted without access to the original dialogue. Entries are embedded and stored with keyword metadata for hybrid retrieval.

3.2 L1: Entity Index Construction

L1 provides a cross-session aggregated view structurally impossible for any single L0 entry to offer. Construction proceeds in three steps: (1) **Grouping**: L0 entries are grouped by person (with alias normalization) then by entity, producing (person, entity) buckets. (2) **Pattern-Aware Synthesis**: for each bucket with $\geq$ 2 facts, an LLM generates a structured summary detecting the dominant pattern type (count, period, change, or profile). (3) **Source Tracing**: each L1 entry stores `source_l0_ids` linking back to contributing L0 entries, actively used during retrieval to fetch original evidence.

Each L1 entry e_1 stores an id, the synthesized summary, pattern type, person, entity, and source L0 ids for tracing. L1 is strictly single-entity, single-attribute: each entry aggregates facts within one (person, entity) bucket and captures the current state of that attribute (possibly with its most recent change). Quality filters suppress trivial entries (single-fact buckets, semantically identical summaries). Full construction prompts are in Appendix F.

3.3 L2: Higher-Order Insight Extraction

L2 captures inferential insights emerging from *connections between* L0 facts that are invisible

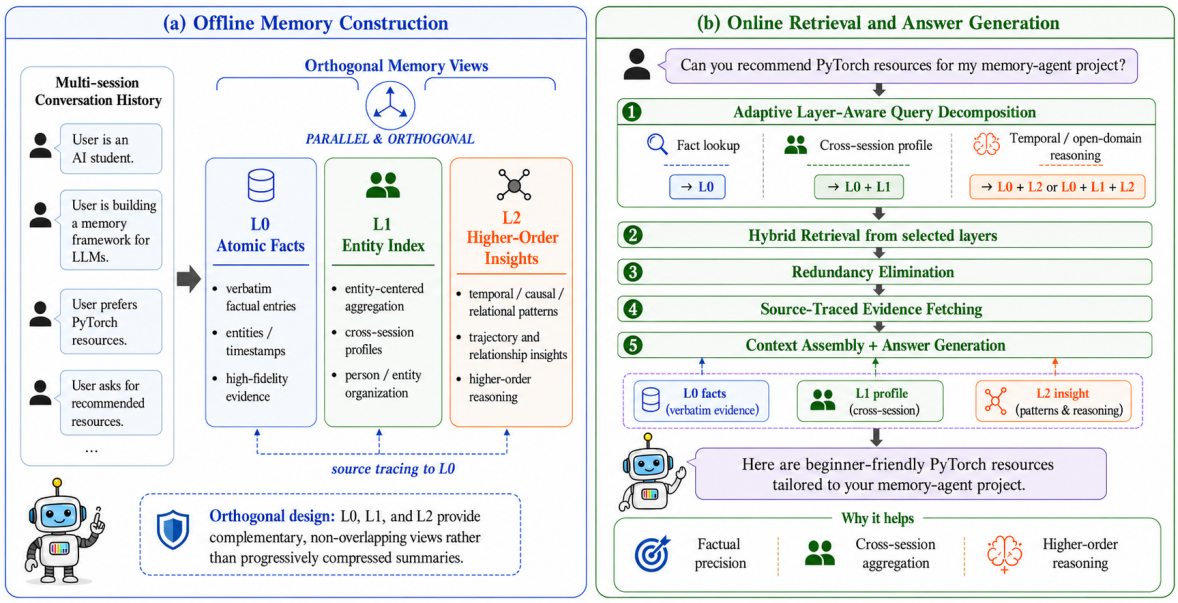

Figure 2: Overview of the OrthoMem architecture. **Left**: Memory construction builds three orthogonal views from multi-session dialogues—L0 (atomic facts), L1 (entity index with cross-session aggregation), and L2 (higher-order insights including trajectories, relationships, and patterns). L1/L2 maintain explicit source tracing links back to L0. **Right**: At query time, layer-aware query decomposition routes the question to optimal layer combinations, followed by two-stage redundancy elimination (reverse cross-layer deduplication and semantic gate) and cross-layer source tracing to assemble a high-density context for answer generation.

from any individual fact or any single L1 entry. While L1 answers "what is X's current state on attribute Y?", L2 answers "how did X evolve over time?" or "what is the relationship between X and Z?"—questions requiring multi-step reasoning across multiple (person, entity) buckets or across entities. We identify candidates via topic clustering (single-entity clusters spanning ≥ 2 sessions for trajectory/pattern insights; cross-entity clusters for relationship/contrast insights). An LLM verifies each candidate, classifying it into four types: *trajectory* (temporal shift), *relationship* (interaction dynamic), *pattern* (recurring tendency), or *contrast* (cross-entity difference), rejecting spurious co-occurrences. Each accepted L2 entry stores `evidence_ids` linking to source L0 facts. Full prompt in Appendix F.

3.4 Layer-Aware Query Decomposition and Retrieval

At retrieval time, a single LLM call decomposes the query into layer-specific sub-queries and decides which layers to activate (full prompt in Appendix F):

$$D(q) = \{\text{L0} : [q_0^1, ..., q_0^k],\ \text{L1} : [q_1^1, ...],\ \text{L2} : [q_2^1, ...]\} \quad (1)$$

Empty lists indicate the layer is skipped. For each activated layer, we perform hybrid retrieval (semantic + BM25) with layer-specific top-k budgets.

Two-Stage Redundancy Elimination. Our orthogonality is defined at the *structural* level: each layer encodes a distinct representational function (verbatim facts, entity aggregation, temporal/causal chains) with complementary schemas and independent generation logic. However, since all layers ultimately derive from the same underlying conversations, partial *semantic* overlap may arise at retrieval time when a query simultaneously activates related entries across layers. This is not an architectural deficiency but a standard multi-source retrieval challenge. We address it with two lightweight filtering stages: (1) **Reverse Cross-Layer Deduplication**: L1/L2 entries whose maximum cosine similarity to retrieved L0 facts exceeds τ_{dedup} are removed (information already available at higher fidelity). (2) **Semantic Gate**: for remaining L2 entries, sentence-level filtering removes individual sentences redundant with L0, retaining only genuinely novel inferential content (formal definition in Appendix E.4).

Cross-Layer Source Tracing. For retained L1/L2 entries, their `source_l0_ids` are used

to fetch corresponding L0 entries not already retrieved, providing both structured aggregations and granular evidence. The final context follows the order: L0 facts → source-traced L0 → L1 profiles → L2 insights.

Fault Tolerance. The pipeline degrades gracefully under routing errors. L0 is activated for *every* query regardless of decomposition decisions—it serves as the complete factual ground truth and provides a guaranteed recall floor. L1 and L2 are purely *additive*: even if the router incorrectly activates a layer, the Semantic Gate filters out entries that do not contribute novel information beyond what L0 already provides. Conversely, if the router fails to activate a beneficial layer, the system reverts to L0-only retrieval (which already achieves competitive performance, as shown in our ablation). This design ensures that routing errors produce graceful degradation (at most reverting to L0 baseline) rather than catastrophic failure.

3.5 Answer Generation and Incremental Update

The answer generator receives the assembled multi-layer context along with the original question and conversation history, producing a natural language answer (full prompt in Appendix F).

In deployment, conversations arrive session by session. OrthoMem supports incremental updates: L0 is append-only; L1 re-synthesizes only the (person, entity) buckets affected by new facts; L2 triggers re-evaluation only when a topic cluster crosses the generation threshold. The overall complexity is $O(|\text{new_facts}| \times |\text{affected_entities}|)$ rather than $O(|\text{total_history}|)$.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate on two established long-term memory benchmarks: 1) **LoCoMo** (Maharana et al., 2024): 10 multi-session conversations (19–32 sessions each, 369–689 turns) with 1,540 human-annotated questions spanning four categories—SingleHop, MultiHop, Temporal, and OpenDomain—evaluated with token-level F1 and BLEU-1. 2) **LongMemEval** (Wu et al., 2025): 500 questions across six categories (Single-Session-User, Single-Session-Assistant, Single-Session-Preference, Multi-Session, Temporal-Reasoning, Knowledge-Update) over 115 conversations, evaluated with LLM-judge accuracy following the official evaluation script. Due to the higher evaluation cost of LongMemEval, we compare against the strongest baselines (SimpleMem, LightMem, and Long-Context) rather than the full set used for LoCoMo.

Baselines. We compare against seven representative baselines spanning different memory paradigms: **NaiveRAG** (retrieval-augmented generation over raw conversation chunks); **Long-Context** (full conversation history fed directly to the LLM, representing the upper bound of context utilization); **LangMem** (LangChain Team, 2025) (semantic memory extraction with LangChain-based consolidation); **Mem0** (Chhikara et al., 2025) (production-grade flat memory with dynamic extraction and hybrid retrieval); **A-MEM** (Xu et al., 2025) (agentic memory with Zettelkasten-style dynamic linking); **LightMem** (Fang et al., 2026) (three-stage hierarchical memory with sleep-time offline consolidation); and **SimpleMem** (Liu et al., 2026) (flat memory with semantic lossless compression, extracting atomic facts via sliding-window processing and retrieving via hybrid semantic-keyword search, representing the "L0-only" paradigm).

Implementation Details. To evaluate robustness across model capabilities, we test all methods with three backbone LLMs: GPT-4.1-mini, GPT-3.5-turbo, and DeepSeek-V3. All methods use the same backbone for both memory construction and answer generation within each setting. Embeddings use text-embedding-v4 (dimension 1024) for all methods. For OrthoMem, L0 extraction uses a 20-turn sliding window with 5-turn overlap. L1 requires ≥ 2 facts per (person, entity) bucket. L2 requires ≥ 3 facts spanning ≥ 2 sessions. Retrieval uses top-5 semantic + top-5 keyword per sub-query for L0, and top-3 for L1/L2. The cross-layer deduplication threshold is 0.7. Answer generation uses temperature 0 for deterministic output; query decomposition uses temperature 0.3.

4.2 Main Results

LoCoMo. OrthoMem achieves 55.33% average F1 with GPT-4.1-mini, outperforming all baselines including Long-Context (54.42%) at 12× lower token cost (2.6K vs. 31K per query). The largest gains over memory-based baselines appear on Temporal (+12.33 points over SimpleMem) and OpenDomain (+3.7% over Long-

Table 1: Results on the LoCoMo benchmark. F1 and BLEU-1 scores (%) are reported per category, with Average computed as sample-weighted mean across all 1,540 questions. Tokens denotes the average total token cost per question during inference. **Bold** indicates best, underline second best within each model group.

Model	Method	MultiHop		Temporal		OpenDomain		SingleHop		Average		Tokens
		F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	
GPT-4.1-mini	Long-Context	**45.04**	**36.81**	59.89	46.35	21.35	17.00	59.26	50.27	54.42	44.92	31,374
	NaiveRAG	40.47	31.94	59.62	46.76	17.10	13.69	57.02	49.27	52.04	43.36	2,693
	LangMem	39.05	32.01	45.90	36.81	22.37	17.72	49.85	42.44	45.34	37.81	2,422
	Mem0	31.02	23.21	49.37	36.72	16.88	12.73	40.25	34.37	39.01	31.47	1,389
	A-MEM	35.05	27.28	56.27	43.70	17.02	13.30	53.22	45.42	48.27	39.74	4,880
	LightMem	36.49	26.65	53.10	41.20	21.20	17.29	50.06	43.93	46.41	38.54	1,731
	SimpleMem	38.69	31.47	49.48	38.40	22.95	18.69	58.76	52.73	50.92	43.73	5,355
	OrthoMem	41.63	33.05	**61.81**	**47.50**	**25.04**	**21.01**	**60.91**	**53.02**	**55.33**	**46.22**	2,638
GPT-3.5-turbo	Long-Context	**42.62**	**34.52**	59.13	45.03	21.96	17.69	59.50	50.40	54.00	44.33	32,083
	NaiveRAG	41.65	33.43	60.47	47.29	16.22	12.29	56.10	48.03	51.88	42.98	2,757
	LangMem	37.21	30.04	35.96	28.62	19.51	16.19	47.50	40.34	41.47	34.51	2,276
	Mem0	29.74	22.93	48.75	36.10	16.99	12.77	40.62	35.03	38.85	31.65	1,397
	A-MEM	36.04	28.78	55.47	43.00	15.40	12.27	53.08	45.67	48.11	39.94	4,952
	LightMem	38.18	29.22	53.82	42.01	**26.85**	**22.62**	52.11	45.33	48.34	40.27	1,750
	SimpleMem	40.64	33.24	52.37	40.87	21.95	18.55	59.13	**52.64**	52.02	44.51	5,325
	OrthoMem	41.46	33.57	**62.84**	**48.29**	23.95	20.03	**59.88**	52.08	**54.89**	**45.90**	2,622
DeepSeek-V3	Long-Context	**40.70**	30.41	57.73	43.49	**23.37**	**19.98**	**60.94**	**53.92**	**54.22**	**45.32**	32,083
	NaiveRAG	39.62	**30.89**	54.13	41.10	13.99	10.90	55.06	47.55	49.48	40.87	2,757
	LangMem	35.29	27.06	34.21	25.26	21.44	17.41	40.72	34.75	37.16	30.28	10,404
	Mem0	25.45	18.75	41.71	31.44	14.65	10.59	36.60	30.68	34.26	27.40	1,841
	A-MEM	33.97	26.64	52.43	39.34	16.50	12.89	51.95	44.32	46.54	38.09	2,063
	LightMem	30.47	20.73	42.03	27.70	17.87	13.75	44.91	34.24	39.98	29.12	1,384
	SimpleMem	37.57	29.24	53.78	39.60	19.88	17.62	47.54	42.56	45.29	37.95	5,768
	OrthoMem	40.37	30.05	**60.28**	**46.29**	23.19	18.31	57.74	50.79	52.94	44.03	2,742

Context) questions—precisely the categories requiring cross-session aggregation (L1) or temporal chain reasoning (L2) that no single-layer system can provide. OrthoMem outperforms SimpleMem by +4.4% F1 while using 2× fewer tokens, and LightMem by +8.9% F1, confirming that structural organization outperforms both brute-force retrieval volume and compression-based hierarchies. Results are consistent across models: with GPT-3.5-turbo, OrthoMem maintains the best F1 (54.89%); with DeepSeek-V3, Long-Context gains a slight edge (+1.3%) leveraging strong native long-context capabilities, but at 12× the cost—OrthoMem still dominates all memory-based baselines and leads on Temporal questions.

LongMemEval. OrthoMem achieves 77.8% accuracy, outperforming SimpleMem (+8.4%) and LightMem (+3.2%). The most revealing finding: Long-Context collapses to 47.2% despite feeding ∼114K tokens of full history. This exposes a fundamental limitation—when questions require cross-session synthesis (e.g., tracking entity changes or identifying multi-session patterns), raw dialogue overwhelms the model with irrelevant context. Structured memory that pre-organizes information by entity, attribute, and temporal dimension proves essential.

OrthoMem's advantage is most pronounced on *knowledge-update* (+8.9% over SimpleMem, +30.8% over Long-Context), where L2 temporal chains explicitly track state changes, and *single-session-assistant* (+41.1% over LightMem), where LightMem's compression discards details that OrthoMem preserves verbatim in L0. This demonstrates the complementary value of maintaining both high-fidelity facts (L0) and higher-order abstractions (L1/L2).

4.3 Ablation Study

To understand the contribution of each component, we conduct ablation experiments on LoCoMo.

Table 3 presents the results. We highlight several key findings:

Memory construction is the primary contributor. Removing L1 or L2 individually causes −3.4 and −2.5 F1 points respectively (GPT-4.1-mini), while removing both results in −4.2 points—greater than either alone, confirming that the two layers provide complementary rather than

Table 2: Results on LongMemEval benchmark. **Bold** indicates best, underline second best within each model group. OrthoMem achieves the best overall accuracy in both settings.

Model	Method	Temporal	Multi-Session	Knowledge-Update	Single-Session-User	Single-Session-Assistant	Single-Session-Preference	Average
GPT-4.1-mini	Long-Context	39.85%	36.84%	52.56%	65.71%	78.57%	10.00%	47.2%
	SimpleMem	69.17%	47.37%	74.36%	91.43%	**85.71%**	73.33%	69.4%
	LightMem	75.19%	**72.93%**	78.21%	**95.71%**	41.07%	**83.33%**	74.6%
	OrthoMem	**76.69%**	68.42%	**83.33%**	94.29%	82.14%	63.33%	**77.8%**
GPT-3.5-turbo	Long-Context	43.61%	37.59%	47.44%	68.57%	78.57%	20.00%	48.6%
	SimpleMem	66.92%	45.86%	73.08%	87.14%	**87.50%**	76.67%	68.0%
	LightMem	**79.70%**	**63.90%**	64.10%	91.43%	46.43%	**80.00%**	71.0%
	OrthoMem	73.68%	58.65%	**85.90%**	**97.14%**	**87.50%**	70.00%	**76.2%**

Table 3: Ablation study on LoCoMo (Average F1/BLEU-1, %). Each row removes one component. The orthogonal design is critical—replacing it with conventional hierarchical summaries causes the largest degradation (−12% F1).

Configuration	GPT-4.1-mini		DeepSeek-V3	
	F1	BLEU	F1	BLEU
OrthoMem (full)	**55.3**	**46.2**	**52.9**	**44.0**
Memory Construction				
w/o L2 (no temporal chains)	52.8	44.1	49.6	40.2
w/o L1 (no entity index)	51.9	43.5	49.8	41.3
w/o L1 & L2 (L0 only)	51.1	42.9	49.1	39.7
Retrieval & Deduplication				
w/o Source Tracing	52.4	43.8	50.3	42.0
w/o Cross-Layer Dedup	53.2	44.5	51.0	42.4
w/o Semantic Gate	52.6	43.9	50.4	41.6
Architecture Variant				
Conventional L1 (summary hierarchy)	43.3	36.1	41.8	34.7

redundant information. L1 (entity gist) contributes slightly more than L2 (temporal chains) on GPT-4.1-mini, though both are essential. On DeepSeek-V3, the pattern is similar (−3.3/−3.1/−3.8 points), demonstrating architecture-agnostic benefit.

Retrieval components provide meaningful but smaller gains. Source Tracing (−2.9% F1) ensures that higher-layer entries carry provenance back to L0, preventing hallucinated gist from propagating unchecked. Cross-Layer Deduplication (−2.1%) removes semantically overlapping evidence across layers, improving context density. The Semantic Gate (−2.7%) filters low-relevance candidates before final assembly. These three components collectively improve precision without adding retrieval volume.

Orthogonal design is critical. The most striking result is the "Conventional L1" variant, which replaces our orthogonal entity gist with a standard hierarchical summary (i.e., L1 summarizes L0). This causes substantial degradation: −12.0% F1 on GPT-4.1-mini (55.3→43.3) and −11.1% on DeepSeek-V3 (52.9→41.8)—the largest single-component drop in the entire ablation. The failure mode is clear: when L1 compresses L0 rather than providing a structurally distinct view, retrieval returns redundant paraphrases that crowd out useful evidence. This validates our core hypothesis that inter-layer orthogonality—not just multi-layer existence—is the key design principle.

4.4 Layer Routing Analysis

To validate that Layer-Aware Query Decomposition effectively routes different questions to appropriate layer combinations, we analyze the actual retrieval patterns across LoCoMo questions (GPT-4.1-mini).

Router Decision Patterns. Figure 3(a) shows that the query decomposition module produces category-sensitive routing decisions. L0-only retrieval dominates across all categories (71% for SingleHop, 69% for MultiHop, 81% for Temporal, 44% for OpenDomain), reflecting that most questions can be primarily addressed with verbatim facts. However, OpenDomain questions have the highest rate of engaging additional layers (56% use L1 or L2), confirming that broadly-scoped questions require diverse evidence types.

Layer Ablation by Category. Figure 3(b) shows F1 drops when restricting to different layer combinations. OpenDomain suffers the most from layer removal (−8.8% with L0 only), consistent with their need for diverse evidence types. Temporal and MultiHop show −6.1% and −5.8% drops

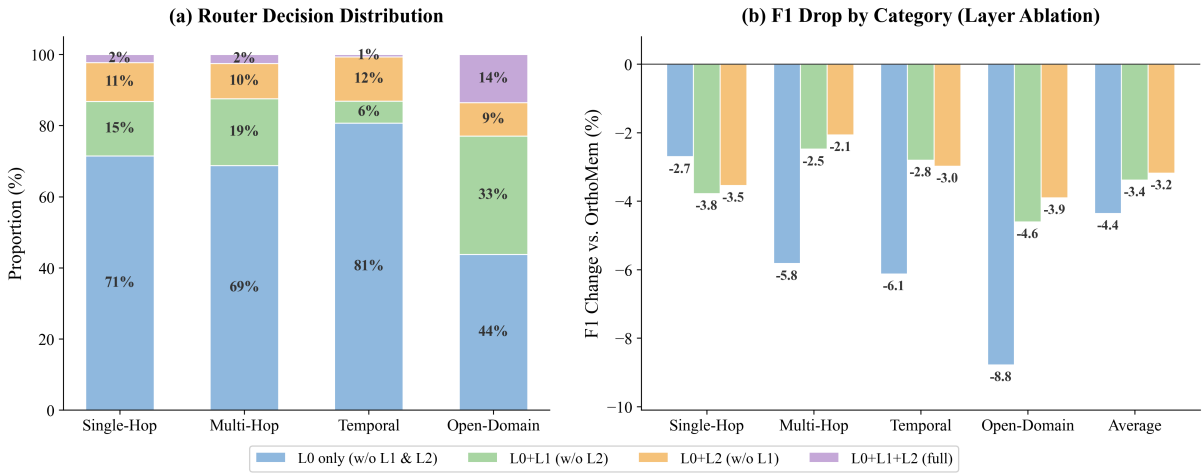

Figure 3: Layer routing and ablation analysis on LoCoMo (GPT-4.1-mini). (a) Router decision distribution by question category. (b) F1 drop by question category under layer ablation.

respectively, confirming that L2 temporal chains and L1 entity profiles are essential for cross-session reasoning. On average, removing both L1 and L2 causes −4.4% F1, while each individual layer contributes −3.4% (L1) and −3.2% (L2), confirming their complementary rather than redundant roles.

4.5 Efficiency Analysis

Token efficiency. As shown in the "Tokens" column of Table 1, OrthoMem consumes ∼2,638 tokens per query on average—comparable to NaiveRAG (2,693) and significantly lower than SimpleMem (5,355), A-MEM (4,880), and Long-Context (31,374). Despite using roughly half the tokens of SimpleMem, OrthoMem achieves +4.41% higher F1 (GPT-4.1-mini), demonstrating that structured orthogonal organization yields higher information density per token than volume-based flat retrieval.

Retrieval-time LLM calls. OrthoMem requires exactly **one** LLM call at retrieval time (query decomposition), after which all retrieval and deduplication operate via embedding similarity without further LLM involvement. In contrast, SimpleMem's reflect-and-re-retrieve mechanism requires 2–3 LLM calls per query on average (up to 4+ for complex multi-hop questions). This single-call design makes OrthoMem's retrieval latency predictable and substantially lower.

Build-time overhead. OrthoMem's primary cost is offline memory construction: L1 entity gist and L2 temporal chains require additional LLM calls during the memory building phase (∼1.8× over L0-only extraction). However, this cost is amortized across all subsequent queries and can be performed asynchronously during idle periods, making it negligible in practical deployment scenarios.

5 Conclusion

We presented OrthoMem, an orthogonal three-view memory architecture that maintains three co-existing traces—verbatim facts (L0), entity gist (L1), and relational gist (L2)—each providing a structurally distinct, complementary perspective rather than progressively compressed summaries. Combined with Cross-Layer Source Tracing and Layer-Aware Query Decomposition, OrthoMem achieves the best average performance among memory-based methods on LoCoMo and LongMemEval while maintaining 12× token efficiency over Long-Context. Our key insight is that effective multi-layer memory requires not *how much* to compress, but *what unique view* each layer should provide—an orthogonality principle we believe can guide future memory architecture design.

Limitations

While we evaluate across three backbone LLMs (GPT-4.1-mini, GPT-3.5-turbo, DeepSeek-V3), OrthoMem does not surpass Long-Context on DeepSeek-V3, suggesting that as models' native long-context capabilities improve, the relative advantage of structured memory may narrow for the strongest models—though OrthoMem still achieves this with 12× fewer tokens. The L1 and

L2 construction requires additional LLM calls during the offline phase (1.8× build cost over flat memory), which may limit applicability in extremely resource-constrained settings. The current system relies on LLM-based query decomposition, which may occasionally misroute queries to inappropriate layers; however, the fault-tolerant design (§3.4) ensures graceful degradation rather than catastrophic failure, and future work could explore learned routing mechanisms for further robustness.

OrthoMem currently does not implement memory forgetting or capacity-bounded pruning. In long-running deployments, the memory store grows monotonically. However, our orthogonal architecture is uniquely well-suited for differential retention policies: FTT predicts that verbatim traces decay faster than gist traces in human memory (Brainerd and Reyna, 2002), suggesting a natural extension where L0 entries are subject to recency- or access-frequency-based decay while L1/L2 representations—which already capture the essential semantic content—are retained indefinitely. Unlike hierarchical systems where higher layers depend on lower layers for meaning, our orthogonal design allows L1/L2 to remain fully functional even after their source L0 entries are pruned, directly mirroring the cognitive pattern of gist persistence beyond verbatim forgetting. We leave the design and evaluation of such differential retention mechanisms to future work.

Additionally, our evaluation focuses on English-language benchmarks; cross-lingual generalization remains untested. Extension to additional memory views (e.g., spatial, emotional) is a promising direction.

References

Richard C. Atkinson and Richard M. Shiffrin. 1968. Human memory: A proposed system and its control processes. In *Psychology of Learning and Motivation*, volume 2, pages 89–195. Academic Press.

Luiz C. Borro, Luiz A. B. Macarini, Gordon Tindall, Michael Montero, and Adam B. Struck. 2026. Memori: A persistent memory layer for efficient, context-aware LLM agents. *arXiv preprint arXiv:2603.19935*.

Charles J. Brainerd and Valerie F. Reyna. 2002. Fuzzy-trace theory and false memory. *Current Directions in Psychological Science*, 11(5):164–169.

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

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. 2026. LightMem: Lightweight and efficient memory-augmented generation. In *Proceedings of the International Conference on Learning Representations*.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 59532–59569.

LangChain Team. 2025. LangMem: Long-term memory for language agents. https://github.com/langchain-ai/langmem. Accessed 2026-05-26.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the International Conference on Learning Representations*.

Minhua Lin, Zhiwei Zhang, Hanqing Lu, Hui Liu, Xianfeng Tang, Qi He, Xiang Zhang, and Suhang Wang. 2026. MemMA: Coordinating the memory cycle through multi-agent reasoning and in-situ self-evolution. *arXiv preprint arXiv:2603.18718*.

Jiaqi Liu, Yaofeng Su, Peng Xia, Yiyang Zhou, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. 2026. SimpleMem: Efficient lifelong memory for LLM agents. *arXiv preprint arXiv:2601.02553*.

Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. 2023. Think-in-memory: Recalling and post-thinking enable LLMs with long-term memory. *arXiv preprint arXiv:2311.08719*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Mingfei Lu, Mengjia Wu, Feng Liu, Jiawei Xu, Weikai Li, Haoyang Wang, Zhengdong Hu, Ying Ding, Yizhou Sun, Jie Lu, and Yi Zhang. 2026. Choosing how to remember: Adaptive memory structures for LLM agents. *arXiv preprint arXiv:2602.14038*.

Can Lv, Heng Chang, Yuchen Guo, Shengyu Tao, and Shiji Zhou. 2026. All-Mem: Agentic lifelong memory via dynamic topology evolution. *arXiv preprint arXiv:2603.19595*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. 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 13851–13869, Bangkok, Thailand. Association for Computational Linguistics.

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22. Association for Computing Machinery.

Swarna Kamal Paul, Shubhendu Sharma, and Nitin Sareen. 2026. GAAMA: Graph augmented associative memory for agents. *arXiv preprint arXiv:2603.27910*.

Valerie F. Reyna. 2012. A new intuitionism: Meaning, memory, and development in fuzzy-trace theory. *Judgment and Decision Making*, 7(3):332–359.

Sahil Sen, Elias Lumer, Anmol Gulati, and Vamse Kumar Subbiah. 2026. Chronos: Temporal-aware conversational agents with structured event retrieval for long-term memory. *arXiv preprint arXiv:2603.16862*.

Yuru Song and Qi Xin. 2026. D-MEM: Dopamine-gated agentic memory via reward prediction error routing. *arXiv preprint arXiv:2603.14597*.

Endel Tulving. 1972. Episodic and semantic memory. In Endel Tulving and Wayne Donaldson, editors, *Organization of Memory*, pages 381–403. Academic Press, New York.

Endel Tulving. 1985. Memory and consciousness. *Canadian Psychology*, 26(1):1–12.

Gordon Winocur and Morris Moscovitch. 2011. Memory transformation and systems consolidation. *Journal of the International Neuropsychological Society*, 17(5):766–780.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *Proceedings of the International Conference on Learning Representations*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-Mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Shannan Yan, Jingchen Ni, Leqi Zheng, Jiajun Zhang, Peixi Wu, Dacheng Yin, Jing Lyu, Chun Yuan, and Fengyun Rao. 2026. AdaMem: Adaptive user-centric memory for long-horizon dialogue agents. *arXiv preprint arXiv:2603.16496*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Hyperparameter Sensitivity

To efficiently validate parameter choices and verify robustness, all hyperparameter sensitivity experiments in this section are conducted on a 5-conversation subset of LoCoMo with GPT-4.1-mini.

Semantic Gate Threshold. Figure 4 analyzes the effect of the cross-layer semantic gate threshold τ_{gate} across three settings. When $\tau_{\text{gate}} = 0.0$ (all L1/L2 entries suppressed, effectively L0-only retrieval), the system uses 14% fewer tokens per query (710 vs. 826) but all categories degrade compared to the default. As τ_{gate} increases from 0.0 to 0.3 to 0.75, F1 improves monotonically across all categories: Temporal shows the largest gain (60.6→63.3→65.8), followed by MultiHop (42.0→42.9→43.5) and SingleHop (59.1→59.8→60.4). Average F1 increases consistently (53.6→54.5→55.2). The results confirm that L1/L2 entries provide complementary information beyond L0 across all question types, with the strongest contribution to temporal and cross-session reasoning. The default $\tau_{\text{gate}} = 0.75$ achieves the best overall performance with moderate token overhead (+16%).

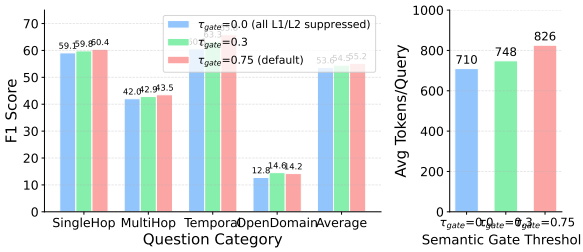

Figure 4: Effect of semantic gate threshold τ_{gate} on LoCoMo F1 and token cost (GPT-4.1-mini). Left: per-category F1 comparison. Right: average context tokens per query.

Window Size. Figure 5 shows the effect of L0 extraction sliding window size on LoCoMo performance. Average F1 decreases monotonically from 53.64 (window=10) to 51.40 (window=40), with the largest sensitivity on OpenDomain (−4.5 points) and Temporal (−2.9 points from peak at window=20). MultiHop shows a non-monotonic pattern, peaking at window=30 (44.14) where broader context aids cross-turn reasoning. The results suggest that smaller windows preserve factual granularity, while larger windows risk diluting per-turn details during extraction.

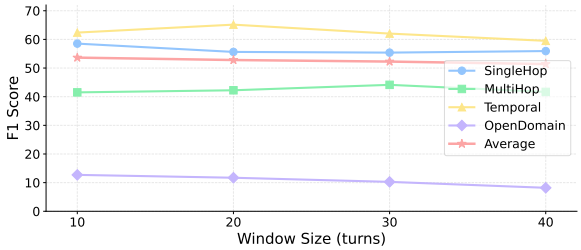

Figure 5: Effect of L0 extraction window size on LoCoMo F1 (GPT-4.1-mini). Smaller windows generally improve factual precision, with the strongest effect on OpenDomain and Temporal categories.

B Memory Construction Statistics

Table 4 reports the memory construction statistics across the 10 LoCoMo conversations. OrthoMem produces approximately 1.0 L0 entry per dialogue turn, confirming near-lossless factual extraction. L1 and L2 represent highly condensed higher-order representations: on average, one L1 entity profile is synthesized from every ∼29 L0 facts, and one L2 insight from every ∼17 L0 facts. Despite their small count, L1 and L2 entries are substantially longer than L0 (76 and 42 words vs. 25 words), reflecting their aggregative nature.

The higher layers serve as compact, high-density representations: L1 entries average 3× the length of L0 entries (aggregating multiple facts into a single profile), while L2 entries distill cross-session patterns into concise inferential statements.

Table 4: Memory construction statistics on LoCoMo (10 conversations, GPT-4.1-mini backbone).

Metric	Total	Per Conv (avg)
Sessions	272	27.2 ± 4.5
Dialogue turns	5,882	588.2 ± 112.0
L0 Atomic Facts	5,910	591.0 ± 111.9
L1 Entity Profiles	≥204	≥20.4
L2 Higher-Order Insights	≥353	≥35.3
L0 entries per turn	1.00	
L0 → L1 ratio	1 L1 per 29.0 L0	
L0 → L2 ratio	1 L2 per 16.7 L0	
Avg entry length (L0)	∼25 words	
Avg entry length (L1)	∼76 words	
Avg entry length (L2)	∼42 words	
Avg retrieved entries/query	30.8 (L0: 28.3, L1: 0.9, L2: 1.7)	

C Case Study

To illustrate how OrthoMem's multi-layer architecture contributes to correct answers—and

where it falls short—we present five representative cases from the LoCoMo benchmark (GPT-4.1-mini backbone).

C.1 Success Cases

(a) Temporal: L2 Trajectory Enables Time Reasoning. The question "How long has it been since Andrew adopted his first pet, as of November 2023?" requires synthesizing a timeline from facts scattered across multiple sessions.

- **OrthoMem** → "4 months" (Correct). L0 retrieves individual mentions: no pets (March 2023), searching (May), new adoption (October), another dog (November). L2 provides a trajectory insight that explicitly models Andrew's adoption journey from "actively searching" through "first adoption (∼July 2023)" to "second dog (October 2023)." This temporal chain allows the LLM to compute the ∼4-month gap from July to November.
- **SimpleMem** → "about one month" (Wrong). Without temporal synthesis, SimpleMem only retrieves the most recent adoption mention (October 2023) and computes the gap to November, missing the earlier first adoption.
- **LightMem** → "Since July 2023" (Wrong). LightMem identifies the approximate adoption time but fails to compute the duration, returning a date rather than a time span.

Why OrthoMem succeeds: L2's trajectory chain provides an explicit temporal narrative that connects the scattered adoption events into a coherent timeline, enabling accurate duration calculation.

(b) Multi-Hop: L1 Entity Aggregation. The question "What type of volunteering have John and Maria both done?" requires identifying a shared activity across two people's histories.

- **OrthoMem** → "volunteering at a homeless shelter" (Correct). L1 provides John's entity profile, which aggregates his community service activities including "went to homeless shelter to give out food and supplies." L2 provides parallel trajectory insights for both John and Maria converging on homeless shelter volunteering. L0 supplies supporting evidence with specific dates and quotes.
- **SimpleMem** → "none" (Wrong). The relevant facts about John's shelter involvement are distributed across sessions and never co-occur with Maria's mentions in any single retrieval window, causing SimpleMem to miss the connection entirely.
- **LightMem** → "Shelter volunteering" (Partial, F1=0.57). LightMem captures the general activity but its hierarchical summary loses the specificity of "homeless" shelter.

Why OrthoMem succeeds: L1's per-entity attribute bucketing pre-aggregates each person's volunteering activities independently, making cross-entity comparison straightforward during retrieval. L2's trajectory insights further confirm the shared pattern.

(c) Open-Domain: Three-Layer Implicit Reasoning. The question "Did Calvin and Dave have a meeting in Boston between August and November 2023?" requires reasoning about the *absence* of an event.

- **OrthoMem** → "No" (Correct). L0 establishes Calvin's Boston timeline (visits in April–May and November, but not August–October). L1's entity profile shows their September interaction was a remote conversation. L2's relationship insights reveal that Dave "consistently seeks visual evidence of Calvin's experiences remotely" and "offers congratulations via chat"—patterns indicating distance rather than co-location.
- **SimpleMem** → "Yes" (Wrong). Retrieves multiple Boston-related mentions scattered across sessions and incorrectly infers a meeting occurred.
- **LightMem** → "Yes" (Wrong). Similarly misled by the presence of Boston mentions without temporal precision to distinguish April/May visits from the August–November window.

Why OrthoMem succeeds: The combination of L0's timestamped facts (establishing *when* Calvin was in Boston), L1's entity context (showing remote interactions), and L2's relationship patterns (confirming habitual distance communication) enables the LLM to reason about event absence—a capability that requires cross-layer evidence synthesis.

Figure 6: Success cases demonstrating multi-layer collaboration. Each case shows the question, retrieved content from each memory layer, and the answers produced by OrthoMem, SimpleMem, and LightMem. (a) L2 trajectory chains enable temporal reasoning by tracking adoption timeline. (b) L1 entity profiles aggregate volunteering activities across sessions for both entities. (c) L2 relationship insights reveal that interactions are consistently remote, enabling correct "No" inference.

C.2 Failure Cases

(a) Context Noise Overwhelms Correct Signal. For "What subject have Caroline and Melanie both painted?" (gold: "Sunsets"):

- **OrthoMem** → "flowers" (Wrong)
- **SimpleMem** → "flowers" (Wrong)
- **LightMem** → "Flowers" (Wrong)

Critically, OrthoMem's retrieved context *does* contain the correct evidence: "Melanie painted a lake sunrise last year" and "Caroline painted the recent sunset painting after visiting the beach." However, the context also contains 5+ mentions of flower-related painting ("drawing flowers is one of her favorites," "the flowers are so pretty"), which dominate the signal. The LLM generator, processing 38 L0 entries, gravitates toward the higher-frequency "flowers" topic. **Root cause:**

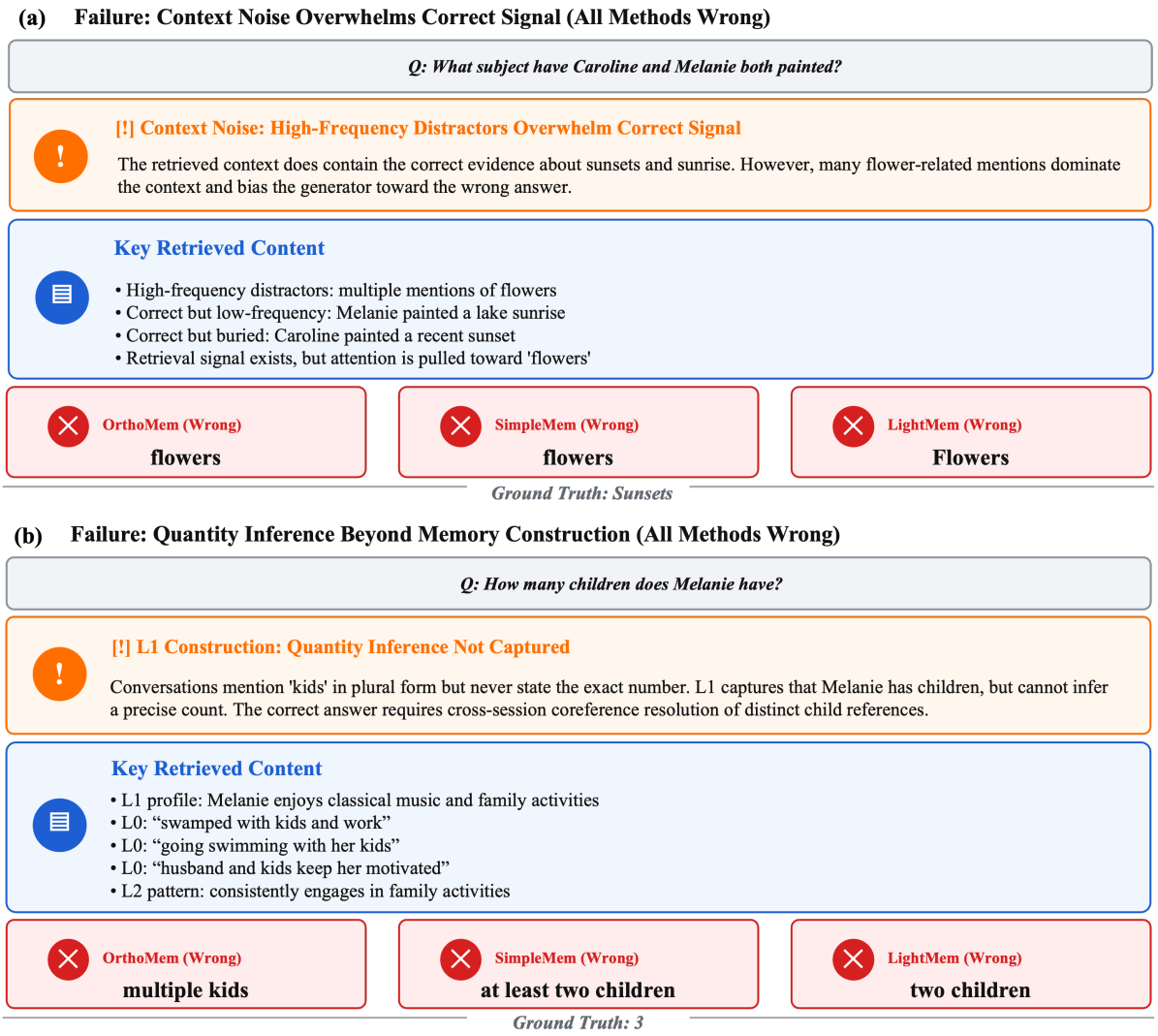

Figure 7: Failure cases revealing system limitations. (a) Retrieved context contains the correct information but high-frequency distractors mislead generation. (b) L1 entity profiling fails to track precise numerical attributes when only vague references exist.

The Semantic Gate deduplicates at the entry level but does not re-rank by query relevance; high-frequency topics create an attention bias. **Potential fix:** Add a relevance re-ranking step after deduplication that weights entries by query similarity before feeding to the generator.

(b) L1 Fails to Track Precise Quantities. For "How many children does Melanie have?" (gold: "3"):

- **OrthoMem** → "multiple kids" (Wrong)
- **SimpleMem** → "at least two children" (Wrong)
- **LightMem** → "Melanie has two children" (Wrong)

The conversations mention "kids" and "children" in plural form multiple times but never state the exact number. The evidence is scattered: museum visit with "the kids," going swimming "with her kids," "husband and kids who keep her motivated." L1's entity profile captures that Melanie has children but does not aggregate a precise count—because no single fact states the number explicitly. The ground truth (3) likely comes from counting distinct child references across sessions, which requires cross-session coreference resolution. **Root cause:** L1 construction extracts attributes from explicit statements but does not perform counting-based inference across facts. L2's pattern synthesis tracks activity patterns rather than entity counts. **Potential fix:** Add a count-aware attribute extraction step in L1 that identifies

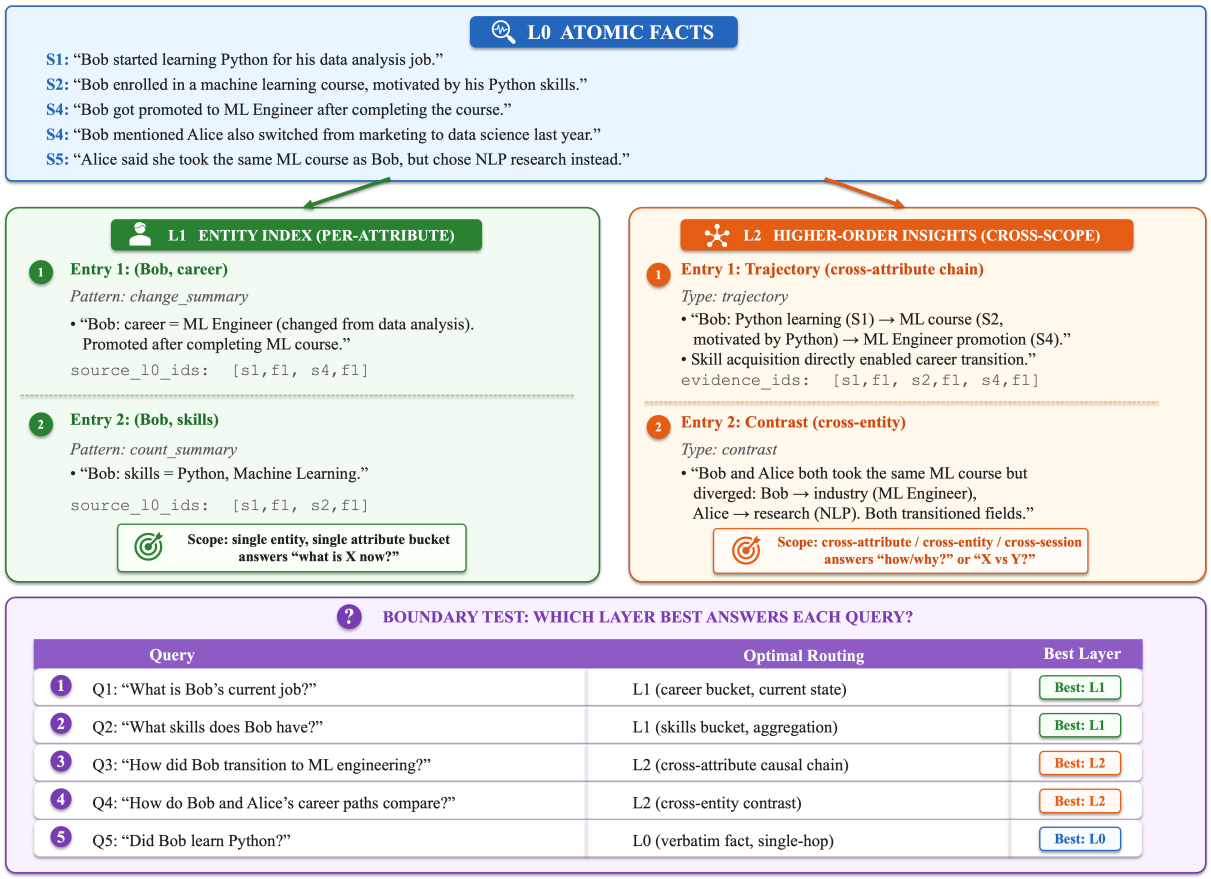

Figure 8: L1 vs. L2 boundary illustration. Given the same L0 facts, L1 produces per-attribute current-state summaries (single entity, single bucket), while L2 produces cross-attribute trajectories and cross-entity contrasts. The boundary test (bottom) shows which layer optimally serves each query type.

and resolves distinct entity references (e.g., distinct children mentioned by name or context).

D L1 vs. L2 Boundary Illustration

Figure 8 illustrates the structural boundary between L1 and L2 using a concrete example. Given the same set of L0 facts about Bob's career development (spanning Sessions 1–5), L1 produces *per-attribute snapshots* (current career state, current skill set), while L2 produces *cross-scope insights* (a trajectory chain linking skill acquisition to career transition, and a cross-entity contrast between Bob and Alice's divergent paths).

The boundary rule is precise: L1 operates within a single (person, entity) bucket and answers "what is X now?"; L2 must cross bucket or entity boundaries and answers "how/why did X evolve?" or "how do X and Y compare?". The bottom panel of Figure 8 provides a boundary test showing which layer optimally serves each query type.

E Implementation Details

E.1 L0: Atomic Fact Extraction

Conversation sessions are processed with a 20-turn sliding window (5-turn overlap). Each window is passed to an LLM extraction prompt (Appendix F) that performs entropy-aware filtering and context normalization, converting raw dialogue into compact, context-independent memory units.

E.2 L1: Entity Index Construction

Two-Level Grouping. L0 entries are grouped by `person` (with embedding-based alias normalization, threshold $\tau_{\text{norm}} = 0.85$, to merge variants like "Bob"/"Bob Smith"), then by `entity` field within each person, producing (person, entity) buckets.

Pattern-Aware Synthesis. For each bucket with ≥ 2 facts, an LLM generates a single structured summary (Appendix F) that detects the dominant pattern among four types: 1) *count summary*—

multiple distinct events of the same kind, enumerating each with specifics; 2) *period summary*—recurring habits, stating frequency and time span; 3) *change summary*—temporal evolution, stating before/after explicitly; 4) *profile fact*—stable attribute confirmed by multiple sources.

Quality Filters. Single-fact buckets are suppressed (would merely duplicate L0). Entries semantically identical to their sole source fact are removed. For large buckets (>50 facts), temporal uniform sampling ensures coverage.

E.3 L2: Higher-Order Insight Extraction

Candidate Generation. We identify potential insight candidates via two strategies: 1) *Single-entity candidates*: for each person, cluster their L0 facts by topic using embedding similarity (cosine threshold 0.6); topic groups spanning ≥ 2 sessions with ≥ 3 facts become trajectory/pattern candidates. 2) *Cross-entity candidates*: cluster all L0 facts by topic regardless of person; groups containing ≥ 2 distinct persons become relationship/contrast candidates.

Insight Verification. Each candidate is evaluated by an LLM (Appendix F) that classifies it into one of four types: 1) *trajectory*—qualitative shift in state/attitude across time; 2) *relationship*—stable interaction dynamic between two people; 3) *pattern*—recurring behavioral tendency across sessions; 4) *contrast*—meaningful difference between people on the same topic. The LLM can reject candidates (`is_meaningful=false`) if the facts merely co-occur without supporting a genuine inference. Each accepted L2 entry stores `evidence_ids` linking to source L0 facts.

E.4 Semantic Gate

For L1 entries, the entry is suppressed if its maximum cosine similarity to any retrieved L0 embedding exceeds τ_{gate}. For L2 entries (multi-sentence), sentence-level gating is applied:

$$\text{keep}(s) = \mathbb{1}\big[\text{sim}(s) < \tau_{\text{gate}}\big], \quad (2)$$
$$\text{where} \quad \text{sim}(s) = \max_{e \in \mathcal{E}_0} \cos(\mathbf{v}_s, \mathbf{v}_e).$$

If the retained text falls below ratio γ of the original length, the entire L2 entry is discarded. Default thresholds: $\tau_{\text{dedup}} = 0.7$, $\tau_{\text{gate}} = 0.75$, $\gamma = 0.3$.

F System Prompts

L0 Atomic Fact Extraction Prompt

[System] You are a professional information extraction assistant, skilled at extracting structured, unambiguous information from conversations. You must output valid JSON format.

[User] Extract every piece of valuable information from the dialogues below into structured memory entries. One entry per distinct fact; prefer over-extraction. Each entry must be fully self-contained — no ambiguous pronouns; always name the subject explicitly.
Field rules:
• `lossless_restatement`: restate the fact in a self-contained sentence. Never rewrite, round, or generalize any specific detail — all named entities, exact values, and key descriptions must appear verbatim.
• `keywords`: terms that make this entry uniquely findable — use the most specific identifiers present (proper names, exact values), not generic labels.
• `entities`: every named entity that appears in this entry.
After extracting, ask: if someone re-read only these entries (not the original dialogue), would they recover every meaningful fact completely and accurately? If not, add or correct entries.
Convert bracketed timestamps to ISO 8601. Resolve relative time expressions against the bracketed timestamp.
{`dialogue_text`}
Return ONLY the JSON array.

L1 Entity Index Construction Prompt

[System] You synthesise a small set of related facts into one rich, evidence-backed summary. The summary should be as long as necessary to preserve every specific detail (names, dates, counts, amounts, places, outcomes) that could answer a future question — never shorten at the cost of losing a detail. Do NOT invent facts. Output valid JSON only.

[User] Below are all the facts about "{person_name}" that involve "{entity_name}".
[Facts] {numbered_facts}
[Task]
Step 1 — Deduplicate: treat facts that say the same thing as one (count them once).
Step 2 — Synthesise: write a summary that covers everything a reader would need to answer any question about {person_name}'s relationship with {entity_name}, without consulting the original facts. Include every specific name, date, count, amount, place, and outcome.
[How to decide what to write]
The dominant pattern across the (deduplicated) facts should shape the summary:
• Multiple distinct events of the same kind → state total count and enumerate each with specifics
• A recurring habit or rhythm → state the pattern, frequency, and time span observed
• Something that changed over time → state the before/after explicitly, with dates
• A single stable fact → state it precisely
[Output format] Choose type from: `count_summary | period_summary | change_summary | profile_fact`
`{"summary": {"type": "<type>", "domain": "<domain>", "key": "<identifier>", "value": "<the summary>", "source_l0_ids": [...], "confidence": <0--1>}}`
Return ONLY JSON.

L2 Higher-Order Insight Extraction Prompt

[System] You synthesise scattered facts into a high-level insight that cannot be read from any single fact alone — a trajectory of change, a cross-person relationship dynamic, a behavioural pattern, or a meaningful contrast. You do NOT restate or list what happened; you draw a conclusion FROM what happened. Output valid JSON only.

[User] Facts about "{entity_name}" related to "{topic}" (chronological):
{numbered_facts}
Your task: decide whether these facts, taken together, support a high-level insight that goes beyond what any single fact states.
Choose the best-fitting insight type — or return `is_meaningful=false`:
• **trajectory**: genuine qualitative shift across time.
  Format: "X moved from [state-A] to [state-B], driven by [key events]."
• **relationship**: stable interaction dynamic between two people spanning multiple sessions.
  Format: "[person-A] consistently [role-A] while [person-B] consistently [role-B]."
• **pattern**: recurring behavioural tendency visible across sessions.
  Format: "[Person] tends to [behaviour] whenever [trigger/context]."
• **contrast**: meaningfully different experiences/attitudes on the same topic.
  Format: "While [A] [X], [B] [Y], revealing [implication]."
Return `is_meaningful=false` if: facts merely repeat the same sentiment without change; the only connection is co-occurrence; you cannot state a conclusion more specific than "they both like X".
{`"is_meaningful": true/false, "insight_type": "trajectory|relationship|pattern|contrast", "conclusion": "one specific inferential sentence", "evidence_ids": [...], "confidence": 0--1`}

Layer-Aware Query Decomposition Prompt

[System] You are a memory retrieval router. Answer in two steps, then output the final JSON.
Memory Layers:
• **L0 FACTS** — Individual events, statements, facts from single conversation turns. Best for: specific facts, single dates, quotes.
• **L1 ENTITY** — Per-entity attribute profiles aggregating facts across ALL sessions. Best for: personality/preference inference, cross-session traits.
• **L2 TEMPORAL** — Narrative arcs that reveal HOW or WHY facts connect across time or across people. Best for: gradual progressions, causal chains, attitude reversals, cross-person parallels. NOT for: counting events or single-event questions.
Step 1 — Analyze information needs (1–2 sentences): What specific information is required? Does it need isolated facts (→L0), aggregated traits (→L1), or a chain of multiple events (→L2)?
Step 2 — Generate layer queries:
• L0 (always required): 2–4 short noun phrases, entity-anchored, 3–8 words each.
• L1 (only if entity-aggregation needed): 1–2 "{entity} {specific_domain}" queries.
• L2 (only if narrative arc needed): 1–2 "{entity} {topic} evolution/journey/cause" queries.
Leave empty [] for layers not needed.
Output ONLY valid JSON: {`"l0": [...], "l1": [...], "l2": [...]`}

[User] Question: {query}

Answer Generation Prompt

[System] You are a memory-based Q&A assistant. Output valid JSON only.

[User] Answer the question using ONLY the context below.
Question: {query}
Context:
{context_str}
[Context organized as: L0 Facts → Source-Traced L0 → L1 Entity Profiles → L2 Insights]
Rules:
1. Carefully analyze the context, think through the reasoning process.
2. Provide a very CONCISE answer: prioritize the most precise and shortest expression from the context (shortest correct phrase only, verbatim from context, no elaboration).
3. Only natural language dates ("21 June 2023", not ISO). Calculate actual absolute dates based on memory timestamps.
Output JSON only:
`{"reasoning": "step-by-step justification", "answer": "concise answer"}`