From Retrieval to Reconstruction: Constructing Evolvable Cognitive Memory for Long-term Dialogue

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are evolving into long-term personal companions, necessitating memory systems that go beyond simple text retrieval. However, existing Retrieval-Augmented Generation (RAG) frameworks typically treat memory as a flat, passive repository. This leads to semantic isolation, where the temporal links between events and the logical connections between entities are lost, hindering complex reasoning in multi-turn dialogues. In this paper, we introduce **CogMem**, a cognitive graph architecture designed to reconstruct long-term context fidelity. CogMem proposes a human-centric **PEC^{2}F (Person-Event-Concept-Claim-Fact)** schema that structurally organizes dialogue into interconnected episodic traces and semantic knowledge. By explicitly integrating **Claim nodes** within this framework, CogMem ensures epistemic clarity, distinguishing subjective attributions from objective records. Shifting from static retrieval pipelines to agentic active recall, we further develop a Cognitive Search Agent that dynamically navigates this graph using atomic operators (e.g., intersection, temporal scanning). Experiments on the LoCoMo and LongMemEval benchmarks demonstrate that CogMem significantly outperforms state-of-the-art baselines in multi-hop reasoning and temporal consistency, validating the necessity of structural reconstruction over passive vector matching.

1 Introduction

The evolution of Large Language Models (LLMs) from stateless text generators to autonomous personal agents has created a critical demand for persistent, high-fidelity memory (Wang et al., 2024; Xi et al., 2025). While recent advancements in scaling context windows (Team et al., 2024) allow models to buffer vast amounts of recent interaction, relying solely on extended context fails to capture the structured, evolving nature of human biography. True long-term intelligence requires not just storage, but the ability to reconstruct narratives, distill semantic truths, and navigate the social nuances of multi-speaker environments (Tulving et al., 1972).

Despite this need, current approaches predominantly follow a paradigm of **passive retrieval**. Standard Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) and early memory systems typically vectorizing text chunks into a flat index. We argue that this "store-and-retrieve" paradigm suffers from two fundamental cognitive deficits. First, vector-based methods face **Semantic Collapse** (Mallen et al., 2023). In a dense vector space, an objective fact (e.g., "*Jon lost his job*") and a subjective opinion about it (e.g., "*Gina thinks Jon's job loss is a mistake*") are often indistinguishable due to high semantic overlap. This leads to memory contamination, where subjective biases are hallucinated as objective reality, violating the principles of Theory of Mind (ToM) (Rabinowitz et al., 2018). Second, static retrieval pipelines lack **intentionality**. Complex dialogue reasoning often requires connecting disparate information points across time. A rigid retrieval algorithm cannot dynamically adjust its search strategy based on intermediate findings, failing to perform the multi-hop reasoning required for deep understanding (Press et al., 2023).

To bridge these gaps, we propose **CogMem**, a schema-aware cognitive architecture that shifts the design philosophy from passive retrieval to **agentic active recall**. Drawing inspiration from the Complementary Learning Systems (CLS) theory (Kumaran et al., 2016), CogMem unifies memory storage and reasoning through a novel **PEC^{2}F (Person-Event-Concept-Claim-Fact)** schema. Unlike generic knowledge graphs, this human-centric schema explicitly incorporates **Claim nodes** to structurally decouple subjective attributions from objective records, ensuring epistemic clarity. To activate this structure, we implement a **Cognitive**

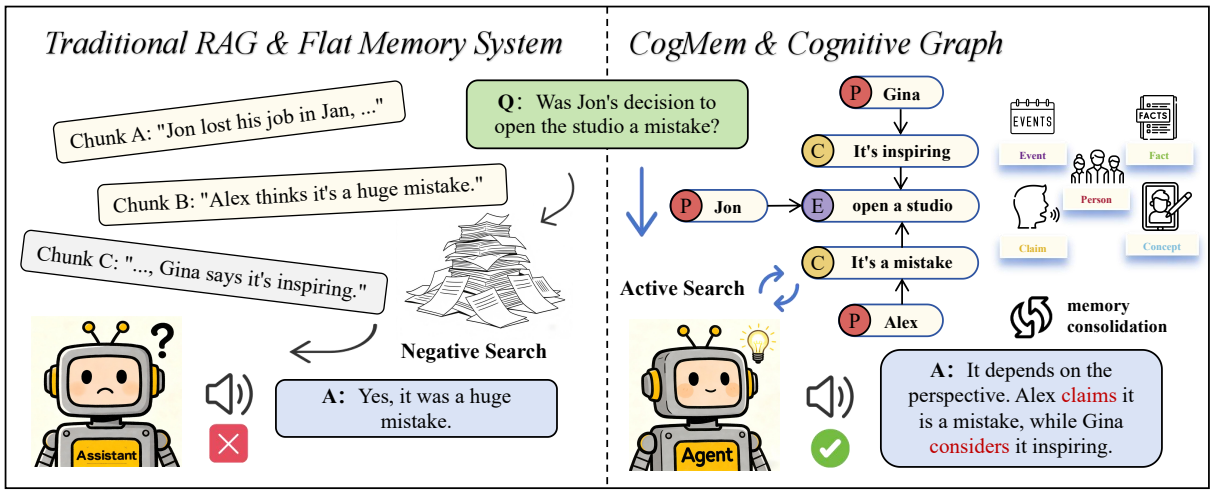

Figure 1: **The Paradigm Shift from Passive Retrieval to Active Reconstruction.** *Left(Traditional RAG & Flat Memory System)* Relies on Passive Retrieval over flat text chunks. The coarse granularity of chunks introduces noise and fails to retrieve relevant counter-evidence (Gina's positive view). Furthermore, even with retrieved context (Alex's view), the flat structure creates epistemic ambiguity, causing the LLM to conflate a subjective claim with objective reality, resulting in a biased, one-sided answer. *Right(CogMem)* Employs a PEC^{2}F Cognitive Graph. The Agent performs Active Reconstruction, navigating the graph to correctly attribute conflicting viewpoints to their sources (Jon vs. Gina) and synthesize a consistent answer.

Search Agent based on the ReAct framework (Yao et al., 2022). Instead of static matching, the agent utilizes atomic graph operators—such as intersection and temporal scanning—to actively navigate the memory space, reconstructing answers through dynamic reasoning paths.

We evaluate CogMem on the LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025) benchmarks. Results demonstrate that CogMem significantly outperforms state-of-the-art baselines, including LightMem (Fang et al., 2025) and General Agentic Memory (Yan et al., 2025), establishing a new standard for consistency in long-term dialogue.

In summary, our contributions are threefold:

- **Epistemic Completeness via the PEC^{2}F Schema.** We propose a novel cognitive schema where the **Subjective-Objective Duality** is modeled through the C^2 components: *Concepts* for semantic grounding and *Claims* for epistemic attribution. This structurally resolves the problem of Semantic Collapse.

- **Dynamic Stability via Memory Consolidation.** We implement a memory evolution mechanism that dynamically synthesizes sparse episodic traces (V_E) into dense semantic knowledge (V_F) using an **Evidence Mounting** strategy. This ensures the memory system remains stable over time without sacrificing granular details, reducing reasoning overhead.

- **Active Plasticity via Agentic Recall.** We shift the retrieval paradigm from static pipelines to **agentic active recall**. We design a Cognitive Search Agent equipped with topological operators to navigate the graph, achieving state-of-the-art performance in multi-hop and temporal reasoning tasks.

2 Related Work

Our work integrates insights from Retrieval-Augmented Generation, Agentic Memory, and Cognitive Science.

2.1 Retrieval-Augmented Generation (RAG)

RAG (Lewis et al., 2020; Guu et al., 2020) mitigates LLM hallucinations by retrieving external context, typically using dense vector indices (Karpukhin et al., 2020; Khattab and Zaharia, 2020). While effective for factoid QA (Chen et al., 2017), vector-based retrieval suffers from semantic isolation in complex reasoning tasks (Mallen et al., 2023; Shuster et al., 2021). To address this, Neuro-symbolic approaches have integrated Knowledge Graphs (KGs) with LLMs (Pan et al., 2024; Yasunaga et al., 2021). Recent advancements like LightRAG (Guo et al., 2025) and HippoRAG 1&2 (Jimenez Gutierrez et al., 2024; Gutiérrez

et al., 2025) employ graph partitioning or personalized PageRank to capture structural dependencies. However, these methods typically utilize generic schemas suited for encyclopedic facts. They lack the specific *episodic-temporal* granularity required for dialogue reconstruction and do not model the *epistemic modality* needed to distinguish objective events from subjective claims.

2.2 Memory Systems for Autonomous Agents

A growing body of work focuses on equipping agents with persistent memory. **Context Management.** Early approaches like MemGPT (Packer et al., 2024) manage infinite context via operating-system-like paging, while MemoryBank (Zhong et al., 2024) uses Ebbinghaus forgetting curves to encode memory updates. **Structured Memory.** Systems such as Mem0 (Chhikara et al., 2025) and MemoryOS (Kang et al., 2025) introduce hierarchical storage for user preferences. A-MEM (Xu et al., 2025) proposes a self-evolving memory bank refined through experience. **Efficiency & Reasoning.** LightMem (Fang et al., 2025) optimizes retention via pruning to ensure low latency, while GAM (Yan et al., 2025) treats recall as an iterative deep research process. *Distinction:* Unlike these systems which predominantly rely on recursive text summarization or flat embeddings, CogMem introduces the PEC^{2}F schema to structurally enforce data provenance. Furthermore, we replace implicit language-based recall with explicit graph operators (e.g., set intersection), offering a more rigorous path for logical reasoning.

2.3 Cognitive Architectures and Theory of Mind

Cognitive frameworks have long inspired AI design (Sumers et al., 2023). Generative Agents (Park et al., 2023) demonstrated that agents could simulate believable behavior by reflecting on memory streams. Our work aligns with the Complementary Learning Systems (CLS) theory (Kumaran et al., 2016; McClelland et al., 1995), formalizing the interplay between fast episodic learning and slow semantic consolidation. Crucially, our architecture addresses the challenge of Theory of Mind (ToM) in LLMs (Sap et al., 2022; **?**). While previous works probe LLMs for ToM capabilities, CogMem provides an architectural guarantee for ToM by explicitly decoupling subjective claims from objective reality via the Claim Node mechanism, preventing the contamination of the agent's world model.

3 Methodology

We propose CogMem, a cognitive architecture designed to reconstruct long-term memory from continuous dialogue streams. Departing from the storage-centric view of traditional RAG systems, we posit that an ideal long-term memory system must satisfy three fundamental cognitive properties: (1) Epistemic Completeness, the ability to structurally differentiate objective reality from subjective belief to prevent memory contamination; (2) Active Plasticity, the capacity to dynamically construct retrieval pathways based on vague intents rather than static indices; and (3) Dynamic Stability, the mechanism to evolve volatile episodic traces into stable semantic knowledge over time. In this section, we formalize how CogMem realizes these properties through a unified graph-agent framework.

3.1 Epistemic Completeness: The PEC^{2}F Schema

To resolve the ambiguity inherent in unstructured memory, we formalize the memory space as a Person-Centric graph $\mathcal{G} = (V, E)$, governed by the **PEC^{2}F** (Person-Event-Concept-Claim-Fact) schema. The vertex set is partitioned into five distinct cognitive dimensions: $V = V_P \cup V_E \cup V_C \cup V_{Claim} \cup V_F$.

Problem Formulation: Semantic Collapse. A fundamental challenge in dialogue memory is the phenomenon of *Semantic Collapse*. In standard dense retrieval systems, the embedding function $\Phi : \mathcal{X} \to R^d$ projects propositional content (e.g., objective facts) and propositional attitudes (e.g., subjective opinions about those facts) into proximal regions of the latent space. Consider an objective fact f denoting "*Jon likes dancing*" and a subjective claim c denoting "*Gina thinks Jon likes dancing*." Our empirical observations reveal that their cosine similarity $\cos(\Phi(f), \Phi(c)) \to 1$. This implies that within the isotropic vector space, the boundary between *truth* and *belief* is blurred, causing retrieval systems to hallucinate subjective biases as objective attributes. To achieve Epistemic Completeness, we introduce $\mathcal{G}$ as a topological space to enforce epistemic decoupling.

Anchors, Traces, and Facts (The Objective Spine). The graph is grounded by **Person** nodes

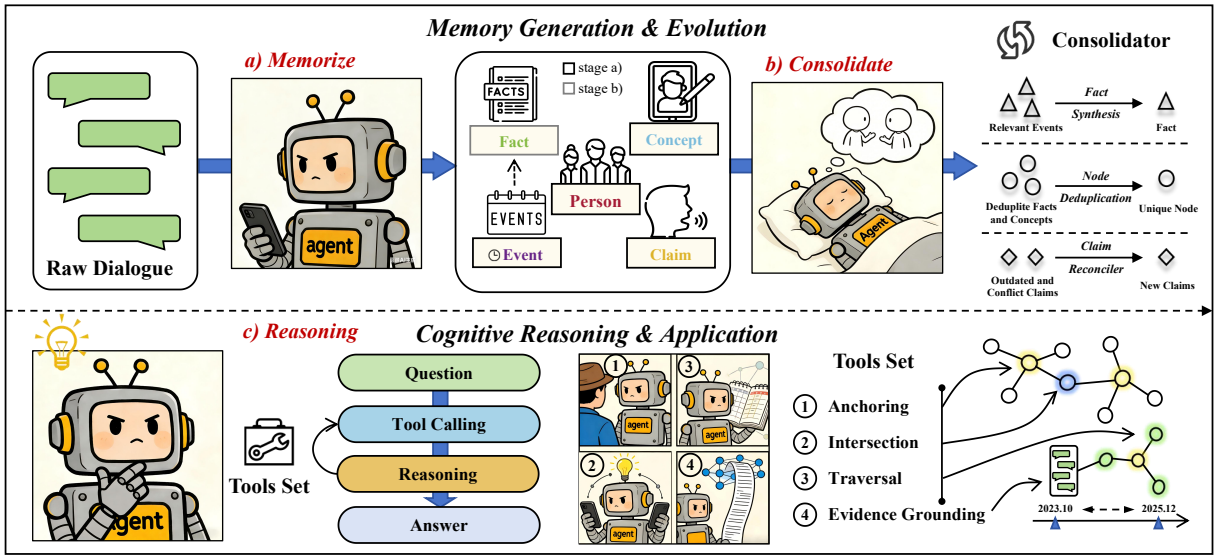

Figure 2: **The Overall Architecture of CogMem.** The framework operates across two layers: (1) **Memory Generation & Evolution (Top)**: (a) *Memorize*: The system ingests raw dialogue to construct the initial cognitive graph, converting episodic traces to graph nodes. (b) *Consolidate*: An offline mechanism (mimicking sleep) refines the graph by synthesizing high-level Facts from Events, deduplicating Facts and Concepts, and reconciling Claims. (2) **Cognitive Reasoning & Application (Bottom)**: (c) *Reasoning*: Shifting from passive retrieval to active recall, the Search Agent utilizes a specialized **Tools Set** (Anchoring, Intersection, Traversal, Evidence Grounding) to dynamically navigate the graph and reconstruct answers via a ReAct loop.

(V_P) representing social agents and **Concept** nodes (V_C) representing abstract semantic entities normalized via vector clustering. These static anchors are dynamically connected by **Event** nodes (V_E), which capture the episodic trace of interactions (Do), and **Fact** nodes (V_F), which represent verified attributes or habits (Is). Drawing from Tulving's theory (Tulving et al., 1972), an event node $e \in V_E$ anchors a set of participants to a specific spacetime coordinate. This design allows the system to distinguish repetitive semantic patterns (Facts) from unique episodic experiences (Events), providing the necessary temporal resolution for narrative reconstruction.

Epistemic Decoupling via Claim Nodes. Crucially, we introduce **Claim Nodes** (V_{Claim}) to model the Theory of Mind (ToM) and resolve Semantic Collapse. Instead of storing a subjective opinion as a direct attribute edge, we *reify* it as a hyper-node structure. A statement like "*Jon thinks Gina is lazy*" is decomposed into a two-hop path:

$$(p_{jon}, \texttt{CLAIMS}, n_{claim}) \in E \quad \wedge \\ (n_{claim}, \texttt{ABOUT}, p_{gina}) \in E \tag{1}$$

where $n_{claim} \in V_{Claim}$ encapsulates the content and sentiment. This topological indirection increases the **structural distance** between a subjective stance and an objective fact. Even if their vector embeddings are indistinguishable, the graph topology enables the system to discern the epistemic status of the information—verifiable reality versus attributed belief—thereby ensuring the integrity of the memory store.

3.2 Active Plasticity: Agentic Retrieval via Cognitive Operators

While the PEC^{2}F schema provides a structured representation of reality, its topological complexity renders static retrieval algorithms (e.g., k-hop expansion) inefficient for natural language queries involving intent and temporality. To activate this schema, we introduce **Active Plasticity**, modeling memory retrieval as a sequential decision-making process formulated as a Markov Decision Process (MDP).

Formalizing the Retrieval Trajectory. We define the retrieval process as a trajectory $\tau = \{s_0, a_0, o_0, \ldots, s_T\}$ generated by the agent over the graph $\mathcal{G}$. At step t, the state s_t encapsulates the query q, the interaction history, and the partial subgraph visited. The agent's policy π_θ, parameterized by the LLM, selects an operator $a_t \in \mathcal{O}$ to maximize information gain. This iterative ReAct loop enables self-correction: if a chosen path yields a null result $\emptyset$, the agent pivots strategy, mirroring

the adaptive nature of human recall.

The Set of Cognitive Operators. We define the action space $\mathcal{O}$ as a set of four orthogonal operators, designed to navigate the specific topology of the PEC^{2}F schema.

Anchoring Operator ($\mathcal{O}_{anchor}$) This operator maps the unstructured query q to a set of entry nodes $V_{start} \subset V$. To mitigate semantic collapse while ensuring precision, we define a hybrid scoring function:

$$S(v, q) = \alpha \cdot I_{\text{lex}}(v, q) + (1 - \alpha) \cdot \cos(\Phi(v), \Phi(q)) \quad (2)$$

where I_{lex} denotes a fuzzy lexical match indicator and Φ is the dense embedding function. The operator returns the top-k nodes maximizing $S(v, q)$, ensuring robustness against entity name variations (e.g., "Jon" vs. "Jonathan").

Spatiotemporal Traversal Operator ($\mathcal{O}_{traverse}$) Unlike standard graph traversal, our operator integrates *Epistemic Filtering* and *Temporal Gating*. Let $\mathcal{N}(v)$ be the neighborhood of node v. The operator filters neighbors based on edge type set R and a temporal constraint T:

$$\mathcal{O}_{traverse}(v, R, T) = \{u \in \mathcal{N}(v) \mid \text{type}(v, u) \in R \wedge \text{time}(u) \in T\} \quad (3)$$

This formulation allows the agent to execute complex queries atomically. For instance, to retrieve opinions, the agent sets $R = \{\texttt{CLAIMS}\}$; to reconstruct a specific narrative, it sets $R = \{\texttt{PARTICIPATED\_IN}\}$ and restricts T to a specific window (e.g., "last month"). This unification prevents the retrieval of irrelevant facts or events outside the temporal scope of interest.

Intersection Operator ($\mathcal{O}_{intersect}$) Designed for high-order reasoning (e.g., commonality discovery), this operator computes the topological overlap between the semantic neighborhoods of multiple entities $\{p_1, \ldots, p_n\}$. Formally:

$$\mathcal{G}_{common} = \bigcap_{i=1}^{n} \left(\bigcup_{r \in R_{sem}} \mathcal{N}_r(p_i) \right) \quad (4)$$

where R_{sem} represents semantic relation types (e.g., `LIKES`, `USES`). By explicitly identifying shared Concepts or Events in the intersection set, this operator resolves queries that are computationally intractable for vector-based retrieval, which lacks the capacity for set-theoretic operations.

Evidence Grounding Operator ($\mathcal{O}_{ground}$) To mitigate the risk of graph construction errors or hallucinations, we introduce a verification mechanism that maps abstract nodes back to the raw corpus $\mathcal{D}$.

$$\mathcal{O}_{ground}(v) \to \{d \in \mathcal{D} \mid \text{source}(v) = d\} \quad (5)$$

This operator retrieves the original text span associated with a node (Event/Fact/Claim). By grounding the graph abstraction in the original context, the agent validates the structural inference, ensuring the final response is textually faithful.

3.3 Dynamic Stability via Systemic Consolidation

Memory is not a static repository but a dynamic system that must reconcile the fidelity of specific episodes with the stability of generalized knowledge. Without consolidation, the continuous influx of dialogue events increases the graph's topological entropy, expanding the search space and degrading retrieval latency. To achieve **Dynamic Stability**, we implement a consolidation mechanism inspired by the Complementary Learning Systems (CLS) theory (Kumaran et al., 2016), modeling the transfer of information from the fast-learning hippocampus (sparse episodic traces) to the slow-learning neocortex (structured semantic nodes).

Graph Reification: From Atomic Edges to Fact Nodes. During the online interaction phase, CogMem employs a "broad-in" ingestion strategy, capturing information primarily as atomic attribute edges (e.g., $p \xrightarrow{LIKES} c$) or discrete event nodes (V_E) to minimize encoding latency. This forms the L1 Associative Layer. The consolidation process operates offline as a graph rewriting function. Formally, let $\mathcal{S}_{p,c} = \{e_1, \ldots, e_k\} \subseteq V_E$ be a cluster of episodic nodes where person p interacts with concept c. When the cluster density $|\mathcal{S}_{p,c}|$ exceeds a threshold τ, the system triggers an inductive abstraction function Ψ parameterized by an LLM to *reify* the pattern into a high-order **Fact Node**:

$$f_{new}, \mathcal{T}_{env} \leftarrow \Psi(\mathcal{S}_{p,c}) \quad (6)$$

where $f_{new} \in V_F$ represents the synthesized semantic proposition (e.g., "*Jon has a habit of dancing*") and $\mathcal{T}_{env} = [t_{start}, t_{end}]$ denotes the computed **Temporal Envelope**. This transformation promotes implicit edge patterns into explicit nodes, effectively compressing the semantic space while preserving temporal boundaries.

Evidence Mounting and Multi-Resolution Retrieval. A critical risk in memory compression is the loss of granular details (catastrophic interference). To resolve the tension between abstraction and precision, CogMem employs an **Evidence Mounting** strategy. Instead of discarding the original episodes after synthesis, we establish provenance edges to anchor the abstract fact in concrete reality:

$$E_{prov} = \{(f_{new}, \texttt{SUPPORTED\_BY}, e_i) \mid \forall e_i \in \mathcal{S}_{p,c}\} \tag{7}$$

This topological hierarchy enables the Search Agent to operate at **dual resolutions**. For high-level queries (e.g., "*What are Jon's hobbies?*"), the agent retrieves the consolidated Fact node directly, minimizing cognitive load. However, should the user probe for evidence (e.g., "*When did he first go?*"), the agent can traverse the SUPPORTED_BY edges to drill down into the original episodic layer. This mechanism ensures that the memory system evolves towards semantic stability without sacrificing the granularity of the original experience.

4 Experiments

To validate the effectiveness of CogMem's PEC^{2}F schema and agentic retrieval mechanism, we conduct comprehensive evaluations on two challenging benchmarks: **LoCoMo** (Maharana et al., 2024) for complex conversational reasoning and **LongMemEval** (Wu et al., 2025) for long-term consistency.

4.1 Experimental Setup

Datasets and Metrics. **LoCoMo** evaluates long-context capabilities across four question types: Single-Hop, Multi-Hop, Temporal, and Open Domain. Following standard protocols, we report **F1 Score** and **BLEU-1**. **LongMemEval** assesses the stability of memory updates and cross-session retrieval. We report **Accuracy (Acc)** derived from LLM-based judging.

Baselines. We compare CogMem against a diverse set of state-of-the-art methods: (1) **Memory-free**: Long-LLM (full context) and Naive RAG. (2) **Memory-based**: Mem0 (Chhikara et al., 2025), MemoryOS (Kang et al., 2025), A-Mem (Xu et al., 2025), LightMem (Fang et al., 2025), and GAM (Yan et al., 2025),. (3) **Graph-based**: LightRAG (Guo et al., 2025) and HippoRAG2 (Gutiérrez et al., 2025).

Implementation. Experiments are conducted using two backbone LLMs: **GPT-4o-mini** and **Qwen2.5-14B-Instruct**. We use **BGE-M3** (Chen et al., 2024) as the embedding model for all retrieval tasks to ensure a fair comparison. The Search Agent is configured with a maximum reasoning depth of 5 steps.

4.2 Main Results

Table 1 summarizes the performance on the LoCoMo benchmark. CogMem demonstrates distinct advantages in tasks requiring structural navigation.

Superiority in Multi-Hop Reasoning. The most significant result is observed in the *Multi-Hop* category. Using Qwen2.5-14B, CogMem achieves an F1 score of **53.49%**, outperforming the strongest baseline GAM (42.96%) by over **10 points** and surpassing LightRAG (26.03%) by a substantial margin. This empirical evidence validates our core hypothesis: while static graph retrieval (LightRAG) fails to connect distant nodes without explicit semantic overlap, our **Search Agent** successfully utilizes the *Intersection Operator* to bridge disconnected subgraphs.

Resilience in Temporal Queries. In the *Temporal* category, CogMem (24.88%) outperforms comparable graph-based methods like LightRAG (20.86%). While GAM achieves higher scores through computationally expensive recursive summarization, CogMem achieves competitive precision via explicit Event nodes and temporal attributes, without the need for extensive context reprocessing.

Long-Term Consistency (LongMemEval). Table 2 presents the accuracy on LongMemEval. This benchmark challenges the system's ability to maintain coherent entity attributes over extended sessions. CogMem achieves competitive performance, particularly in **Multi-Session** and **Single-User** tracking. This success is attributed to the *Concept Manager* and *Consolidation* mechanism, which effectively merge redundant entity mentions into unified Concept nodes, preventing the fragmentation of long-term profiles often seen in Naive RAG.

4.3 Ablation Studies

To dissect the contribution of CogMem's core components, we conduct ablation studies using the Qwen2.5-14Binstruct backbone on the LoCoMo dataset. We analyze three dimensions: the necessity of the agentic loop, the impact of memory con-

Table 1: Performance comparison on the **LoCoMo** benchmark. Metrics are F1 Score and BLEU-1. Best results are in **bold**, second best are underlined.

Backbone	Method	Single Hop		Multi Hop		Temporal		Open Domain	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
GPT-4o-mini	LONG-LLM	46.68	37.54	29.23	22.76	25.97	19.42	16.87	13.70
	NAIVE RAG	52.45	47.94	27.50	20.13	46.07	40.35	23.23	17.94
	LIGHTRAG	42.57	33.82	28.46	23.75	22.85	16.18	**54.33**	**49.61**
	HIPPORAG2	39.81	31.19	39.79	37.40	26.74	22.31	51.41	50.15
	A-MEM	44.65	37.06	27.02	20.09	45.85	36.67	12.14	12.00
	MEM0	47.65	38.72	38.72	27.13	48.93	40.51	28.64	21.58
	MEMORYOS	48.62	42.99	35.27	25.22	41.15	30.76	20.02	16.52
	LIGHTMEM	41.79	37.83	29.78	24.80	43.71	39.72	16.89	13.92
	GAM	**57.75**	**52.10**	42.29	34.44	**59.45**	**53.11**	33.30	26.97
	COGMEM	51.20	46.50	**55.12**	**50.80**	29.15	23.40	36.50	41.20
Qwen2.5-14B	LONG-LLM	46.05	39.56	32.08	24.46	30.51	24.45	14.89	11.41
	NAIVE RAG	47.87	42.79	26.38	19.54	30.78	25.97	14.16	10.52
	LIGHTRAG	41.50	34.42	26.03	21.73	20.86	16.83	**54.03**	**48.65**
	HIPPORAG2	33.10	25.74	34.78	31.15	24.88	18.56	53.42	48.98
	A-MEM	33.75	30.04	22.09	15.28	27.19	22.05	13.49	10.74
	MEM0	42.58	35.15	31.73	24.82	28.96	26.24	15.03	11.28
	MEMORYOS	46.33	41.62	38.19	29.26	32.24	27.86	20.27	15.94
	LIGHTMEM	34.92	31.22	25.45	19.61	32.03	27.70	15.81	11.81
	GAM	**58.93**	**53.74**	42.96	34.48	**51.52**	**44.43**	30.63	26.04
	COGMEM	30.10	26.77	**53.49**	**49.11**	25.12	28.56	34.95	39.66

Table 2: Overall Accuracy (%) on **LongMemEval**. Best results in **bold**, second best are underlined.

Method	Accuracy (%)	
	GPT-4o-mini	Qwen2.5-14B
FULL TEXT	56.80	54.80
NAIVE RAG	61.00	60.80
LIGHTRAG	52.93	47.63
HIPPORAG2	54.34	51.61
A-MEM	62.60	**65.20**
MEM0	53.61	39.51
MEMORYOS	44.80	49.60
LIGHTMEM	64.29	61.95
GAM	63.82	58.71
COGMEM (OURS)	**65.10**	63.45

solidation on reasoning efficiency, and parameter sensitivity.

4.3.1 Impact of Agentic Active Recall

A core premise of CogMem is that complex schemas require dynamic exploration. We compare our full **Agentic** framework against a **Fixed Flow** baseline, which executes a rigid sequence (Anchoring → 1-hop Traversal → Answer) without self-correction or intersection tools. Removing agentic capabilities leads to a catastrophic performance drop: the F1 score plummets by **65.3%** (0.373 → 0.129). Detailed numerical results and trend analysis are provided in **Appendix C**.

4.3.2 Effectiveness of Memory Consolidation

We evaluate the impact of the offline consolidation mechanism, specifically focusing on how the transformation from episodic events to semantic facts affects both accuracy and reasoning cost. Table 3 compares performance *Before* and *After* consolidation.

The results demonstrate a dual advantage. **Improved Accuracy**: Multi-Hop F1 scores increase by **4.86%** (53.49% → 58.35%). By synthesizing scattered events into high-level Fact nodes (e.g., generalizing repeated "dancing" events into a "habit"), the system creates semantic shortcuts that facilitate graph traversal.**Enhanced Efficiency**: Crucially, consolidation significantly reduces the cognitive load on the agent. The average reasoning steps for Multi-Hop queries drop from **3.42 to 2.67**. This indicates that the agent can retrieve aggregated knowledge directly from Fact nodes without traversing exhaustive event chains, validating the **Dynamic Stability** of our architecture.

Table 3: Impact of **Memory Consolidation** on LoCoMo. We compare F1 Score and Average Reasoning Steps (inference turns). Consolidation improves accuracy while reducing reasoning cost.

Category	F1 Score (%)		Avg. Steps	
	Before	**After**	Before	**After**
Single Hop	30.10	**32.44**	2.84	**2.35**
Multi Hop	53.49	**58.35**	3.42	**2.67**
Temporal	24.88	24.88	2.78	**2.44**
Open Domain	34.95	**36.26**	2.86	**2.55**

4.3.3 Hyperparameter Sensitivity

We further analyze the sensitivity of key retrieval parameters, specifically the Top-K values for the *Anchoring Operator* and *Intersection Operator*. Our experiments reveal an inverted U-shape performance curve. Extremely low K values lead to recall failures (anchoring blindness), while excessively high K values introduce noise that distracts the agent, marginally degrading F1 scores while linearly increasing context consumption. We identify an optimal operational zone where recall saturates before noise becomes detrimental. Detailed numerical results and trend analysis for these parameters are provided in **Appendix B.1**.

4.3.4 Cost-Benefit Analysis

While agentic reasoning inherently incurs higher inference latency than static retrieval, our efficiency analysis (see **Appendix E**) demonstrates that CogMem remains highly competitive. Specifically, our memory construction cost (1,578k tokens) is lower than major agentic baselines like A-MEM and Mem0. CogMem strikes a pragmatic balance: it accepts a moderate increase in inference cost to achieve the breakthrough in multi-hop reasoning (F1 +31.4% vs. A-mem) required for high-fidelity personal agents.

5 Conclusion

In this work, we introduced **CogMem**, a cognitive architecture that transitions the paradigm of LLM long-term memory from passive, flat retrieval to active, structural reconstruction. By formulating memory retrieval as a sequential reasoning process over a PEC^{2}F cognitive graph, we addressed the critical failures of existing RAG systems: the collapse of semantic distinctions and the inability to navigate complex relational paths.

Our empirical results on the LoCoMo and LongMemEval benchmarks validate two central hypotheses. First, **Structure is Cognition**. The introduction of the PEC^{2}F schema, provides the necessary Epistemic Completeness to decouple subjective opinions from objective reality, resolving memory contamination issues that plague vector-only systems. Second, **Stability requires Dynamics**. The integration of an offline consolidation mechanism ensures Dynamic Stability, effectively evolving sparse episodic traces into dense semantic knowledge, thereby improving both reasoning accuracy and efficiency. We posit that CogMem serves as a blueprint for the next generation of Personal AI agents, demonstrating that high-quality memory is not merely a storage problem, but a structural modeling challenge.

6 Limitations

While CogMem establishes a strong baseline for cognitive memory, several avenues remain for future exploration.

Towards a More Complete Cognitive Schema. Although the PEC^{2}F schema covers the essential dimensions of dialogue (Do, Is, Use, Belief), human cognition involves even deeper layers. Future iterations could expand the schema to include dynamic Goal Nodes that track the user's evolving intent over time, or Emotion Vectors that modulate the retrieval weight of memories based on affective intensity, mimicking the flashbulb memory effect.

Evolving the Cognitive Toolset. The current Search Agent relies on a set of hand-crafted, orthogonal operators (e.g., Intersection, Traversal). While effective, this set is fixed. A promising direction is Neural Tool Learning, where the agent can autonomously compose or even invent new retrieval primitives based on feedback from failed queries. For instance, the agent could learn a composite operator for "Counterfactual Search" to verify conflicting claims without explicit hard-coding.

Real-time vs. Offline Consolidation. Our current consolidation mechanism operates periodically (offline), analogous to human sleep. However, biological systems also perform *online consolidation* (wakeful rest). Future work will investigate **Incremental Consolidation** algorithms that can update semantic facts in real-time during the conversation stream, reducing the latency between an event's occurrence and its availability as generalized knowledge.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jianlv Chen and 1 others. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *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. 2025. Lightmem: Lightweight and efficient memory-augmented generation. *Preprint*, arXiv:2510.18866.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *Preprint*, arXiv:2502.14802.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory os of ai agent. *Preprint*, arXiv:2506.06326.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Dharshan Kumaran, Demis Hassabis, and James L McClelland. 2016. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512–534.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *Preprint*, arXiv:2402.17753.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

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.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Joon Sung Park, Joseph O'Brien, Carrie Jun 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.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. 2018. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3762–3780, Abu

Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. 2023. Cognitive architectures for language agents. *Transactions on Machine Learning Research*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Endel Tulving and 1 others. 1972. Episodic and semantic memory. *Organization of memory*, 1(381-403):1.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *Preprint*, arXiv:2410.10813.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

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*.

B. Y. Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. 2025. General agentic memory via deep research. *Preprint*, arXiv:2511.18423.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

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 Implementation Details

A.1 Hardware Infrastructure

All experiments were conducted on a high-performance computing cluster node equipped with **4 × NVIDIA A800 GPUs** (80GB VRAM each), a **48-core CPU**, and **720GB of RAM**. This robust infrastructure ensured efficient inference for large-scale baselines (e.g., Qwen2.5-14B) and rapid graph construction processing.

A.2 Model Configurations

Backbone Models. We utilized two primary LLMs for generation and agentic reasoning:

- **GPT-4o-mini**: Accessed via the OpenAI API, serving as a representative of closed-source advanced models.
- **Qwen2.5-14B-Instruct**: Deployed locally using vLLM for high-throughput inference, representing open-source models with strong reasoning capabilities.

Embedding Model. For all vector-based retrieval tasks (including the vector components of CogMem, LightRAG, and so on), we employed BGE-M3. Using a unified embedding model ensures that performance differences are attributable to the system architecture rather than the quality of semantic representation.

A.3 Baseline Settings

To ensure a fair comparison, all baseline methods were reproduced using their official open-source implementations with **default configurations** recommended by their respective authors.

For CogMem, the Search Agent was configured with a maximum reasoning depth of 5 steps, and the consolidation threshold was set to 3 events per concept cluster.

B Detailed Hyperparameter Analysis

B.1 Detailed Hyperparameter Analysis

In this section, we present a granular sensitivity analysis of the core cognitive operators defined in our methodology. We investigate the impact of the retrieval limit (K) and search depth (D) on the system's reasoning capabilities.

B.2 Anchoring Operator Sensitivity

Table 4 illustrates the impact of the number of candidate nodes (K) retrieved during the initial *Anchoring* phase. The results follow an inverted U-shape. A low K ($K = 3$) leads to "anchoring blindness," where the agent fails to retrieve the correct entry points for the graph, resulting in a significant performance drop (-4.6%). Performance saturates between $K = 15$ and $K = 18$, after which introducing more candidates adds noise and marginally degrades the F1 score while increasing the context load.

Table 4: Sensitivity analysis of the **Anchoring Operator**. Increasing K improves recall up to a saturation point.

Top-K	F1 Score	Avg. Steps
$K = 3$	32.70	2.76
$K = 5$	34.72	2.68
$K = 8$	35.09	2.71
$K = 10$	35.82	2.72
$K = 13$	36.74	2.68
$K = 15$	36.74	2.72
$K = 18$	**37.30**	2.69
$K = 20$	37.11	2.73

B.3 Traversal Operator Sensitivity

The *Traversal Operator* is governed by two parameters: the maximum number of neighbors to retrieve per hop (Max Results) and the maximum depth of the subgraph exploration (Max Depth).

Max Results Limit. As shown in Table 5, retrieving too few neighbors ($K = 5$) truncates potential reasoning paths. The optimal balance is found at $K = 15$. Beyond this point, the LLM struggles to attend to the relevant edges among the noise, leading to a slight performance regression.

Search Depth. Table 6 analyzes the impact of traversal depth. A depth of 1 (immediate neighbors) is insufficient for complex multi-hop reasoning. However, increasing the depth to 3 or 4 does not yield linear gains; instead, it exponentially increases the context size, often confusing the agent with irrelevant distant connections. A depth of 2 (exploring the neighborhood of neighbors) proves to be the optimal setting for the LoCoMo dataset.

Table 5: Impact of the **Traversal Operator's** neighbor limit (Max Results) on performance.

Max Results	F1 Score	Avg. Steps
$K = 5$	35.94	2.72
$K = 10$	35.94	2.72
$K = 15$	**37.30**	2.69
$K = 20$	35.94	2.67

Table 6: Impact of **Traversal Depth**. A depth of 2 provides the best trade-off for multi-hop reasoning.

Max Depth	F1 Score	Avg. Steps
$D = 1$	36.47	2.66
$D = 2$	**37.30**	2.71
$D = 3$	36.47	2.67
$D = 4$	36.47	2.70

B.4 Intersection Operator Sensitivity

The *Intersection Operator* (finding common ground) is critical for solving comparison queries. Table 7 shows that the optimal K for intersection candidates is 10. This suggests that salient commonalities (e.g., shared hobbies or events) typically appear within the top-ranked connections. Expanding the search space further ($K > 15$) dilutes the semantic focus, reducing the F1 score while increasing computational cost.

Table 7: Sensitivity of the **Intersection Operator**. Performance peaks at $K = 10$, indicating that commonalities are usually found in top-ranked connections.

Top-K	F1 Score	Avg. Steps
$K = 3$	35.17	2.68
$K = 5$	35.36	2.72
$K = 8$	36.91	2.73
$K = 10$	**37.30**	2.72
$K = 13$	35.75	2.68
$K = 15$	36.72	2.68
$K = 18$	35.94	2.75
$K = 20$	35.94	2.73

C Detailed Ablation: Agentic vs. Fixed Flow

In Section 4.3, we discussed the necessity of the agentic loop. Here, we provide the detailed quantitative comparison between our **Cognitive Search Agent** and a **Fixed Flow** baseline.

Fixed Flow Configuration. The Fixed Flow baseline is designed to mimic a standard GraphRAG retrieval pipeline without dynamic reasoning. It executes a rigid, pre-defined sequence of operations for every query:

1. **Anchoring**: Retrieve top-K nodes based on the query.
2. **Traversal**: Perform a 1-hop expansion from these anchors to retrieve immediate neighbors.
3. **Answer**: Force the LLM to generate an answer based solely on the retrieved context, with no opportunity for self-correction, intersection, or multi-step navigation.

Results Analysis. Table 8 presents the comprehensive results. The **Agentic** approach outperforms the Fixed Flow across all quality metrics by a wide margin (approx. 65% improvement). Notably, the Fixed Flow records a higher average number of steps (4.00 vs. 3.09). This is counter-intuitive but instructive: the Fixed Flow is hard-coded to execute a full retrieval sequence regardless of early success, whereas the Cognitive Agent often identifies the answer via a "Direct Hit" (e.g., finding the answer immediately in the Anchoring observation) and terminates early. This demonstrates that **Active Plasticity** not only improves accuracy but also computational efficiency by avoiding unnecessary graph traversals.

Table 8: Ablation study of the Agentic loop. "Fixed" denotes the static pipeline baseline.

Method	F1	BLEU	Judge	Steps
CogMem (Ours)	**37.3**	**38.0**	**70.4**	**3.09**
Fixed Flow	12.9	12.9	24.4	4.00
Change (%)	*-65.3*	*-66.1*	*-65.4*	*+29.4*

D Detailed Analysis of Memory Consolidation

This section provides a granular breakdown of the offline memory consolidation process. We analyze the structural changes in the cognitive graph and the computational costs involved.

D.1 Graph Structure Evolution

Table 9 illustrates the changes in graph topology following the consolidation process. Notably, we observe a net **increase** in total nodes (+540) and edges (+4,708). This is intentional: our current consolidation strategy employs **Additive Evidence Mounting**. When synthesizing high-level **Fact Nodes**, we **retain** the original Event nodes and link them via `SUPPORTED_BY` edges rather than deleting them. This conservative approach is designed to preventing **error propagation**—if the LLM hallucinates during abstraction, the original ground truth remains accessible for verification. While this increases the storage footprint, it significantly reduces the logical search space for the agent by providing semantic shortcuts. Optimizing storage efficiency via active pruning or archival mechanisms remains a direction for future work.

Table 9: **Graph Topology Changes**. Consolidation synthesizes new Fact nodes while merging redundant Concepts.

Node Type	Before	After	Δ	Change (%)
Person	99	99	0	0.0%
Event	697	697	0	0.0%
Concept	832	819	-13	-1.6%
Fact	0	497	**+497**	–
Claim	2013	2013	0	0.0%
Total Nodes	3641	4181	+540	+14.8%
Total Edges	9935	14643	+4708	+47.4%

D.2 Computational Cost vs. Benefit

Memory consolidation incurs an offline computational cost to optimize online retrieval efficiency.

- **Offline Cost**: The process required 537.85 seconds and approximately 0.79M tokens to consolidate the LoCoMo dataset. This is a one-time cost per batch update.
- **Online Benefit**: As shown in the main text (Table 3), this investment yields a **21.9% reduction** in inference steps for Multi-Hop queries (3.42 → 2.67) and a **+4.86%** gain in F1 Score.

D.3 Category-wise Performance Breakdown

Table 10 provides the raw performance metrics for each question category before and after consolidation. The results confirm that consolidation is most

effective for reasoning-intensive tasks (Multi-Hop) where semantic shortcuts significantly reduce the cognitive load.

Table 10: **Detailed Impact of Consolidation**. Consistent with Table 3 in the main text.

Category	F1 Score (%)		Avg. Steps	
	Before	**After**	Before	**After**
Single Hop	30.10	32.44	2.84	2.35
Multi Hop	53.49	**58.35**	3.42	**2.67**
Temporal	24.88	24.88	2.78	2.44
Open Domain	34.95	36.26	2.86	2.55

E Efficiency and Cost Analysis

In this section, we analyze the computational overhead of CogMem. We present a theoretical complexity analysis and an empirical evaluation of token consumption, distinguishing between the one-time *Memory Construction* phase and the recurring *Inference* phase.

E.1 Theoretical Complexity

Table 11 formalizes the computational complexity. A key advantage of CogMem is that the inference complexity is decoupled from the total document count N. Instead, it scales with the number of reasoning steps R and the local graph density, ensuring scalability for long-context scenarios where N is large.

Table 11: **Theoretical Complexity Analysis.** Notation: N: Num. docs; P: Persons; K: Concepts/person; Q: Queries; R: Reasoning steps; C, E, F: Num. Concept/Event/Fact nodes; T_{llm}: LLM latency.

Stage	LLM Complexity	Vector Ops	Graph Ops
Indexing	$O(N)$	$O(N(C+E))$	$O(N \cdot E)$
Consolidate	$O(P \cdot K)$	$O(C^2 + F^2)$	$O(P \cdot K \cdot E)$
Inference	$O(Q \cdot R)$	$O(Q)$	$O(Q \cdot R \cdot D)$
Total	$O(N + P \cdot K + Q \cdot R)$		

E.2 Memory Construction Cost

We compare the token consumption of CogMem against representative agentic memory baselines during the memory construction phase (Indexing + Summarization/Update). We implement optimizations including KV-Caching for system prompts and strict JSON schema decoding.

- **Indexing**: 926k tokens.
- **Consolidation**: 652k tokens.
- **Total Construction**: **1,578k tokens**.

Comparison. As shown in Table 12, CogMem's construction cost is highly competitive among LLM-driven memory systems. It is lower than both **A-MEM** (1,626k) and **Mem0** (1,799k), indicating that our *Atomic Extraction* strategy is more cost-effective than the recursive summarization used in other baselines.

Note on LightMem: We exclude LightMem from this direct token comparison. LightMem relies heavily on external lightweight models for text compression, which significantly reduces the reported backbone LLM tokens but incurs hidden computational costs in runtime and deployment complexity. Its token usage metric is therefore not directly comparable to systems that perform semantic processing via the main LLM.

Table 12: **Construction Cost Comparison (LoCoMo).** Total tokens (Input + Output) required to build the memory bank using the Qwen2.5-14B backbone. CogMem (Optimized) proves more efficient than major agentic baselines.

Method	Total Tokens (k)	LLM Calls
CogMem (Ours)	**1,578.2**	1,216
A-MEM	1,626.8	1,175
Mem0	1,799.4	1,614
MemoryOS	2,991.8	2,938

E.3 Inference (QA) Overhead

The inference phase involves the dynamic ReAct loop. For the full LoCoMo dataset (1,542 questions), CogMem consumed approximately **10.19M tokens**, averaging **6,610 tokens per query**.

Table 13 details the reasoning statistics. The average reasoning depth is **2.89 steps**, with Multi-Hop questions requiring deeper exploration (3.18 steps). Notably, the optimized system prompt caching reduces the input overhead significantly, making the agentic approach viable for real-time applications despite the iterative nature.

F Detailed Results on LongMemEval

Table 14 presents the comprehensive category-wise accuracy breakdown for the LongMemEval benchmark. This breakdown validates our hypothesis that CogMem's structural schema offers specific advantages in handling temporal dynamics and conflicting information.

Table 13: **Inference Statistics (Post-Consolidation).** Breakdown of reasoning steps and token usage by question type. Step counts align with the optimized "After" results in the ablation study.

Category	Avg. Steps	Est. Tokens/Query
Single Hop	2.35	∼5.3k
Multi Hop	2.67	∼6.3k
Temporal	2.44	∼5.8k
Open Domain	2.55	∼6.0k
Overall	**2.65**	**6,060**

Analysis.

- **Temporal Reasoning**: CogMem achieves **68.42%** (GPT) and 56.39% (Qwen) in the Temporal category. This confirms that explicitly modeling *Event Nodes* with temporal envelopes allows the agent to reconstruct timelines more accurately than vector-based methods like Naive RAG (39.85%) or even graph baselines like HippoRAG2 (55.40%).

- **Knowledge Update**: CogMem dominates this category with **84.62%** (GPT) and **76.92%** (Qwen). The *Claim Node* mechanism effectively isolates old vs. new beliefs, preventing the "Semantic Collapse" seen in LightMem and GAM where conflicting updates often merge.

- **Complex vs. Simple**: While CogMem trails behind Naive RAG in simple *Single-User/Assistant* retrieval (due to the overhead of graph traversal), it significantly outperforms them in complex, evolving scenarios (Multi-Session and Update), aligning with the goal of building long-term companions.

G Qualitative Analysis: Reasoning Trajectories

To illustrate the cognitive reasoning process of CogMem's Search Agent, we present four detailed case studies covering Single-Hop, Multi-Hop, Open-Domain, and Temporal queries. These traces demonstrate how the agent dynamically selects operators to navigate the PEC^{2}F graph.

Case 1: Single-Hop Reasoning (Fact Retrieval)

Query: "What did Caroline research?"
Reasoning Trace:
Step 1: Anchoring
Thought: Identify the subject 'Caroline'.
Action: $\mathcal{O}_{anchor}$("Caroline", type = [PERSON])
Obs: Found Node: `p_caroline` (Score: 1.0).
Step 2: Traversal
Thought: Explore attributes to find research topics.
Action: $\mathcal{O}_{traverse}$(p_caroline, rel = [HAS_ATTR])
Obs: Found edges to `Concept: Adoption Agencies` and `Concept: Inclusive Agency`.
Step 3: Evidence Grounding
Thought: Verify specific context in source text.
Action: $\mathcal{O}_{ground}$(p_caroline, q = "research?")
Obs: "...researching adoption agencies lately, especially those that are inclusive..."
Answer: Adoption agencies, specifically inclusive ones supporting LGBTQ+ individuals.

Case 2: Multi-Hop Reasoning (Commonality)

Query: "What activities do Caroline and Melanie both enjoy?"
Reasoning Trace:
Step 1 & 2: Anchoring
Thought: Locate nodes for both individuals.
Action: $\mathcal{O}_{anchor}$("Caroline") → p_caroline
Action: $\mathcal{O}_{anchor}$("Melanie") → p_melanie
Step 3: Intersection
Thought: Query asks for "both", implying intersection. I will search for shared Concepts.
Action: $\mathcal{O}_{intersect}$([p_caroline, p_melanie], type = [CONCEPT])
Obs: Found Common Nodes: {`Concept: Camping`, `Concept: Art`}.
Step 4: Verification Traversal
Thought: Confirm the relation type is positive (enjoy/like).
Action: $\mathcal{O}_{traverse}$([p_car, p_mel], rel = [LIKES])
Obs: Validated edges: `p_caroline` $\xrightarrow{LIKES}$ `Camping`; `p_melanie` $\xrightarrow{LIKES}$ `Camping`.
Answer: They both enjoy Camping and Art.

Case 3: Open Domain (Detail Extraction)

Query: "What did Melanie realize after the charity race?"
Reasoning Trace:
Step 1: Anchoring
Action: $\mathcal{O}_{anchor}$("Melanie") → p_melanie
Step 2: Direct Grounding
Thought: The query asks for an internal mental state (realization), which is likely best retrieved from the raw episodic narrative rather than structured nodes.
Action: $\mathcal{O}_{ground}$(p_melanie, q = "realize after race")
Obs: Retrieved Chunk (2023-07-15): "...After finishing, I realized how important self-care is... it's necessary if I want to keep helping others..."
Answer: She realized that self-care is important and necessary for effectively helping others.

Table 14: **Full Category-wise Accuracy (%)** on LongMemEval. We compare CogMem against Memory-free, Graph-based, and Memory-based baselines across two LLM backbones. Best results are in **bold**, second best are underlined.

Backbone	Method	Temporal (133)	Multi-Sess. (133)	Know. Upd. (78)	S-User (70)	S-Asst. (56)	S-Pref. (30)
GPT-4o-mini	FULL TEXT	31.58	45.45	76.92	87.14	89.29	36.67
	NAIVE RAG	39.85	48.48	67.95	90.00	**98.21**	53.33
	LIGHTRAG	43.60	36.80	51.30	82.50	85.70	40.00
	HIPPORAG	55.40	41.20	62.50	81.50	91.00	16.67
	A-MEM	47.36	48.87	64.11	**92.86**	96.43	46.67
	MEM0	40.15	46.21	70.12	81.43	41.07	60.00
	MEMORYOS	32.33	31.06	48.72	80.00	64.29	30.00
	LIGHTMEM	62.10	66.50	70.51	81.43	32.14	66.67
	GAM	55.60	58.20	66.50	82.50	88.50	43.33
	COGMEM	**68.42**	**66.17**	**84.62**	55.71	42.86	**70.00**
Qwen2.5-14B	FULL TEXT	33.08	35.61	74.82	82.86	87.50	50.00
	NAIVE RAG	36.84	47.73	65.38	**91.43**	**98.21**	70.00
	LIGHTRAG	38.20	30.50	48.10	75.40	82.10	35.00
	HIPPORAG	46.21	34.09	57.14	76.81	89.29	10.00
	A-MEM	51.88	51.12	76.93	90.00	96.43	40.00
	MEM0	41.94	28.13	28.57	55.32	26.09	**81.82**
	MEMORYOS	28.57	36.84	61.54	72.86	92.86	33.33
	LIGHTMEM	**56.40**	**61.50**	66.67	80.00	31.25	80.00
	GAM	49.62	51.13	55.13	81.43	92.86	26.67
	COGMEM	53.50	56.50	**76.92**	81.43	71.00	53.33

Case 4: Temporal Reasoning (Timeline Reconstruction)

Query: "When did Caroline move from Sweden?"
Reasoning Trace:
Step 1: Anchoring
Action: $\mathcal{O}_{anchor}$("Caroline") $\rightarrow$ p_caroline
Step 2: Temporal Scanning
Thought: I need to find a specific event regarding "moving from Sweden". I will scan her event timeline with a filter.
Action: $\mathcal{O}_{time}$(p_caroline, q = "move from Sweden")
Obs: Found Event: `moved from Sweden` (Timestamp: 2019-06-15).
Step 3: Verification (Grounding)
Thought: The timestamp seems precise. I will verify the source text to ensure accuracy.
Action: $\mathcal{O}_{ground}$(EventID)
Obs: Text: "When I moved from Sweden in 2019..."
Answer: June 2019.

H Qualitative Analysis: Error Analysis

To provide a deeper understanding of CogMem's performance characteristics and limitations, we conduct a qualitative analysis of representative failure cases from the LoCoMo benchmark. We categorize these errors into five distinct patterns, each revealing specific challenges in cognitive memory modeling.

H.1 Error Pattern 1: Tool Mis-selection in Single-Hop Queries

Case Study:

- **Question**: "What did Caroline research?"
- **Gold Answer**: "Adoption agencies"
- **Predicted**: "Adoption Process"

Analysis: The Search Agent correctly identified the anchor entity (*Caroline*) but selected a suboptimal retrieval tool. Instead of using *Get_Evidence* to find specific `Fact` nodes (e.g., *Fact: Caroline researched adoption agencies*), the agent used *Neighbor Traversal* on the `HAS_ATTRIBUTE` relation. This returned the high-level concept "Adoption Process" (an interest area) rather than the specific object of her research. **Implication:** This highlights the nuanced distinction between an entity's *attributes* (static profile) and their *actions* (dynamic research). Future work should refine the agent's prompt to better distinguish between attribute-lookup and content-retrieval intents.

H.2 Error Pattern 2: Semantic Drift in Vector Retrieval

Case Study:

- **Question**: "What did Melanie realize after the charity race?"
- **Gold Answer**: "self-care is important"
- **Predicted**: "Melanie realized the importance of community support..."

Analysis: This failure stems from **Semantic Collapse** in the vector space. The query "realize after charity race" retrieved `Claim` nodes related to the race (e.g., "community support is important") with moderate similarity scores (0.445), but missed the target chunk containing the specific realization about "self-care." The vector embedding for the target claim was likely dominated by the context of the "charity race" rather than the abstract concept of "realization." **Implication:** While the P-E-C-F schema mitigates semantic collapse for structured data, unstructured open-domain queries still rely heavily on the quality of underlying embeddings. Hybrid retrieval combining keyword matching (BM25) with dense vectors is crucial for such fine-grained detail extraction.

H.3 Error Pattern 3: Temporal Resolution Failure

Case Study:

- **Question**: "When did Melanie paint a sunrise?"
- **Gold Answer**: "2022"
- **Predicted**: "8 May 2023"

Analysis: The system retrieved the correct event (*painted lake sunrise*) but failed to resolve the relative temporal expression correctly. The raw evidence contained the phrase "last year" relative to a session date of "2023-05-08". The extraction module prioritized the session date (2023) over the resolved absolute date (2022). **Implication:** This underscores the complexity of temporal grounding in dialogue. A robust memory system requires a dedicated *Temporal Reasoning Module* that normalizes all relative time expressions (e.g., "yesterday", "last year") into absolute timestamps during the ingestion phase, rather than relying on query-time resolution.

H.4 Error Pattern 4: Information Granularity Mismatch

Case Study:

- **Question**: "What career path has Caroline decided to pursue?"
- **Gold Answer**: "counseling or mental health for Transgender people"
- **Predicted**: "Mental health counseling"

Analysis: The agent successfully retrieved the core concepts (*counseling*, *mental health*) but missed the specific qualifier (*for Transgender people*). This information loss occurred during the **Event-to-Fact Consolidation** phase. The consolidation algorithm abstracted the specific detail into a more general category (*Mental Health Work*) to reduce graph complexity, sacrificing granularity for stability. **Implication:** There is an inherent trade-off between memory compression (consolidation) and detail retention. A potential solution is the **Evidence Mounting** strategy proposed in our methodology, where the generalized Fact node retains links to original Event nodes, allowing the agent to drill down for specific details when needed.

H.5 Error Pattern 5: Incomplete Search Scope

Case Study:

- **Question**: "What books has Melanie read?"
- **Gold Answer**: "Nothing is Impossible", "Charlotte's Web"
- **Predicted**: "No specific titles mentioned"

Analysis: The agent restricted its search to `Fact` and `Event` nodes but failed to check `Claim` nodes. In our schema, a statement like "This book is worth a read" might be stored as a subjective `Claim` rather than an objective `Fact`. By narrowing the search scope too early, the agent missed relevant information stored in a different cognitive modality. **Implication:** This validates the necessity of the **multi-modal search strategy** in our agent design. Agents should be encouraged to explore multiple node types (Events, Facts, Claims) when initial searches yield no results.

H.6 Summary of Limitations

These failure cases reveal that while CogMem's structural approach solves many high-level reasoning problems (e.g., multi-hop, consistency), it introduces new challenges in:

1. **Granularity Control**: Balancing abstraction (Facts) with detail (Events).

2. **Temporal Normalization**: Accurately resolving relative time in narrative streams.

3. **Agent Strategy Optimization**: Ensuring the agent selects the optimal tool for subtle semantic distinctions.

Addressing these limitations through improved prompt engineering and hybrid retrieval strategies remains a focus for future work.

I Prompts

This appendix provides the core prompts used in CogMem's Ingestion, Retrieval, Consolidation, and Eval phases. We present the prompts in a condensed format for readability.

I.1 Ingestion Phase Prompts

ATOMIC_EXTRACTION_SYSTEM (Schema Classification)

Role: You are a Cognitive Graph Extraction Engine. Map conversation text into a structured knowledge graph, distinguishing between OBJECTIVE REALITY (Events/Facts) and SUBJECTIVE MINDS (Claims).
Classification Rules:

- **EVENT (Do)**: Specific, one-time actions with a time reference (e.g., "I went hiking yesterday").
- **FACT (Is/Feel - Self)**: Stable traits, habits, or self-reported preferences (e.g., "I am a doctor", "I love pizza").
- **CLAIM (Believes - Others)**: Opinions about others or concepts (e.g., "Jon thinks Gina is lazy"). *Crucial: Must track Source and Target.*
- **CONCEPT**: Keywords representing topics or objects (normalized to lowercase).

JSON Output Format: `{ "events": [...], "facts": [...], "claims": [...] }`

I.2 Retrieval Phase Prompts (Search Agent)

SEARCH_AGENT_SYSTEM_PROMPT (Cognitive Strategy)

Role: You are CogMem, a cognitive memory assistant. Use the ReAct loop to answer questions.
Tool Usage Strategy:

- **Anchoring**: ALWAYS start with `find_anchors` to get Node IDs.
- **Opinion Queries**: If asked "What does A think of B?", use `traverse_neighbors(A, relation=["CLAIMS"])`.
- **Commonality Queries**: If asked "What do A and B share?", use `find_common_ground([id_a, id_b])`.
- **Temporal Queries**: Use `traverse_neighbors` with the `time_query` parameter.
- **Open Domain**: For detailed explanations ("how/why"), use `get_evidence` to retrieve original text.

Constraints:

1. **Direct Hit**: If the initial observation contains the answer, output `final_answer` immediately.
2. **No Hallucination**: Do not invent Node IDs. Use exact IDs returned by tools.
3. **Provenance**: If evidence is a Claim, explicitly state who holds the opinion.

I.3 Consolidation Phase Prompts

MEMORY_CONSOLIDATION_SYSTEM (Event-to-Fact Synthesis)

Role: You are a Memory Consolidation Engine. Simulate the human process of converting episodic memory into semantic knowledge.
Input: A list of Events where a Person interacted with a Concept. **Task**: Synthesize a generalized Fact.
Rules:

- **Generalization**: 3x "went hiking" $\rightarrow$ "has a habit of hiking".
- **Temporal Envelope**: Calculate `valid_from` (first event) and `valid_to` (last event).
- **Predicate Selection**: Choose precise verbs (e.g., `has_habit`, `is_experienced_in`).

JSON Output: `{ "content": "...", "predicate": "...", "valid_from": "...", "confidence": 0.8 }`

I.4 Answer Prompts

ANSWER_PROMPT

Task: Based on the retrieved evidence, generate a concise final answer.
Refinement Rules:

- **Specific Extraction**: For "What did X research?", extract the *content* (e.g., "Adoption agencies"), not the *attribute* (e.g., "Research topic").
- **Time Formatting**: Convert ISO dates to natural language (e.g., "7 May 2023"). Prefer relative time phrases if present in the source (e.g., "the week before...").
- **Completeness**: List ALL items found (e.g., "violin AND clarinet").
- **Brevity**: Keep answers under 15 words unless explaining a complex "How/Why" question.

I.5 LLM-as-Judge

Temporal Reasoning Tasks(LongMemEval)

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response is equivalent to the correct answer or contains all the intermediate steps to get the correct answer, you should also answer yes. If the response only contains a subset of the information required by the answer, answer no. In addition, do not penalize off-by-one errors for the number of days. If the question asks for the number of days/weeks/months, etc., and the model makes off-by-one errors (e.g., predicting 19 days when the answer is 18), the model's response is still correct.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Standard Tasks(LongMemEval)

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response is equivalent to the correct answer or contains all the intermediate steps to get the correct answer, you should also answer yes. If the response only contains a subset of the information required by the answer, answer no.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Knowledge Update Tasks (LongMemEval)

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response contains some previous information along with an updated answer, the response should be considered as correct as long as the updated answer is the required answer.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Single-session Preference Tasks (LongMemEval)

I will give you a question, a rubric for desired personalized response, and a response from a model. Please answer yes if the response satisfies the desired response. Otherwise, answer no. The model does not need to reflect all the points in the rubric. The response is correct as long as it recalls and utilizes the user's personal information correctly.
Question: {question}
Rubric: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Abstention Tasks(LongMemEval)

I will give you an unanswerable question, an explanation, and a response from a model. Please answer yes if the model correctly identifies the question as unanswerable. The model could say that the information is incomplete, or some other information is given but the asked information is not.
Question: {question}
Explanation: {answer}
Model Response: {response}
Does the model correctly identify the question as unanswerable? Answer yes or no only.

All Tasks(Locomo)

System prompt You are an expert evaluator for QA systems. Your task is to determine if the "Predicted Answer" is semantically CORRECT compared to the "Gold Answer".
Rules for "is_correct": 1. TRUE if the meaning is the same, even if wording differs. 2. TRUE if the core facts are present. 3. FALSE if the answer is "I don't know" but the Gold Answer has facts. 4. FALSE if the answer points to the wrong entity, time, or location. 5. FALSE if the answer is partially correct but misses the main point.
Return JSON: "is_correct": <bool>, "explanation": "<brief explanation, max 20 words>"
User prompt Question: question Gold Answer: gold_answer Predicted Answer: predicted_answer f'Category: category' if category else ''
Output the verdict in JSON format.