APEX-MEM: Agentic Semi-Structured Memory with Temporal Reasoning for Long-Term Conversational AI

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, Ankit Chadha

Amazon, AGI, Sunnyvale, USA

{pratyay, mmasud, ssangu, misrami, ankitrc}@amazon.com

Abstract

Large language models still struggle with reliable long-term conversational memory: simply enlarging context windows or applying naïve retrieval often introduces noise and destabilizes responses. We present APEX-MEM, a conversational memory system that combines three key innovations: (1) a property graph which uses domain-agnostic ontology to structure conversations as temporally grounded events in an entity-centric framework, (2) append-only storage that preserves the full temporal evolution of information, and (3) a multi-tool retrieval agent that understands and resolves conflicting or evolving information at query time, producing a compact and contextually relevant memory summary. This retrieval-time resolution preserves the full interaction history while suppressing irrelevant details. APEX-MEM achieves 88.88% accuracy on LOCOMO's Question Answering task and 86.2% on LongMemEval, outperforming state-of-the-art session-aware approaches and demonstrating that structured property graphs enable more temporally coherent long-term conversational reasoning.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in understanding and generating human-like text, yet they fundamentally struggle with maintaining coherent memory across extended conversations. In applications such as personal assistants and complex goal-oriented tasks, users expect systems to remember prior discussions, accumulate knowledge across sessions, and adapt responses based on evolving context. This expectation is magnified in open-domain conversational agents, which must sustain far more turns across diverse topics (Kim et al., 2023).

A naïve solution is to leverage models with larger context windows to retain longer conversational histories. However, longer contexts trade memory for noise, severely increasing the risk of irrelevant facts or hallucinations in the output of LLMs (Maharana et al., 2024; Zhang et al., 2024; Du et al., 2025). To reduce noise, researchers started leveraging retrieval augmented generation (RAG) techniques to identify most relevant parts of conversation (Packer et al., 2023; Lee et al., 2024; Alonso et al., 2024). In this approach, textual segments or summaries are stored and retrieved during conversation. Yet these methods also suffer from fundamental limitations: retrieval accuracy does not reliably translate to answer accuracy when summaries lose critical details, and increasing the number of retrieved segments frequently reintroduces noise (Pan et al., 2025; Zhang et al., 2024). More broadly, RAG treats memory as unstructured text and offers no mechanism to maintain canonical entities, track evolving facts, or distinguish persistent information from ephemeral conversational content. As conversational histories grow, LLMs struggle to maintain factual consistency, entity continuity, and temporal coherence (Byerly and Khashabi, 2024).

These limitations motivate the shift toward structured memory representations, which explicitly organize conversational knowledge to reduce noise, improve retrieval precision, preserve coherence across long-term interactions, and enable relational and temporal reasoning that unstructured text memory cannot support. Recent research has explored this direction through various architectures: Mem0^g represents memory as entity-centric graphs to capture relational structure (Chhikara et al., 2025), A-MEM constructs dynamic knowledge networks inspired by strategic note taking technique (Zettelkasten) to maintain long-term memory consistency (Xu et al., 2025a), Zep builds temporally-aware knowledge graphs to track evolving facts across sessions (Rasmussen et al., 2025), and Semantic Anchoring leverages linguistic structure and entity-aware memory to enhance persistence and coherence in conversational reasoning (Chatterjee

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16470–16489

and Agarwal, 2025). These approaches demonstrate that structured memory provides a principled foundation for improving long-term conversational performance beyond what search-based unstructured text memory can achieve. Despite these advances, current memory systems face two critical limitations. First, entity-centric graph approaches like Mem0, have limited entity classes and store information primarily as relationships between entities, limiting their ability to capture nuanced attributes and temporal evolution of facts. Second, existing systems consolidate (or overwrite previous) information which risks losing important contextual details needed for temporal reasoning and conflict resolution.

To overcome the limitations of existing structured memory systems, we present APEX-MEM, a conversational memory framework that combines property graphs supported by a domain-agnostic ontology, append-only event storage, and retrieval-time temporal resolution. Our contributions are threefold: 1) We introduce a hybrid entity-event ontology for conversational memory that combines entity-centric and event-centric temporal modeling. Unlike purely entity-focused approaches that struggle with temporal evolution, our ontology represents conversational events as first-class citizens enabling fine-grained temporal reasoning while maintaining entity coherence. 2) To avoid the risk of losing information, we use append-only event storage where facts are anchored to temporally grounded events rather than directly to entities. This preserves the full evolution of information including contradictions and revisions enabling retrieval-time resolution based on temporal validity rather than premature commitment to a single current state. 3) We develop a complementary multi-tool retrieval framework that combines entity linking (ENTITYLOOKUP), structured graph traversal (GRAPHSQL), and hybrid (semantic+lexical) search (SEARCH), all supported by meta-level planning guidance tool (SCHEMAVIEWER). ENTITYLOOKUP provides entity-centered access and canonicalization, enabling resolution of surface mentions to canonical graph identifiers. GRAPHSQL enables structured temporal reasoning through SQL-based graph traversal, supporting fact evolution tracking, temporal ordering via validity intervals, duration calculations, and multi-hop relationship traversal across events. SEARCH enables flexible hybrid retrieval across the entire graph memory to retrieve high-relevance subgraphs especially for open-domain questions.

2 Related Work

Early Approaches to Conversational Memory. Initial efforts to extend conversational reasoning in LLMs relied on larger context windows or retrieval-augmented generation (RAG). The LOCOMO (Maharana et al., 2024) showed inherent limitations in both approaches due to added noise in the context window: while GPT-4-Turbo achieved 51.6% F1 with 128K context vs 35.9% for GPT-3.5-Turbo at 16K, adversarial performance dropped sharply to 15.7% F1 as models attended to irrelevant details. Hence RAG methods only offered modest gains (GPT-3.5 + RAG reaching 43.3% F1 vs 22.4% no retrieval) (Zhang et al., 2024).

First-Generation Memory Systems. Researchers introduced explicit memory management architectures to address these weaknesses. MemGPT (Packer et al., 2023) pioneered an OS-inspired memory hierarchy (26.65% F1 on LOCOMO). ReadAgent (Lee et al., 2024) used paginated "gist memory" effective for narrative QA but achieved only 9.15% F1 on conversations. MemoryBank (Zhong et al., 2023) adopted psychologically motivated updates but struggled with factual retention (5.0% F1). OpenAI's Memory feature performed comparably to GPT-4-Turbo (52.9%), indicating general-purpose memory layers remained insufficient. These systems demonstrated that architectural complexity alone was inadequate without mechanisms for preserving information fidelity and mitigating noise accumulation.

Advanced Memory Architectures. Recent work introduced sophisticated architectures for improved retrieval and reasoning. A-MEM (Xu et al., 2025b) proposed agentic, Zettelkasten-inspired memory with autonomous linking (27.0% F1 single-hop). H-MEM (Sun and Zeng, 2025) adopted hierarchical retrieval, reporting +21-point gains on multi-hop tasks. Mem0 (Chhikara et al., 2025) implemented entity-centric relational graphs achieving 67.1% single-hop and 51.2% multi-hop accuracy with 91% latency reduction; Mem0^g improved temporal reasoning (58.1%) but lacked rich property attributes. Semantic Anchoring (Chatterjee and Agarwal, 2025) demonstrated benefits of linguistic structure for factual stability. Zep builds temporally-aware knowledge graphs to track evolving facts across sessions (Rasmussen et al., 2025). However, it relies heavily on text retrieval tech-

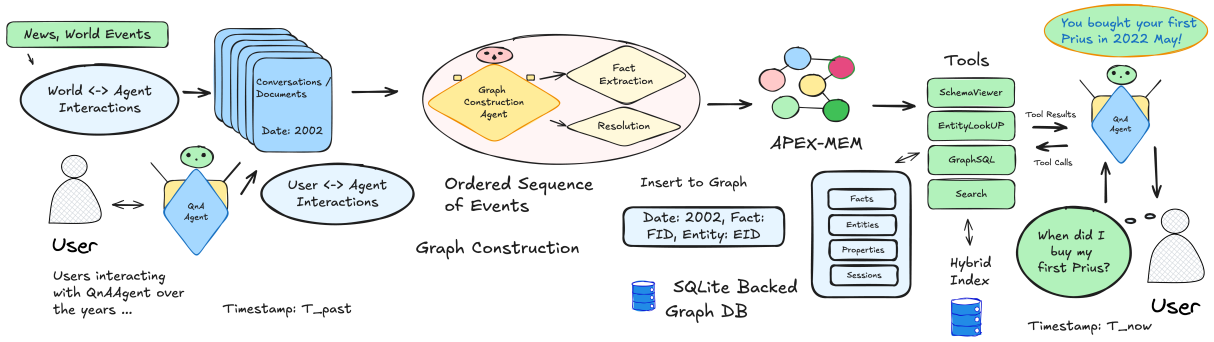

Figure 1: End-to-end pipeline for constructing and querying APEX-MEM Graph, showing data flow from unstructured conversation to GraphSQL QA Agent.

nique to search context. MIRIX (Wang and Chen, 2025) achieved state-of-the-art 85.4% accuracy using six specialized memory stores with multi-agent routing, though its complexity and eager-update strategy risked losing nuanced temporal information. These systems show benefits of structured memory and bring up trade-offs in complexity, expressiveness, and temporal coherence preservation.

3 APEX-MEM Graph Construction

APEX-MEM is a directed property graph $G = (V, E, \Pi, \Lambda)$ where: V is the set of nodes (entities, events, facts, etc.), $E \subseteq V \times V$ is the set of labeled, typed edges, $\Pi : V \cup E \to \mathcal{P}(K \times S)$ maps each element to a set of key–value property pairs, $\Lambda : V \cup E \to T$ assigns a type (from the ontology) to every node and edge.

For every source document ($d_i \in D$) we induce a sub-graph (g_i). Merging is performed incrementally using a *soft-canonicalization* function:

$$G^{(t+1)}; \leftarrow; \mathrm{Merge}\big(G^{(t)}, g_t\big) \qquad \mathrm{Merge}(G, g) := (V \cup V_g, ; E \cup E_g, ; \ldots) \tag{1}$$

where candidate entity nodes and properties are fused when corresponding criteria for Entity and Property Resolution is met. Property graphs allow rich key-value representation of structured information about the entity and how it is referred in the conversation, in contrast to Mem0 where conversation is not ontologized.

3.1 Ontology

We propose a temporal event ontology for conversational memory systems grounded in world knowledge representation. The ontology defines a comprehensive entity type hierarchy $\tau \in \mathcal{T}$ comprising 35 classes analogous to YAGO taxonomies (Suchanek et al., 2024), spanning agents (Person, Organization, Corporation), living organisms (Animal, Plant, Taxonomy), spatiotemporal constructs (Place, Event, Time), physical artifacts (Product, Device, Vehicle), digital objects (Software, Dataset, Service), information resources (CreativeWork, Document, Message), financial instruments (Stock, Contract), health concepts (Food, Medication, Disease), and abstract entities (Topic, Metric, Task). Each entity $e = (n, \tau, \rho, \mathrm{id})$ is characterized by name n, type τ, conversational role $\rho \in \mathcal{R}$ (Speaker, Listener, Agent, Mentioned), and optional external identifier. Facts are represented as temporally-grounded subject-property-value assertions $f = (s, p, v, \delta, [t_{\text{from}}, t_{\text{to}}], c, \mathcal{E})$ where s denotes the subject entity, p the property, v the value with data type δ, $[t_{\text{from}}, t_{\text{to}}]$ the temporal validity interval, $c \in [0, 1]$ the confidence score, and $\mathcal{E}$ the evidence set. All facts are anchored to conversational events $\varepsilon = (\text{type}, T, L, P, F, \mathcal{E}_\varepsilon)$ where T represents the event timestamp, L the location, $P \subseteq \{e_i\}$ the participant set, $F \subseteq \{f_j\}$ the associated facts, and $\mathcal{E}_\varepsilon$ the supporting textual evidence, enabling temporal reasoning over evolving world knowledge extracted from dialogue. This ontology allows flexible property attachment beyond strict schema constraints, enabling domain adaptation while maintaining structural consistency.

3.2 Entity and Property Resolution

To ensure robust entity and property canonicalization across conversational turns we resolve entities and properties in the conversation text against existing set of entities. This maintains provenance through confidence scoring and explicit rationale generation. We use a retrieval-augmented generation approach combining dense semantic search over known entities with structured LLM reasoning. Given a mention

m in conversational text, the entity resolver retrieves top-k candidates $C = \{c_1, \ldots, c_k\}$ from a dense vector index using cosine similarity, where each candidate $c_i = (\text{id}_i, \text{text}_i, s_i)$ includes an identifier, textual representation, and similarity score $s_i \in [0, 1]$. A structured LLM then evaluates candidates against the mention and contextual information to produce a resolution decision $d \in \{\text{choose\_existing}, \text{propose\_new}, \text{none}\}$ with output $o = (d, \text{id}, n_{\text{norm}}, \tau, A, c, r)$ where id is the resolved or newly generated entity identifier, n_{norm} the normalized name, τ the entity type, A a set of aliases, $c \in [0, 1]$ the confidence score, and r the rationale. Property resolution follows an analogous pipeline, additionally normalizing property names to snake_case and inferring data types $\delta \in \{\text{str}, \text{int}, \text{float}, \text{bool}, \text{date}, \text{datetime}, \text{enum}, \text{url}, \text{list}\}$.

3.3 Fact Extraction

Structured fact extraction is performed through few-shot prompted large language models with schema-constrained generation. Given a conversational turn $u = (s, l, \text{text}, t_{\text{anchor}}, \text{ctx})$ where s denotes the speaker, l the listener, t_{anchor} the anchor timestamp (the conversation temporal timestamp), and ctx the recent conversational context, the extractor generates a structured event representation $\varepsilon = (\text{type}, T, L, P, F, \mathcal{E})$ conforming to a predefined schema, defined in the Ontology. The extraction prompt comprises hand-crafted high-quality few-shot exemplars demonstrating comprehensive fact extraction across diverse conversational patterns, including factual assertions, numerical data, emotional states, environmental conditions, and personal attributes. The LLM output is validated, with strict type specifications: participants must conform to $\tau \in \mathcal{T}$ and $\rho \in \mathcal{R}$, facts must specify data types $\delta \in \Delta$, and temporal expressions are normalized to ISO 8601 format relative to t_{anchor}. All facts include confidence scores $c \in [0, 1]$ and supporting evidence $\mathcal{E} = \{(e_{\text{text}}, e_{\text{turn}}, e_{\text{span}})\}$ linking assertions to source utterances. This structured approach ensures consistent, machine-readable knowledge extraction while preserving provenance and enabling downstream temporal reasoning over evolving conversational state.

4 Graph Agents

APEX-MEM Graph QnA agent is a ReAct-style agent (Yao et al., 2022) operating over a temporal property-graph database $\mathcal{G}$ instantiated by the SQLite schema in `DB_SCHEMA` (tables `entities`, `properties`, `facts`, `events`, `evidence`, `turns` and their lexical search views). Given a natural-language question x and interaction history h_t, the agent implements a policy π_θ that, at step t, generates a reasoning trace r_t and selects an action $a_t \in \mathcal{A}$,

$$(r_t, a_t) \sim \pi_\theta(\cdot \mid x, h_t).$$

Actions are either (i) tool invocations $a_t = (T_t, z_t)$, where $T_t \in \{\text{SCHEMAVIEWER}, \text{ENTITYLOOKUP}, \text{GRAPHSQL}, \text{SEARCH}\}$ and z_t are structured arguments conforming to the corresponding tool inputs, or (ii) a special ANSWER action that emits a final answer y. The agent resolves temporal references in each turn into dates and date-ranges which is used when a tool is invoked. Tool outputs o_t are appended to the history, $h_{t+1} = h_t \cup \{(r_t, a_t, o_t)\}$, enabling multi-step reasoning that interleaves natural-language planning with structured access to entities, properties, events, evidence, and conversation turns stored in $\mathcal{G}$ via the tools exposed by `QnAAgent`.

4.1 SCHEMAVIEWER

The SCHEMAVIEWER tool is a schema- and strategy-inspection operator

$$T_{\text{schema}} : \{0, 1\}^2 \to \mathcal{S}, \tag{2}$$

which, given boolean flags $(b_{\text{ex}}, b_{\text{guide}})$ indicating whether to include examples and a usage guide, returns a structured schema view $s \in \mathcal{S}$. The agent uses SCHEMAVIEWER both to inspect the relational schema of the graph-backed SQLite database and to obtain query and tool-usage recommendations (e.g., when to call entity, event, instructions on how to do temporal reasoning using SQL), effectively acting as a meta-level planner aid for Graph QnA.

4.2 ENTITYLOOKUP

The ENTITYLOOKUP tool is a entity-retrieval operator

$$T_{\text{ent}} : \mathcal{Q} \times \mathbb{N} \to \mathcal{D}_{\text{ent}}, \tag{3}$$

mapping a free-text query $q \in \mathcal{Q}$ and a top-k budget K to a ranked list of entity documents $(d_1, \ldots, d_K) \in \mathcal{D}_{\text{ent}}$. It first retrieves candidate entity ids from a hybrid index (combining dense and lexical search), then queries the underlying `GraphDB` over `entities`,

`facts`, `properties`, `events`, and `evidence` to construct, for each entity id e, a document $d = (id, name, type, latest, anchors, last\_anchor, facts)$, where *latest* and *facts* are markdown tables summarizing recent property values and *anchors/last_anchor* expose temporal context via *events.anchor_datetime*. The agent uses ENTITY-LOOKUP to canonicalize surface forms to graph ids and to retrieve time-aware fact snapshots that ground downstream reasoning and SQL queries.

4.3 GRAPHSQL

The GRAPHSQL tool is a read-only SQL interface

$$T_{\text{sql}} : \mathcal{S}_{\text{sql}} \times \mathcal{P}_{\text{sql}} \to \mathcal{R}_{\text{sql}},$$

where $\mathcal{S}_{\text{sql}}$ is the set of safe SQLite *SELECT* (or *WITH* ... *SELECT*) statements over the whitelisted tables: `events`, `facts`, `evidence`, `entities`, `event_participants`, `properties`, `turns`. $\mathcal{P}_{\text{sql}}$ is the space of named-parameter maps, and $\mathcal{R}_{\text{sql}}$ is the space of result tables. The tool first validates the statement, enforcing a single read-only statement and forbidding Updates, or DDL then executes it against the `GraphDB` and returns sql outputs wrapped as markdown. GRAPHSQL is invoked when the agent needs precise graph reasoning (e.g., joining entities and events), aggregations, mathematical, or temporal computations based on `anchor_datetime`, going beyond what pure retrieval can provide.

4.4 SEARCH

The SEARCH tool conceptually exposes a hybrid retrieval layer over the graph, its relational views, and semantic indices, mapping a query q to a composite context

$$T_{\text{search}} : \mathcal{Q} \to \mathcal{C}, \quad T_{\text{search}}(q) = (E_q, P_q, \mathcal{V}_q, \mathcal{T}_q),$$

where E_q are candidate entities, P_q are candidate properties, $\mathcal{V}_q$ are candidate events/evidence, and $\mathcal{T}_q$ are relevant conversation turns. From the agent's perspective, SEARCH provides a unified, hybrid graph–entity–property–SQL and semantic search capability: it retrieves a high-relevance subgraph around the question, which can then be further filtered or aggregated via GRAPHSQL before the ReAct policy emits the final answer.

5 APEX-MEM Online Construction

For very-long conversations, where the number of documents is a magnitude higher $> 10^3$, and its infeasible and unnecessary to construct a complete APEX-MEM Graph offline, as significant set of conversations or documents are irrelevant to expected user questions. For such cases, we construct APEX-MEM online, where, given a set of Documents (D) and input question (Q), we determine document relevance using semantic and lexical search, and limit Graph construction to those subset of temporally-ordered Documents D_{rel}, with $Relevance(d_i|Q) > \Theta_{rel}$.

6 Experiments and Analysis

We evaluate APEX-MEM on the following different datasets that require diverse reasoning, signal-to-noise ratio, and are challenging to LLMs. We try to answer the following research questions: RQ1. How does APEX-MEM compare to existing state-of-the-art for Memory-based Tasks? RQ2. How important is each tool for APEX-MEM-based QnAAgent ? RQ3. Does APEX-MEM improve QA Agent's ability to handle complex scenarios compared to other Deep research agents? RQ4. How well does APEX-MEM generalize?

6.1 Datasets

LOCOMO (Long-term Conversational Memory) is a benchmark for evaluating agent memory over extended multi-session dialogues spanning weeks-long interactions with evolving user preferences, personal facts, and temporal events (Maharana et al., 2024). Its evaluation targets long-term recall, relevance discrimination, and cross-session consistency. Questions are categorized as single-hop, multi-hop, temporal, open-domain, and adversarial (unanswerable). Following Chhikara et al. (2025), we use LLM-as-a-Judge to assess factual accuracy, relevance, completeness, and contextual appropriateness of generated answers against ground truth.

LongMemEval examines LLM ability to process and reason over extremely long inputs including multi-document collections, extended narratives, and dense conversational histories (Wu et al., 2024). The benchmark emphasizes context-length generalization, factual recall, cross-episode reasoning, and robustness under large-scale sequences, providing a standardized framework for evaluating long-context architectures. We measure performance using the recommended LLM-as-a-Judge for answer quality and factuality scores.

SealQA-Hard challenges search-augmented LLMs on fact-seeking questions where web

Method			Single-Hop	Multi-Hop	Temporal	Open-Domain	Adversarial	Overall	w/o Adv.
	Graph	**QA Agent**							
APEX-MEM	APEX-MEM	Claude 4.5 Haiku	85.46%	84.74%	79.17%	89.18%	**87.22%**	84.92%	84.25%
	APEX-MEM	Claude 4.5 Sonnet	**89.36%**	**86.92%**	**90.63%**	87.75%	86.10%	88.41%	89.08%
	APEX-MEM	Claude 3.5 Sonnet	87.58%	84.74%	87.50%	88.94%	86.10%	86.90%	87.13%
	APEX-MEM	GPT5	89.88%	86.29%	**90.63%**	**91.68%**	86.77%	**88.88%**	**89.49%**
	APEX-MEM	GPT4o	88.47%	85.46%	83.49%	86.46%	84.98%	86.35%	86.75%
	Retrieval	**Agent**							
LOCOMO Baselines	SimpleSearch	GPT5	76.60%	71.96%	72.92%	83.95%	72.65%	77.90%	
	SimpleSearch	GPT4	46.10%	51.40%	37.50%	73.60%	57.17%	60.67%	
	SimpleSearch	Claude 4.5 Sonnet	71.99%	73.52%	73.96%	81.33%	74.22%	76.79%	
	SimpleSearch	Claude 3.5 Sonnet	55.32%	61.99%	61.46%	79.55%	72.87%	70.90%	
MemInsight	MemInsight + Attribute Retrieval	Claude 3.5 Sonnet	74.82%	74.45%	72.92%	80.02%	78.92%	77.79%	
	MemInsight + RAG Baseline	Claude 3.5 Sonnet	79.43%	82.55%	77.08%	75.15%	79.15%	77.95%	
	MemInsight	LLamaV3	76.95%	81.31%	80.21%	76.93%	78.92%	78.25%	
	MemInsight	Mistral	74.47%	82.55%	81.25%	78.60%	81.17%	79.36%	
	Memory Agent								
Memory	OpenAI		63.79%	42.92%	62.29%	21.71%	N/A	52.90%	52.90%
	AMEM		39.79%	18.85%	54.05%	49.91%	N/A	48.38%	48.38%
	Zep		61.70%	41.35%	76.60%	49.31%	N/A	75.14%	75.14%
	MemGPT/LangMem		62.23%	47.92%	71.12%	23.43%	N/A	58.10%	58.10%
	Memory-R1		59.83%	53.01%	68.78%	51.55%	N/A	N/A	N/A
	Mem0		65.71%	47.19%	75.71%	58.13%	N/A	68.44%	68.44%
	Mnemosyne		62.78%	49.53%	60.42%	53.03%	N/A	N/A	N/A
	Nemori		84.9%	75.1%	77.60%	51.0%	N/A	79.4%	79.4%
	MIRIX		85.11%	83.70%	65.62%	88.39%	N/A	85.38%	85.38%
Full Context	**QA Agent**								
	GPT4o		88.53%	77.70%	71.88%	92.70%	N/A	87.52%	87.52%

Table 1: LOCOMO Category Type Evaluation Results

Model	Fact Extraction	Schema Coverage	Entity/Prop. Resolution
GPT4o	94.2%	75.7%	98.1%
Claude Sonnet 4.5	97.3%	91.1%	98.2%
Claude Haiku 4.5	95.8%	90.3%	95.4%
Qwen3-14B	95.4%	88.9%	92.5%

Table 2: APEX-MEM Construction metrics for fact extraction (precision of extracted facts), schema coverage (plausible properties covered), and entity resolution (detecting and linking to proper entity). We measure these metrics with 500 Random turns from LoCoMo and LongMemEval. GPT5 is used as the judge.

search yields conflicting or noisy results. We use SEAL-HARD to assess factual accuracy and reasoning on questions where chat models (e.g., GPT-4.1) achieve near-zero accuracy. SEALQA presents long-context, multi-document "needle-in-a-haystack" scenarios with 30 web-retrieved documents containing 1/2 gold documents at unknown positions. To align with memory benchmarks, we order documents by published time as observation time, processing each as an agent-world interaction. Each question-search pair constitutes a separate session. Performance is measured using LLM-as-a-Judge for answer quality and factuality.

6.2 Experimental Setup

For APEX-MEM construction task, refer to Table 2, for how different LLMs perform at Fact Extraction and Entity Resolution. We construct APEX-MEM using Claude Sonnet 4.5 for Fact Extraction and Claude Haiku 4.5 for Entity and Property Resolution, to balance cost versus task performance. We test different LLMs (Claude, GPT) for QnA Agents. All Tools are used with a max limit of 40 for ReACT tool invocations. We adopt APEX-MEM Online for LongMemEval and SealQA, extracting Entities and Facts from conversations which are marked relevant with $\Theta_{rel} > 0.2$. For LOCOMO, we construct a APEX-MEM for all input sessions in respective conversations. We re-implement LOCOMO baselines to measure the impact of new versions of Claude and GPTs. For LongMemEval we implement a stronger Search baseline of a Expanded Sessions that include top-5 relevant Sessions. For all other benchmarks we used reported numbers from (Salama et al., 2025; Pham et al., 2025; Wang and Chen, 2025). To make the results replicable, we set the temperature to 0 wherever applicable. For each LLM-as-a-Judge, we report mean of 3 trials, with $< \pm 1$ standard deviation.

Method		Single-Hop	Multi-Hop	Temporal	Open-Domain	Adversarial	Overall
Graph	**QA Agent: Claude 4.5 Haiku w/ Tools**						
APEX-MEM	SchemaViewer, EntityLookUp	80.85%	76.64%	72.92%	76.34%	77.80%	77.19%
	+ GraphSQL	80.78%	79.75%	82.29%	78.00%	81.16%	79.45%
	+ Search	85.46%	84.74%	79.17%	89.18%	87.22%	87.00%

Table 3: APEX-MEM Ablations of different tools

6.3 Results

RQ1: How does APEX-MEM compare to existing state-of-the-art for memory-based Tasks? APEX-MEM achieves state-of-the-art performance across multiple conversational memory benchmarks, substantially outperforming previous systems. On the LOCOMO benchmark, APEX-MEM with GPT5 achieves 88.88% overall accuracy (Table 1), surpassing the previous best system MIRIX at 85.38% by 3.50 percentage points. APEX-MEM demonstrates strong performance across all question categories: 89.88% on single-hop, 86.29% on multi-hop reasoning, 90.63% on temporal queries, 91.68% on open-domain, and 86.77% on adversarial examples. With GPT4o as the QnA agent, APEX-MEM achieves 86.35% overall, demonstrating that the architecture generalizes across LLM backends, though we observe higher error rates for GPT4o in SQLite query generation and tool usage. On the LongMemEval, APEX-MEM with Claude 4.5 Sonnet achieves 86.2% overall score (Table 4), improving over the strongest baseline Nemori (Nan et al., 2025) at 74.6% by 11.6 percentage points and session-aware RAG baselines at 72.5% by 13.7 points. These results demonstrate that APEX-MEM provides a robust foundation for long-term conversational memory.

RQ2: How important is each tool for APEX-MEM-based QnAAgent? The ablation study in Table 3 reveals that each tool component contributes significantly to APEX-MEM's performance. Using only SchemaViewer and EntityLookUp tools, the system achieves 77.19% overall accuracy on LOCOMO. Adding GraphSQL capabilities improves performance to 79.45%, representing a 2.26% gain, with particularly strong improvements on multi-hop reasoning (76.64% to 79.75%) and temporal queries (72.92% to 82.29%). The addition of the Search tool further boosts overall accuracy to 87%, a 7.55 point improvement, with substantial gains across all categories including single-hop (80.78% to 85.46%), multi-hop (79.75% to 84.74%), open-domain (78.00% to 89.18%), and adversarial questions (81.16% to 87.22%). This demonstrates that the combination of entity linking, structured graph traversal via SQL, and hybrid semantic search is essential for achieving optimal performance in conversational memory tasks. In the appendix, Table 6 contains analysis on tool distribution for different ablation methods, and Figure 2 for task v/s accuracy v/s # of tool calls. GraphSQL and Search tools complement each other to enhance both quality and efficiency: while GraphSQL-only systems require 3.3x more tool calls (27,282 vs 8,260) to achieve 79.45% accuracy, the hybrid approach leverages Search for rapid context retrieval and GraphSQL for precise structured reasoning, achieving superior 87% accuracy with balanced tool usage. GraphSQL is a complex structured query generation task, as we are using SQLite to model graph-queries. In future we will evaluate the impact of Graph databases.

RQ3: Does APEX-MEM improve QA Agent's ability to handle complex scenarios compared to other Deep research agents? APEX-MEM demonstrates superior performance on complex, multi-document reasoning tasks compared to state-of-the-art research agents. On the SealQA-Hard benchmark (Table 5), which evaluates search-augmented LLMs on challenging fact-seeking questions with conflicting and noisy web search results, APEX-MEM with GPT5 achieves 40.15% accuracy, substantially outperforming baseline agents including O3 at 34.6%, DeepSeek-R1 at 15.4%, GPT4o at 15%, and O4-Mini-HIGH at 12%. This 5.55 percentage point improvement over the strongest baseline demonstrates APEX-MEM's ability to effectively resolve contradictions and filter noise through its append-only property graph architecture and retrieval-time resolution strategy. The system's ontology-grounded entity linking and GraphSQL traversal enable more precise identification of relevant information across multiple conflicting sources, a critical capability for real-world applications where information quality varies significantly. See Table 8 for GraphSQL examples.

RQ4: How well does APEX-MEM generalize? APEX-MEM generalizes across diverse rea-

Method			Overall Score
APEX-MEM	**Graph**	**QA Agent**	
	APEX-MEM Online	Claude 4.5 Haiku	82.8%
		Claude 4.5 Sonnet	**86.2%**
		Claude 4 Sonnet	81.0%
		GPT5	85.2%
		GPT4o	75.0%
Baseline	**Retrieval**	**Agent**	
	Full-Context	GPT4o	60.2%
	Full-Context	Claude 4.5 Sonnet	62.2%
	Full-Context + Chain-of-Note	Claude 4.5 Sonnet	63.9%
	SimpleSearch (K=V+fact), top-5 + expanded Sessions	Claude 4.5 Sonnet	72.5%
LongMemEval	SimpleSearch (K=V+fact)	Mistral-Nemo-Instruct-2407	66.6%
Memory	**MemoryAgent**		
	Zep		71.2%
	Mem0		71.3%
	A-Mem		59.3%
	Nemori		74.6%

Table 4: LongMemEval Evaluation Results

Method		Acc.
APEX-MEM (Online)	QnAAgent	
	Claude 4 Sonnet	28.9%
	Claude 4.5 Sonnet	35.2%
	GPT5	**40.1**%
	GPT4o	19.0%
Baselines w/ Web-Search Tool		
AGENT	GPT4o	15%
	GPT5	38.6%
	O4-Mini-HIGH	12%
	O3	34.6%
	QWEN3-235B	11.4%
	DeepSeek-R1	15.4%

Table 5: SealQA Evaluation Results

soning types and task categories. On LOCOMO (Table 1), the system maintains consistently high performance across different question types with less than 5 percentage points variation. This contrasts sharply with previous systems such as MIRIX (20% drop in temporal accuracy 65.62% temporal) and Mem0 (10% drop in single-hop and 18% drop in multi-hop), which show significant performance disparities across question types. Furthermore, APEX-MEM's strong results on both LOCOMO (88.88%), LongMemEval (86.2%), and SealQA-Hard (40.15%) demonstrate effective generalization across different benchmarks, conversation lengths, and information quality conditions, validating the domain-agnostic nature of the ontology-supported property graph architecture.

7 Conclusion

We have introduced APEX-MEM, an ontology-supported property graph architecture for long-term conversational memory that achieves state-of-the-art performance across multiple challenging benchmarks. By combining append-only semantics at construction time with retrieval-time resolution through a graph agent equipped with entity linking, GraphSQL traversal, and hybrid search capabilities, APEX-MEM addresses fundamental limitations in existing memory systems. Our approach achieves 88.88% accuracy on LOCOMO's Question Answering task, surpassing the previous best system MIRIX by 3.50 percentage points, and 86.2% on LongMemEval, improving over strong session-aware RAG baselines by 13.7 points. These results demonstrate that property graphs provide a robust foundation for capturing rich semantic information while enabling efficient retrieval and intelligent conflict resolution in conversational contexts.

Our evaluation focuses on conversational Question Answering tasks across LOCOMO-QA, LongMemEval, and SealQA-Hard. APEX-MEM is architecturally designed as a retrieval-augmented QA system optimized for query-driven fact selection and temporal resolution. Extending to Event Summarization and Multimodal Dialog Generation—which require generative narrative synthesis and different evaluation metrics—represents valuable future work that builds on our append-only temporal memory foundation.

Despite these advances, significant opportunities remain for future work. First, while APEX-MEM substantially outperforms existing systems, performance gaps persist in SealQA-Hard (40.15% accuracy), where noisy, conflicting multi-document scenarios continue to challenge even our best mod-

els. Improving fact extraction to represent more complete entity information—including better handling of implicit relationships, temporal nuances, and contextual dependencies—will be critical for closing this gap. Second, our current graph agent requires multiple tool invocations to converge on solutions, with the ablation study showing that all three tool types (EntityLookUp, GraphSQL, and Search) contribute meaningfully to performance. Future work will focus on developing more efficient query planning strategies and learned retrieval policies that reduce the number of agent tool calls while maintaining or improving answer quality. Third, extending the ontology to capture domain-specific knowledge structures and exploring automated ontology refinement from conversational data could further enhance the system's ability to represent and reason over specialized knowledge domains. Finally, investigating the integration of APEX-MEM with emerging long-context models and exploring hybrid architectures that combine parametric memory with our structured external memory approach represent promising directions for achieving human-level performance across all conversational memory tasks.

Limitations

While APEX-MEM demonstrates strong performance across multiple benchmarks and reasoning tasks, several limitations remain and highlight areas for future exploration.

First, the graph construction process incurs significant computational costs, particularly during entity resolution and property extraction phases. The current implementation relies on large language models for fact extraction, which can be resource-intensive for real-time conversational applications. Future work should explore smaller, more efficient models and optimized algorithms to reduce construction overhead while maintaining graph quality and precision. Additionally, the accuracy of the constructed graph depends heavily on the quality of entity and property resolution—errors or ambiguities in these processes can propagate through the graph structure, potentially affecting downstream retrieval and reasoning performance.

Second, while APEX-MEM employs a domain-agnostic ontology with 35 entity classes, standardizing ontology schemas across different conversational domains remains challenging. The current ontology may not capture all domain-specific nuances, and queries requiring highly specialized knowledge structures may benefit from extended or customized ontologies. Developing standardized ontology frameworks and exploring automated ontology refinement from conversational data could enhance the system's ability to support more precise querying and reasoning across diverse application domains.

Third, APEX-MEM's performance is sensitive to the QnA agent's ability to generate correct SQLite queries and use tools effectively. With GPT4o as the QnA agent, we observed critically high error rates in graph query generation and tool selection, resulting in 86.35% overall accuracy on LOCOMO compared to 88.88% with GPT5. We had to add explicit query generation error examples to the GPT4o prompt to achieve this level. Both Claude Sonnet 3.5 and 4.5 show similar tool-use success rates and comparable performance, suggesting that the architecture works best with models that natively support structured tool interactions. This dependency on the base model's tool-use capabilities represents a practical deployment consideration.

Additionally, APEX-MEM's performance on particularly SealQA-Hard (40.15% accuracy) indicates room for improvement in handling noisy, conflicting multi-document scenarios. The current graph agent also requires multiple tool invocations to converge on solutions, which can impact response latency in interactive settings. Optimizing the agent's reasoning strategy and reducing the number of tool calls needed would improve efficiency without sacrificing accuracy.

Finally, we acknowledge that APEX-MEM's current implementation is limited to text-based interactions. Future work could extend the system to support multimodal inputs, such as images, audio, or video, enabling richer and more comprehensive contextual representations. Integration with emerging long-context models and exploration of hybrid architectures that combine parametric memory with structured external memory also present promising directions for enhancing the system's capabilities.

Ethical Considerations

We have conducted a comprehensive review of all scientific artifacts utilized in this research, including datasets and computational models, to ensure their licenses explicitly permit academic research and publication use. All datasets employed in our

experiments have been properly de-identified to maintain participant anonymity and protect individual privacy.

Our proposed APEX-MEM framework offers significant potential for reducing both the economic and environmental impact associated with LLM enhancement. By minimizing the requirements for extensive data collection and manual annotation processes for Knowledge Graph construction, our approach streamlines model development while providing robust safeguards for user privacy and data protection. This reduction in data collection needs helps mitigate the risk of information leakage during training corpus assembly and reduces the computational resources typically required for model improvement. Moreover, we aim to further reduce LLM inference costs by improving smaller LLMs by using Knowledge Graphs.

Throughout the development and evaluation of this work, generative AI systems were employed in a limited and transparent manner. Specifically, AI assistance was utilized for language refinement, paraphrasing, and grammatical checking during the manuscript preparation process. Additionally, generative AI was used to assist in writing code test cases for system validation and served as an automated judge for evaluation tasks, supplementing human evaluation where appropriate. All core research contributions, experimental design, analysis, and conclusions presented in this paper are the result of human intellectual effort and scientific reasoning.

References

Nick Alonso, Tomás Figliolia, Anthony Ndirango, and Beren Millidge. 2024. Toward conversational agents with context and time sensitive long-term memory. *Preprint*, arXiv:2406.00057.

Adam Byerly and Daniel Khashabi. 2024. Self-consistency falls short! the adverse effects of positional bias on long-context problems. *arXiv preprint arXiv:2411.01101*.

Maitreyi Chatterjee and Devansh Agarwal. 2025. Semantic anchoring in agentic memory: Leveraging linguistic structures for persistent conversational context. *arXiv preprint arXiv:2508.12630*.

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*.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. 2025. Context length alone hurts llm performance despite perfect retrieval. *arXiv preprint arXiv:2510.05381*.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13677–13700, Singapore. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2402.09727*.

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 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. 2025. Nemori: Self-organizing agent memory inspired by cognitive science. *Preprint*, arXiv:2508.03341.

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*.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. 2025. Secom: On memory construction and retrieval for personalized conversational agents. In *International Conference on Representation Learning*, volume 2025, pages 91851–91885.

Thinh Pham, Nguyen Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. 2025. Sealqa: Raising the bar for reasoning in search-augmented language models. *Preprint*, arXiv:2506.01062.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. *Preprint*, arXiv:2503.21760.

Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. 2024. Yago 4.5: A large and clean knowledge base with a rich taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and*

Development in Information Retrieval, SIGIR '24, page 131–140, New York, NY, USA. Association for Computing Machinery.

Haoran Sun and Shaoning Zeng. 2025. Hierarchical memory for high-efficiency long-term reasoning in llm agents. *arXiv preprint arXiv:2507.22925*.

Yu Wang and Xi Chen. 2025. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2504.19413*.

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

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

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*.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*.

A Appendix

A.1 Comparative Analysis: APEX-MEM vs. GraphSQL-Only Retrieval

To demonstrate the value of APEX-MEM's hybrid tool architecture, we compare the full system against a GraphSQL-only ablation that relies exclusively on SQL queries for memory retrieval. This analysis reveals how APEX-MEM's multi-tool approach achieves superior performance through complementary retrieval strategies.

A.1.1 Tool Usage Patterns and Strategic Differences

Table 6 illustrates the contrasting tool usage patterns between APEX-MEM and its GraphSQL-only variant. APEX-MEM employs a balanced retrieval strategy with 8,260 GraphSQL calls, 8,900 Search calls, and 5,160 EntityLookup calls, demonstrating effective utilization of all available tools. In contrast, the GraphSQL-only agent compensates for the absence of hybrid retrieval by invoking GraphSQL 27,282 times—a 3.3× increase over the full system. This dramatic escalation reflects the agent's need to express all reasoning operations through declarative queries, even when alternative retrieval methods might be more efficient.

Notably, while the GraphSQL-only variant increases PropertySearch calls from 3,346 to 5,016 and EntityLookup calls from 5,160 to 5,546. These auxiliary tools remain essential for identifying entity IDs and property names required to construct valid SQL queries, highlighting that even SQL-centric approaches depend on entity resolution capabilities.

A.1.2 Performance Implications

The architectural differences translate directly into performance outcomes. APEX-MEM achieves 87.00% overall accuracy on LOCOMO, outperforming the GraphSQL-only ablation (79.45%) by 7.55 percentage points. This advantage is particularly pronounced in open-domain questions, where APEX-MEM's 89.18% accuracy surpasses the GraphSQL variant by 11.18 points, demonstrating the critical value of the Search tool for retrieving information not easily accessible through structured queries alone.

Interestingly, the GraphSQL-only agent achieves competitive or superior performance on temporal reasoning tasks (82.29% vs. 79.17%), where SQL's native temporal operators and date functions provide natural expressiveness. Table 7 reveals that temporal queries constitute the largest category (6,775 instances) in the GraphSQL-only configuration, followed by aggregate queries (4,586), SELECT queries (1,850), and JOIN operations (430). These statistics underscore SQL's strength in temporal and quantitative reasoning, while highlighting the limitations of a purely query-based approach for diverse conversational memory tasks.

A.1.3 Query Complexity and Reasoning Patterns

Table 8 provides representative examples of the SQL queries generated by both systems. The temporal query example illustrates the sophistication achievable through declarative queries, computing multiple temporal metrics (days, weeks, approximate months) using SQLite's Julian day func-

Tool	Claude 4.5 Haiku	Claude 4.5 Haiku: Entity	Claude 4.5 Haiku: GRAPHSQL	Claude 4.5 Sonnet
ENTITYLOOKUP	5160	17414	5546	4894
SEARCH	8900	1058	0	10544
PROPERTYSEARCH	3346	358	5016	3342
SCHEMAVIEWER	3958	4004	3622	3964
GRAPHSQL	8260	0	27282	7168

Table 6: Tool Call Distribution Across Methods

SQL Category	Claude 4.5 Haiku	Claude 4.5 Haiku: Entity	Claude 4.5 Haiku: GraphSQL	Claude 4.5 Sonnet
SELECT	315	0	1850	88
JOIN	81	0	430	31
AGGREGATE	1417	0	4586	1230
TEMPORAL	2317	0	6775	2235

Table 7: SQL Query Categories by Method

tions. The aggregate query demonstrates set operations over distinct entities, while the JOIN example shows relationship traversal across facts and evidence tables.

However, it is important to note that these query categories are not mutually exclusive—a single conversational question often requires multiple SQL queries of different types to construct a complete answer. APEX-MEM mitigates this complexity by strategically selecting the most appropriate tool for each reasoning step, reducing the total number of tool invocations needed to converge on correct answers. The GraphSQL-only variant, lacking this flexibility, must decompose complex questions into multiple SQL operations, contributing to both increased query volume and reduced overall accuracy.

A.1.4 Implications for System Design

This comparative analysis demonstrates that while GraphSQL provides powerful capabilities for structured reasoning—particularly for temporal and quantitative queries—APEX-MEM's hybrid architecture achieves superior performance by leveraging complementary tool strengths. The full system balances structured queries with entity-based retrieval and semantic search, enabling more efficient and accurate conversational memory access across diverse reasoning requirements.

B Tool Call Efficiency Analysis on LOCOMO

Figure 2 presents a comprehensive analysis of the relationship between tool call frequency and accuracy across different question categories on the LOCOMO dataset, with tool calls capped at 40. This analysis reveals distinct behavioral patterns depending on the agent's available tool set and the underlying language model's capabilities.

B.1 Agent-Specific Behavioral Patterns

The tool call distribution demonstrates that different agent configurations exhibit markedly different behaviors as the number of tool calls increases. APEX-MEM with Claude 4.5 Sonnet achieves the highest efficiency, reaching approximately 84-86% accuracy with just 10 tool calls across most categories and answering 80-90% of questions within the first 10-20 tool calls. The full APEX-MEM system with Claude 4.5 Haiku shows similar patterns but requires slightly more tool calls to converge to comparable accuracy levels.

In contrast, the GraphSQL-only ablation requires significantly more tool calls to achieve competitive performance. This agent must first discover the graph structure through SchemaViewer and EntityLookup operations before constructing effective SQL queries, resulting in a more gradual accuracy improvement curve. However, with sufficient tool calls (approximately 20-30), the GraphSQL-only agent outperforms the EntityLookup-only baseline across most categories, demonstrating that structured queries can eventually compensate for the absence of hybrid retrieval capabilities.

B.2 Adversarial Questions: The Noise Introduction Effect

Adversarial questions exhibit a unique pattern where increased tool calls can introduce noise rather than improve accuracy. At 10 tool calls, Claude 4.5 Sonnet achieves 100% accuracy (albeit with only 1% coverage), while the full Haiku sys-

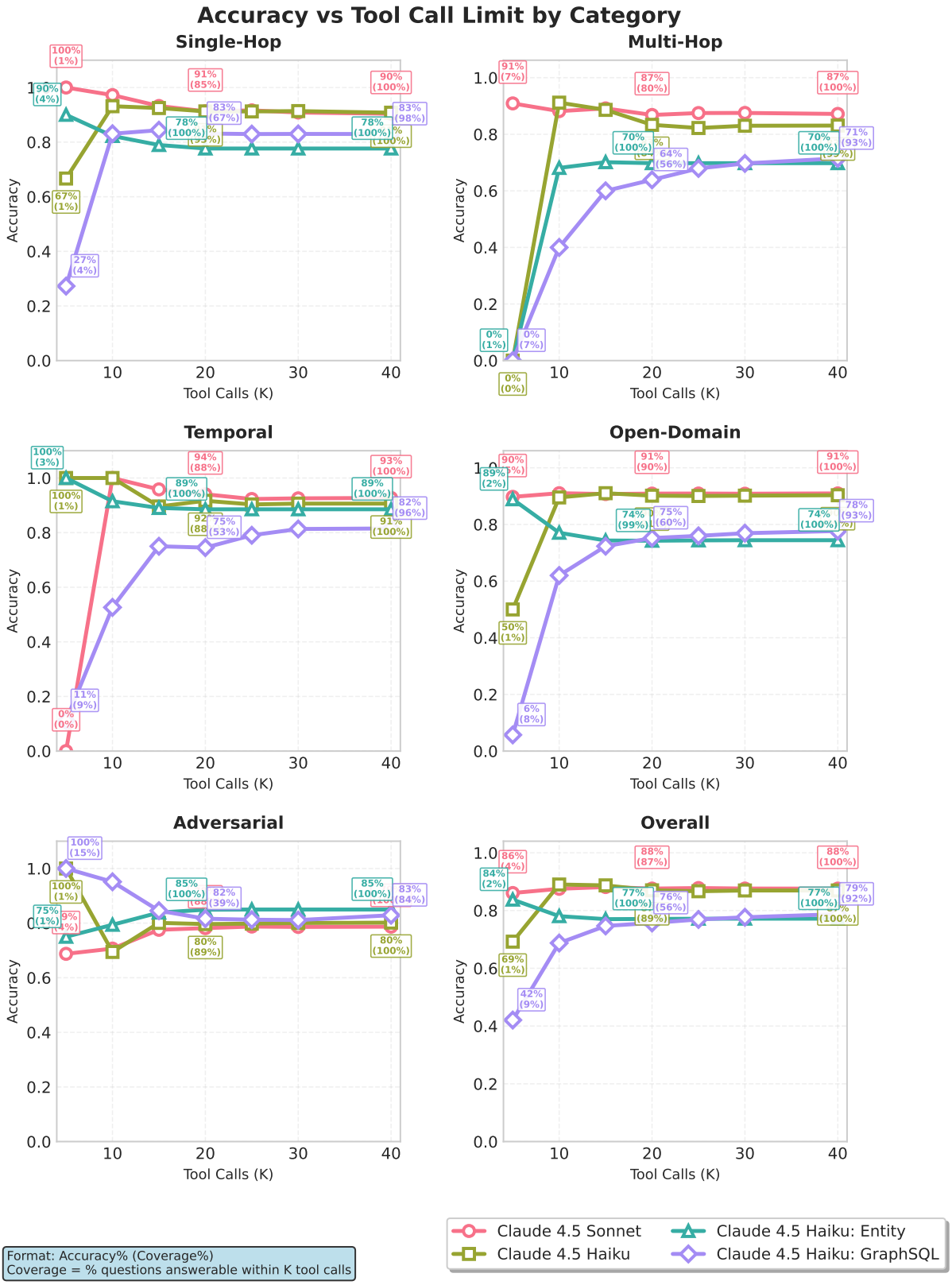

Figure 2: Analysis of Tool Calls v/s Accuracy on LOCOMO Dataset. We cap the max Tool calls at 40.

Category	Example Query
SELECT	`SELECT DISTINCT entity_id, entity_name, entity_type FROM entities WHERE entity_name LIKE '%Anthony%' COLLATE NOCASE`
JOIN	`SELECT f.property_name, f.value_json, f.dtype FROM facts f JOIN evidence e ON e.fact_id = f.id WHERE e.event_id = 518`
AGGREGATE	`SELECT COUNT(DISTINCT device_name) as device_count FROM ( SELECT 'Fitbit Versa 3' as device_name UNION SELECT 'nebulizer machine' as device_name UNION SELECT 'Accu-Chek Aviva Nano' as device_name UNION SELECT 'hearing aids' as device_name )`
TEMPORAL	`SELECT f.value_json as start_date, date(:question_date) as question_date, julianday(:question_date) - julianday(json_extract(f.value_json, '$')) as days_difference, CAST((julianday(:question_date) - julianday(json_extract(f.value_json, '$'))) / 30.44 AS INTEGER) as months_approx, CAST((julianday(:question_date) - julianday(json_extract(f.value_json, '$')))/7 AS INTEGER) as weeks_approx FROM facts f WHERE f.subject_id = :user_id AND f.property_name = :property_id ORDER BY f.created_at DESC LIMIT 1`

Table 8: Sample SQL Queries by Category

tem reaches 75% accuracy. However, as tool calls increase to 20, all agents converge to relatively stable performance levels: Sonnet at 85%, Haiku at 80%, and GraphSQL at 78% accuracy. Beyond 20 tool calls, performance remains largely constant, with all methods eventually converging to 80-85% accuracy at 40 tool calls.

This convergence pattern suggests that for adversarial questions—which are specifically designed to be misleading or contain conflicting information—additional tool invocations may retrieve contradictory evidence that introduces uncertainty into the reasoning process. Most agents effectively reach their performance ceiling at approximately 20 tool calls, after which further retrieval provides diminishing or even slightly negative returns.

B.3 Consistency of Stronger LLM Agents

For non-adversarial categories (single-hop, multi-hop, temporal, and open-domain questions), stronger LLM agents demonstrate remarkable consistency in accuracy regardless of tool call count. APEX-MEM with Claude 4.5 Sonnet maintains high accuracy even with relatively few tool calls, requiring at least 10 tool calls to confidently answer approximately 80% of queries. This efficiency reflects the model's superior reasoning capabilities and its ability to strategically select the most informative tool invocations.

The full APEX-MEM system reaches 87-91% accuracy across most categories with 40 tool calls, while the GraphSQL-only variant plateaus at 79-83% depending on the question type. Notably, the GraphSQL agent shows competitive performance on temporal queries (82% accuracy), where SQL's native temporal operators provide natural expressiveness, but struggles more significantly on open-domain questions (78% maximum), where the absence of the Search tool limits its ability to retrieve relevant contextual information.

B.4 Graph Discovery and Multi-Call Requirements

The GraphSQL-only agent's performance trajectory illustrates the inherent cost of graph discovery in structured query-based approaches. Unlike the full APEX-MEM system, which can leverage EntityLookup and Search tools for rapid information access, the GraphSQL agent must iteratively explore the graph schema, identify relevant entities and properties, and construct appropriate SQL queries. This discovery process manifests as a more gradual accuracy improvement curve, with the agent requiring 20-30 tool calls to reach per-

formance levels that APEX-MEM achieves with 10-15 calls.

Despite this efficiency gap, the GraphSQL-only agent's ability to eventually outperform the EntityLookup-only baseline (which plateaus at approximately 77% overall accuracy) demonstrates the value of structured querying for complex reasoning tasks. The EntityLookup-only approach, while efficient for simple entity retrieval, lacks the compositional reasoning capabilities needed for multi-hop, temporal, and aggregate queries, resulting in consistently lower maximum accuracy across all categories.

These findings underscore the importance of APEX-MEM's hybrid tool architecture, which balances the efficiency of direct entity lookup, the expressiveness of structured queries, and the flexibility of semantic search to achieve superior performance with minimal tool call overhead.

B.5 Additional Future Work

While APEX-MEM achieves state-of-the-art accuracy, the current implementation faces efficiency challenges that warrant future investigation. Our analysis reveals that the system requires multiple tool invocations to converge on solutions, with most agents reaching their performance ceiling at approximately 20 tool calls. The 20-call performance ceiling and diminishing returns beyond this threshold suggest opportunities for intelligent stopping criteria.

improve query planning: 1. Develop reinforcement learning-based query planning that: 1/Predicts optimal tool sequences based on question type, 2/ Learns when to stop retrieval, 3/ Reduces average tool calls from 20-30 to 10-15 while maintaining accuracy **efficient Model Alternatives**: Claude 4.5 Sonnet's superior efficiency (84-86% at 10 calls) demonstrates that stronger reasoning reduces tool call requirements. **improve efficiency in graph construction**: 1/ Fine-tune smaller models (7B-14B parameters) for specific subtasks to achieve Sonnet-level efficiency at lower cost, 2/ Target the 95-97% fact extraction precision and 95-98% entity resolution accuracy, and 3/ Explore specialized models for entity resolution, property extraction, and query planning 3.GraphSQL-only requires 3.3× more calls due to schema discovery overhead, while EntityLookup-only plateaus at 77%. **improve tool usage efficiency**: 1/ Investigate adaptive tool selection strategies that choose optimal tools based on query characteristics, 2/ Develop caching mechanisms for frequently accessed schema patterns to reduce discovery overheads, 3/ Explore parallel tool execution when dependencies allow.

C Cost and Resource Comparison

Table 9 presents a comprehensive token-based cost comparison across all memory methods. Token counts provide a pricing-agnostic, stable comparison metric, as dollar costs vary by model and change frequently. Every memory method has two cost components: (1) **Graph/Memory Construction (GC)**—a one-time cost per conversation, amortized over queries, and (2) **Query Answering (QnA)**—per-query retrieval and generation cost. Graph construction accounts for only 16.6% of APEX-MEM's total cost; the majority is spent on tool access and agentic reasoning loops. Overall, APEX-MEM's per-query token consumption is significantly lower than MIRIX, which performs approximately 8 LLM calls per interaction.

Table 10 decomposes APEX-MEM's per-query token usage by component, revealing that memory retrieval content (26.6%) and tool framing overhead (27.3%) dominate, while graph construction is amortized to only 16.6%.

D GraphSQL Execution Statistics

Table 11 presents GraphSQL execution analysis across different LLM backends. Claude Sonnet 4.5 achieves the highest success rate (97.6%), while GPT-5 shows a higher error rate (6.6%), primarily due to SQLite syntax differences.

The agent successfully recovered from 87% of SQL failures through three mechanisms: (1) **SchemaViewer consultation** (45% of recoveries): re-examining table structure and correcting syntax; (2) **Fallback to EntityLookup** (28%): retrieving structured entity documents when SQL is too complex, then identifying correct properties for structured queries; (3) **Fallback to Search** (14%): using semantic retrieval when graph structure is insufficient. This demonstrates the robustness of the multi-tool architecture: no single tool failure blocks progress, and the agent adapts its retrieval strategy based on tool success.

E Qualitative Case Studies

We present three case studies illustrating APEX-MEM's temporal resolution, multi-hop reasoning, and failure modes.

Method	GC Tok/Conv	GC Calls	Amort. GC/Q	Mem Tok/Q	QnA Tok/Q	Total Tok/Q	Acc (w/o Adv)
MIRIX (est.)	∼15.2M	∼4,704	∼98,750	∼4,500	∼13,500	∼112,000	85.38%
Zep/Graphiti	∼9.4M	∼5,292	∼60,900	2,247	∼3,900	∼64,800	75.14%
Mem0^g (est.)	∼4.9M	∼2,352	∼31,882	3,616	∼5,300	∼37,200	68.44%
APEX-MEM	2.69M	3,717	13,557	8,000‡	∼16,000	∼30,000	84–89%†
Full Context	0	0	0	23,653	∼25,000	∼25,000	87.52%
Mem0 (est.)	∼1.9M	∼882	∼12,409	1,764	∼3,500	∼15,900	68.44%
Nemori (est.)	∼1.0M	∼765	∼6,422	2,745	∼4,500	∼10,900	79.40%

Table 9: Token consumption comparison across memory methods. †Accuracy depends on QnA model. ‡Tunable via top-k retrieval budget. GC = Graph/Memory Construction; Amort. = Amortized over queries per conversation.

Component	Tok/Q	%	Source
Graph constr. (amort.)	13,557	16.6	Extraction, resolution
System prompt	7,854	9.6	Fixed per architecture
Memory retrieval	21,745	26.6	Graph search (top-k)
Agent loop overhead	16,174	19.8	Prior msgs in loop
Tool framing	22,274	27.3	Tool call/response fmt
Total (mean)	**81,604**	**100**	

Table 10: APEX-MEM token decomposition per query.

Metric	Sonnet 4.5	GPT-5	Sonnet (SQL)	Haiku
SQL Executions	3,659	2,163	66,580	4,277
Successful	3,574	2,020	65,900	4,080
SQL Errors	87	143	791	197
Success Rate	**97.6%**	93.4%	**98.8%**	95.4%

Table 11: GraphSQL execution analysis across LLM backends. "Sonnet (SQL)" denotes the GraphSQL-only ablation.

Case 1: Temporal Contradiction Resolution. *Question:* "What is Alice's current favorite restaurant?"

Timeline: Session 1 (2024-01-15): Alice says "I love Italian Garden! Their pasta is the best in town." Session 5 (2024-03-20): Alice says "Italian Garden closed down last month. Now I go to Sakura Sushi every week instead."

APEX-MEM Processing: During construction, the system extracts `(Alice, favorite_restaurant, "Italian Garden", from=2024-01-15)` and later `(Alice, favorite_restaurant, "Sakura Sushi", from=2024-03-20)` **without deleting the previous fact**. At retrieval time, a GraphSQL temporal query returns both facts ordered by timestamp; the agent selects the most recent valid entry ("Sakura Sushi"). This preserves the full temporal evolution with evidence links, enabling follow-up questions like "When did Alice's favorite restaurant change?"

Case 2: Multi-Hop Reasoning Success. *Question:* "What is the title of Bob's manager?"

Processing: The agent performs multi-hop traversal via GraphSQL: Bob → `reports_to` → Sarah Chen → `job_title` → VP of Engineering. The structured graph enables compositional queries that would be difficult with flat text retrieval.

Case 3: Failure Mode—Entity Linking Error. *Question:* "How many times did Bob visit restaurants in Paris last month?"

Root Cause: Entity linking failed to connect restaurant names ("Le Jules Verne") to the Paris location entity from contextual clues ("Eiffel Tower"). The system correctly identified the restaurants as entities but did not resolve the implicit spatial relationship.

Lesson: High-quality entity linking is critical for spatial/relational queries. Integrating external knowledge bases (e.g., Wikidata) for landmark-to-location resolution could address this limitation.

F Append-Only vs. Eager Update Strategies

Table 12 provides indirect evidence supporting APEX-MEM's append-only design by comparing temporal reasoning accuracy across systems with different update strategies.

System	Append-Only?	Temporal	Δ
APEX-MEM	Yes	**90.63%**	—
Mem0	No (consolid.)	75.71%	–14.92
MIRIX	No (state merge)	65.62%	–25.01
Zep	Partial (temp. KG)	76.60%	–14.03

Table 12: LOCOMO temporal accuracy: append-only vs. eager update strategies. Δ is the difference from APEX-MEM in percentage points.

APEX-MEM's +14 to +25 point advantage on temporal queries strongly suggests that preserving full history enables better temporal reasoning. Systems that consolidate or overwrite facts during construction lose the temporal provenance needed to resolve "when did X change?" or "what was Y before Z?" queries, which require access to superseded facts.

G APEX-MEM Construction and Evolution

Table 13 demonstrates how APEX-MEM transforms conversational text into structured graph representations. The examples show the systematic extraction of entities, properties, and typed values from natural language, illustrating the rich semantic structure captured by our ontology-guided approach.

H Graph Structure Visualization

Figure 3 illustrates the layered structure of APEX-MEM's property graph, showing how conversational elements are organized into temporal, entity, and relationship layers. The visualization demonstrates the systematic organization of turns, events, entities, and their properties in a coherent graph structure that enables efficient querying and reasoning.

I Ontological Architecture

Figure 4 presents the complete ontological architecture of APEX-MEM, showing both the structural relationships between data components (sessions, turns, events, facts, entities) and the semantic taxonomy of entity types. This meta-level view illustrates how conversational data flows through the system's layered architecture to create a rich, typed knowledge graph.

The ontological architecture consists of five distinct layers that transform diverse information sources into structured knowledge:

Temporal Layer: The foundation layer captures the chronological structure of information flows from multiple sources. *Sessions* represent complete interaction sequences with unique identifiers and timestamps, encompassing not only user-agent conversations but also news cycles, document streams, or event sequences. *Turns* represent individual information instances within sessions, containing source, recipient, and content. For example, in conversational AI, turns capture user-agent exchanges; in news monitoring, turns represent individual news articles where "World" is the speaker and "LLM Agent" is the listener; in document processing, turns represent document ingestion events. This layer preserves temporal ordering essential for understanding information context and evolution across diverse domains.

Event Layer: This layer extracts meaningful temporal events from information turns across multiple domains. *Events* represent semantically significant occurrences with structured metadata including event type, participants, location, and temporal anchors. Examples include conversational meetings, global news events (elections, natural disasters, policy changes), document publications, or system interactions. *Evidence* provides the textual spans that support event extraction, maintaining traceability between structured representations and original content sources. This design enables APEX-MEM to process heterogeneous information streams—from intimate conversations to global news feeds—using the same architectural principles.

Knowledge Layer: The core semantic layer transforms events into structured knowledge regardless of information source. *Facts* represent subject-property-value assertions extracted from events, with temporal validity intervals and confidence scores. These facts can describe personal

Original Text	Entities	Types	Properties	Values	Data Types
"I went to a LGBTQ support group yesterday and it was so powerful."	Caroline	PERSON	P:attended_event	"LGBTQ support group"	str
			P:experience_ description	"powerful"	str
	LGBTQ support group	EVENT	P:event_date	"2023-05-07"	date
			P:event_type	"support group"	str
"I just signed up for a pottery class yesterday. It's like therapy for me."	Melanie	PERSON	P:activities_ participated	"pottery class"	str
			P:activity_benefit	"therapy for me"	str
			P:emotional_outlet	"pottery helps express emotions"	str
	pottery class	EVENT	P:enrollment_date	"yesterday"	date
	pottery	CREATIVE_ WORK	P:purpose	"self-expression"	str
			P:therapeutic_value	true	bool
"You'd be a great counselor! Your empathy will help people."	Melanie	PERSON	P:career_suggestion	"counselor"	str
			P:personal_qualities	"empathy"	str
			P:predicted_impact	"will help people"	str
	counselor	PERSON	P:required_skills	"empathy"	str
"I'm swamped with the kids & work."	Melanie	PERSON	P:has_children	true	bool
			P:employment_status	"working"	str
			P:emotional_state	"overwhelmed"	enum
			P:relationship_status	"swamped with kids and work"	str
"The transgender stories were so inspiring! I was so thankful."	Caroline	PERSON	P:emotional_state	"thankful"	str
			P:reaction_to_content	"inspiring"	str
	transgender stories	CREATIVE_ WORK	P:content_type	"personal narratives"	str
			P:emotional_impact	"inspiring"	str
	support	TOPIC	P:context	"LGBTQ community"	str

Table 13: Examples of text-to-graph conversion in APEX-MEM. Each conversational utterance is systematically transformed into structured entity-property-value triplets with appropriate semantic types, enabling sophisticated querying and temporal reasoning over conversational knowledge.

preferences from conversations, geopolitical developments from news, or technical specifications from documents. *Properties* define the semantic relationships and attributes that can be asserted about entities, with typed data schemas (string, boolean, date, etc.) that ensure consistency across diverse information domains and enable unified querying.

Entity Layer: This layer manages the canonical representation of all named entities mentioned across information sources. *Entities* serve as the primary nodes in the knowledge graph, with unique identifiers, normalized names, and semantic type classifications. The entity resolution process ensures that mentions of the same entity across different contexts—whether in personal conversations, news articles, or documents—are properly linked and canonicalized, enabling cross-domain knowledge integration.

Semantic Layer: The top layer provides the ontological taxonomy that governs entity classification across all information domains. Entity types include *PERSON* (individuals in conversations, public figures in news), *CREATIVE_WORK* (personal stories, published content), *PLACE* (conversation locations, global regions), *PRODUCT* (personal tools, commercial products), *GROUP* (social circles, organizations, nations), *TOPIC* (discussion subjects, news themes), *EVENT* (personal activities, world events), and *NATURAL_PHENOMENON* (local weather, global climate events). This semantic typing enables sophisticated reasoning and querying that spans from personal context to global knowledge.

The knowledge extraction process (vertical flow) systematically transforms unstructured information from any source through these layers, while the temporal flow (horizontal) preserves chronological relationships essential for understanding information dynamics and evolution over time. This architecture's generality enables APEX-MEM to serve as a unified memory system for diverse AI applications, from personal assistants processing

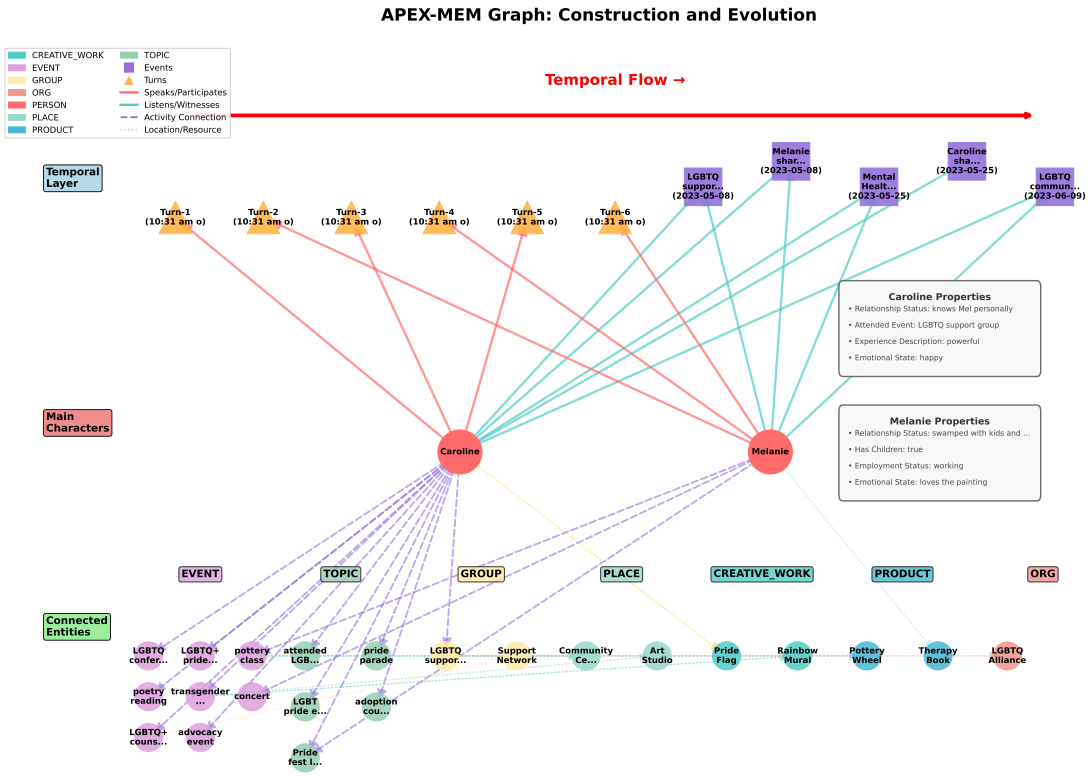

Figure 3: APEX-MEM Graph Structure: The figure demonstrates how conversational turns and events connect to entities through participation relationships, with temporal information preserved and entities organized by semantic type for efficient querying and reasoning.

conversations to news analysis systems processing global events, all using the same ontological foundation.

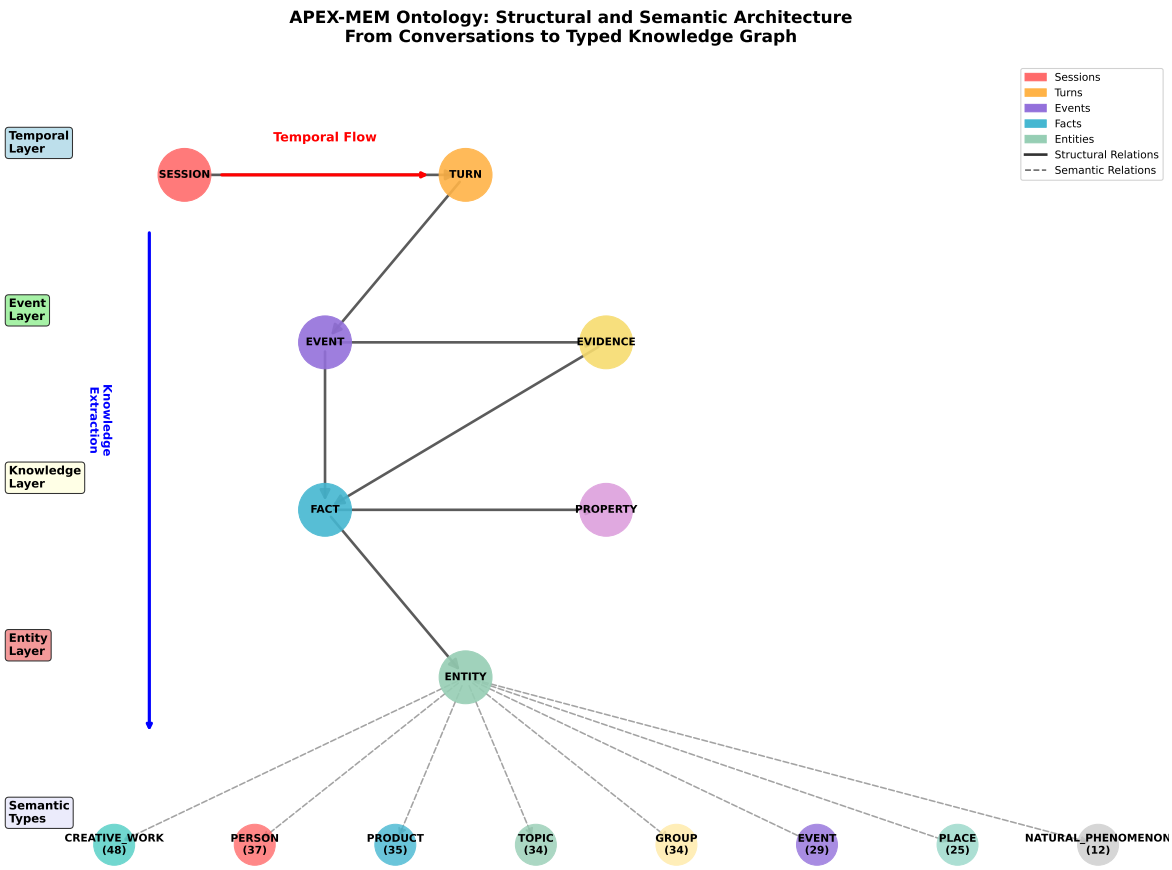

Figure 4: APEX-MEM Ontological Architecture: Complete structural and semantic view showing the flow from conversational sessions through temporal events to typed knowledge extraction. The diagram illustrates both table-level relationships (solid arrows) and semantic type instantiation (dashed arrows), demonstrating how the system transforms unstructured conversations into a rich, ontology-grounded knowledge graph with diverse entity types including PERSON, CREATIVE_WORK, PLACE, PRODUCT, and others.