

Graph-Based Personalized Memory for LLM Agents: Representation, Evolution, Retrieval, and Evaluation

Dac Duy Anh Nguyen

School of ICT

Griffith University

Brisbane, Australia

rodo.nguyen@griffith.edu.au

Zhangchi Qiu

School of ICT

Griffith University

Gold Coast, Australia

z.qiu@griffith.edu.au

Shigeng Chen

School of ICT

Griffith University

Gold Coast, Australia

shigeng.chen@griffithuni.edu.au

Alan Wee-Chung Liew[†]

School of ICT

Griffith University

Gold Coast, Australia

a.liew@griffith.edu.au

Abstract—Large Language Model (LLM) agents are evolving from single-session tools toward long-term personal assistants that must adapt to individual users across tasks, contexts, and interactions. This shift makes memory a core requirement for personalization, since user preferences, goals, constraints, relationships, and past experiences are accumulated gradually and often change over time. Graph-based personalized memory provides a structured way to model such user information through explicit relations, temporal context, and evidence links. Such representations can model not only what an agent remembers about a user but also how memories are connected, revised, and retrieved to support personalized decisions. However, existing work remains fragmented across personalized agents and generic graph memory frameworks, making it difficult to understand the design space as a whole. This survey develops a lifecycle-oriented view of graph-based personalized memory for LLM agents. We organize existing studies around memory representation, memory evolution, memory retrieval, and memory evaluation. We further compare key design choices, discuss current evaluation practices, and open challenges in building reliable long-term personalized agents. This survey aims to clarify how graph-based memory can support adaptive, controllable, and user-centric LLM agents. The repository for this survey is continuously maintained at <https://github.com/icedpanda/awesome-personalized-graph-memory>.

Index Terms—Large Language Models, Agents, Personalized Memory, Graph Memory, Agent Memory

I. INTRODUCTION

Recent advances in large language models (LLMs) have enabled a new class of autonomous agents that can follow complex instructions, plan over multiple steps, use different tools, and carry out long-horizon tasks in dynamic environments [1], [2]. Building on these capabilities, they are beginning to serve as personal assistants that reflect and shape their own behaviors to individual users, such as OpenClaw [3] and Hermes [4]. As agents move toward sustained, user-facing interaction, personalization becomes a central focus, since effective long-term assistance requires not only general reasoning but also an understanding of the user’s goals, preferences, constraints, and prior experiences [5]–[8].

Despite LLMs’ increasingly large context windows, including a user’s full interaction history in every prompt remains impractical [9], [10]. Personalized agents therefore require a persistent memory that can accumulate user-specific knowledge across sessions, update it as preferences and circumstances change, and ultimately guide agent behavior tailored to individual users [5], [11].

Early agent memory systems explored a range of formats to retain user-specific knowledge across sessions, from dialogue-derived records [12], [13] to segment-level or compressed natural-language memories [14], [15]. These designs can record and recall individual facts but typically treat each entry as self-contained, offering limited capacity to capture how facts interrelate, how they evolve over time, or how one event gives rise to the next [8], [10].

To overcome these limitations, graph-based memory has emerged as a structured alternative that organizes user facts, preferences, interaction events, and the relations among them into an interconnected memory space [8], [16]–[18], transitioning agent memory from a passive archive of entries into an explicit model of the user that preserves how personal information is connected and how it changes over time.

This design is a natural fit for personalization because user-specific memory often connects people, places, items, tasks, goals, domains, episodes, and time [5], [19], [20]. Unlike flat logs, summaries, or vector stores, graph-based memory can express this structure directly thanks to its intrinsic ability to link scattered evidence into coherent user personas, to traverse multi-hop context that resolves who and what a request refers to, to trace each profile claim back to its supporting interactions, and to track how preferences and circumstances evolve [11], [18], [21]–[24].

Such capabilities turn the graph from a passive store of user facts into an operational model of the user, although realizing them depends on how the graph is constructed, updated, and queried rather than on the adoption of a graph alone [8], [25].

Although recent surveys have examined personalized agents and agent memory, they use different primary lenses. Surveys of personalized agents frame personalization through agent

[†]Corresponding author.

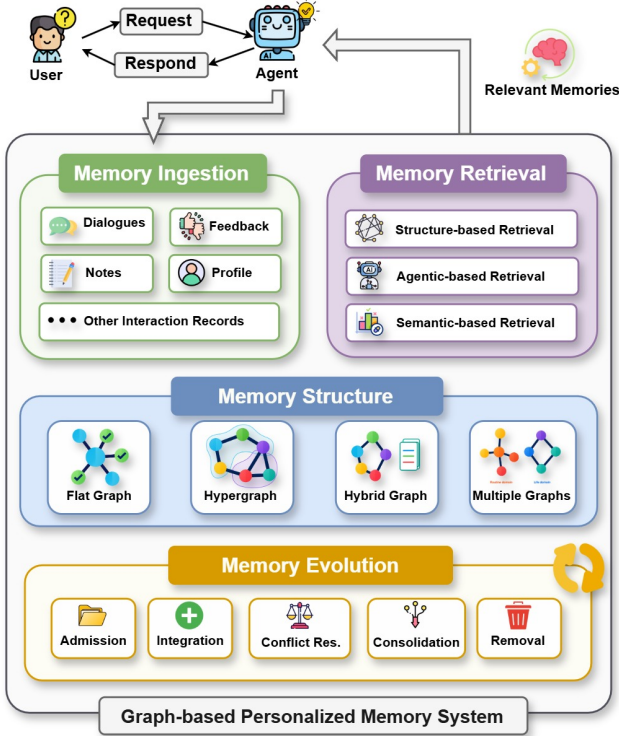


Fig. 1. Overview of a graph-based personalized agent memory system.

capabilities, including profile modeling, memory, planning, and action execution. They treat persistent user memory as one component, rather than as the central object of representation [5]. The survey of graph-based agent memory organizes the field around memory extraction, storage, retrieval, and evolution across broad agent settings. It considers personalization alongside other applications [8]. Broader agent-memory surveys focus on memory-module design and evaluation, modular architectures and benchmarked strategies, or an evolutionary progression from storage to reflection and experience [26]–[28]. Together, these perspectives do not provide a unified account of how graph structures represent, update, retrieve, and evaluate persistent personalized memory for LLM agents.

To address this gap, this survey develops a lifecycle-oriented view of **graph-based personalized memory** for LLM agents. We synthesize how existing systems construct user-centered graph representations, update and consolidate new evidence, retrieve personalized context for response grounding, and evaluate memory quality through current benchmarks and metrics. This perspective connects graph design choices with the practical requirements of long-term personalization and identifies open challenges in personalized memory. As Fig. 1 illustrates, graph-based personalized memory connects memory ingestion, structure, evolution, and retrieval within the personalized agent workflow.

Our contributions are as follows:

- **Taxonomy.** We develop a lifecycle taxonomy that organizes existing studies on graph-based personalized memory around representation, evolution, retrieval, and

evaluation.

- **Evaluation.** We summarize existing benchmarks and metrics and identify gaps in graph quality, user control, lifecycle validity, and realistic personalization.
- **Timely Review.** For each category, we review recent methods and discuss their motivations, graph structures, and user-modeling functions.
- **Future Directions.** We outline several open research directions to guide future research on this promising topic.

II. PRELIMINARIES

A. Graph-based Personalized Memory

Personalization refers to an LLM agent adapting its responses, retrieval, and actions to an individual user. This adaptation depends on **graph-based personalized memory**, which contains user-specific state covering a user’s facts, preferences, goals, constraints, relationships, and experiences. The memory persists across sessions, is stored outside the model parameters, and is organized as an explicit graph. The *memory* aspect separates it from the transient prompt context, and the *personalized* aspect separates it from general knowledge shared across users. We therefore exclude sole generic graph-augmented generation and agent memory unless they support user modeling.

Agent memory has traditionally been stored as raw dialogue logs [12], [13] or natural-language summaries [14], [15], which recall individual entries but leave their relations, supporting evidence, validity, and contradictions implicit. Graph-based memory instead represents user information through *nodes* (the remembered items) and *edges* (the relations among them), making these connections and revisions explicit and queryable. The rest of this survey follows this representation through its lifecycle: how it is constructed (Section III), evolves (Section IV), is retrieved (Section V), and is evaluated (Section VI).

B. Personal Memory Dimensions

Personal memory is the user-specific state an agent preserves beyond the current context window, spanning facts, preferences, goals, constraints, episodes, feedback, emotions, and corrections to earlier claims. Rather than a flat checklist of traits, we characterize it along five *dimensions* that capture what kind of information is remembered, summarized in Table I. The dimensions are not mutually exclusive: a single memory may be semantic in content, bounded in time, relationally anchored, and affectively charged at once. Together, these definitions define what a graph must represent; later sections show how graph structures encode, update, and retrieve it.

III. MEMORY REPRESENTATION

We examine graph memory representation at the element and structure levels. At the *element level*, systems specify node roles, relation types, and lifecycle metadata. At the *structure level*, systems adopt one or more structural designs, including

TABLE I
DIMENSIONS OF PERSONAL MEMORY.

Dimension	What it Captures	Example
Semantic	Stable facts, preferences, goals, or constraints	"I'm vegetarian; learning French"
Temporal	When a memory holds and how it changes	"Joined the gym last month"
Relational	Links to people, places, tasks, or items	"Likes the coffee shop downtown"
Experiential	Past events, outcomes, routines, and habits	"Missed my morning flight last time"
Affective	Emotional valence and intensity (satisfaction, frustration)	"I hated that noisy hotel"

flat, hierarchical, hypergraph, hybrid, and multi-graph designs. These choices together shape evidence traceability, state validity, and retrievable granularity.

A. Memory Elements

We distinguish four functional node roles, which may overlap within one graph. *Evidence nodes* preserve original interaction records, such as turns, episodes [29], or sentences [30]. *Fact nodes* store extracted atomic assertions about the user or related entities [21], [31]. *Abstraction nodes* synthesize repeated observations into higher-level user models, such as personas, scenes [18], or reflections [29]. *Domain-specific nodes* capture entities relevant to particular topics, such as personal interests, jobs, or more specific topics like health care and patient routines [32]. A node may serve more than one role.

Common relation functions include temporal, provenance, associative, and domain-specific links. *Temporal relations* encode event order or progression [22], [29]. *Provenance relations* connect assertions or abstractions to source evidence [10], [21]. *Associative relations* link contextually or semantically related memories [30], [33]. *Domain-specific relations* encode task-relevant semantics [32], [34]. Beyond relations, lifecycle metadata such as timestamps, validity intervals, and invalidation markers [16], [31] can distinguish active user state from historical evidence. Evidence-rich designs preserve source context but increase storage demand; fact- or abstraction-centered designs offer compactness at the risk of propagating extraction errors; domain-specific schemas improve task alignment but reduce generality.

B. Memory Structures

We organize the systems into five representation patterns, as shown in Fig. 2: flat, hierarchical, hypergraph, hybrid, and multiple disjoint graphs. These patterns are not mutually exclusive, and a system may combine several patterns. Flat, hierarchical, and hypergraph patterns describe organization within one graph. Hybrid designs combine graphs with non-graph memory substrates, whereas multiple-graph designs partition memory across separate graph instances.

1) *Flat Graph*: We use *flat graph* to refer to a memory graph that stores user-related facts, preferences, and relations as nodes and typed edges without explicit abstraction layers. Heterogeneous node and edge types do not themselves constitute a hierarchy. UMGR [19] is a domain-oriented heterogeneous graph linking users, items, attributes, and opinions for

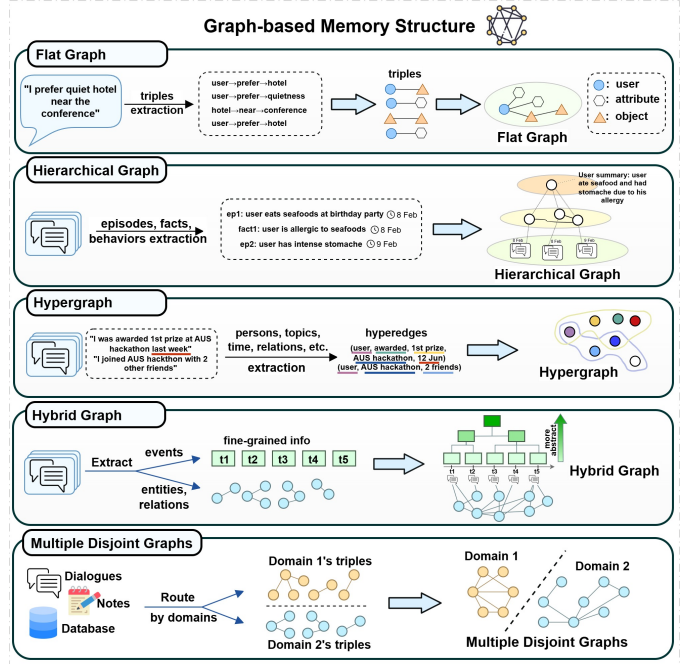


Fig. 2. Representative graph-based memory patterns for personalized LLM agents.

recommendation; KGT [35] stores personalized factual triples representing editable user beliefs; Mem0^g [31] constructs an open-domain entity-relation graph incrementally from conversation. These systems share the absence of explicit abstraction layering while their node schemas and scopes differ. Flat topology stores user-related facts and relations explicitly but does not itself separate source evidence from extracted facts or from higher-level abstractions. Flat graphs can still include provenance and validity metadata.

2) *Hierarchical Graph*: Hierarchical graph memories organize user evidence into multiple layers, making user state accessible at different levels of specificity. Hierarchy need not be reduced to monotonic abstraction; cross-level links can keep evidence reachable from higher-level representations [18], [29]. Hierarchies serve two main functions in the systems considered here. For evidence-to-abstraction organization, GAAMA [29] preserves turns as episode nodes, extracts facts and concepts at an intermediate level, and synthesizes reflections capturing higher-order user patterns; Bi-Mem [18] groups fact-level memories into thematic scenes and distills

them into a global persona, with cross-level links enabling calibration against local evidence. For granularity-preserving organization, GAM [22] organizes active event graphs, topic-level nodes, and archived events into distinct layers, while SGMem [30] decomposes conversations into sessions, turns, and sentences, keeping fine-grained detail accessible alongside coarser summaries and extracted facts.

Hierarchical organization supports traceability across levels between raw evidence and higher-level user representations. Generated abstractions such as personas or reflections may propagate extraction errors or overgeneralize transient evidence. Systems that preserve bidirectional or cross-level links can check higher-level abstractions against lower-level evidence, but this introduces cross-layer consistency requirements that grow with depth.

3) *Hypergraph*: Hypergraphs preserve joint context whose meaning depends on several connected elements. HingeMem [36] groups boundary-triggered dialogue segments into hyperedges indexed by elements such as participants, locations, and times; HyperMem [37] organizes memory into topic, episode, and fact levels, using episode hyperedges to group episodes under the same topic and fact hyperedges to group facts within an episode. Both preserve joint contextual groupings, but HingeMem emphasizes matching a query to an event, whereas HyperMem emphasizes grouping related information across a long conversation before coarse-to-fine (top-down) retrieval.

4) *Hybrid Graph*: Hybrid memory systems distribute the user model across graph topology and a complementary non-graph substrate, such as a tree, summary store, passage store, or evidence buffer. The graph provides relational access among user facts, entities, and events; the non-graph substrate holds long-form interaction evidence, temporal organization, or source text outside the graph. H-Mem [38] combines a temporal-semantic tree that organizes memory fragments from timestamped events to higher-level summaries with an entity graph that records relations and links entities to the fragments that mention them.

Other hybrid designs use a lightweight graph as an index to richer stores. LiCoMemory [10] stores entity-relation pairs in a graph, while session summaries and dialogue chunks carry the content. MemWeaver [39] partitions memory into a structured relation graph, an experience store for repeated interaction patterns, and a passage store for source text. These specialized stores separate relational indexing from evidence storage but require consistency across components.

5) *Multiple Disjoint Graphs*: Multiple-graph designs split user memory across separate graph instances that serve distinct, specialized functions. Evidence for this design is currently limited to DEMENTIA-PLAN [32], which maintains a daily routine graph for time-sensitive care context and a life memory graph for longer-term autobiographical identity. This arrangement separates user states by timescale and purpose across distinct instances. The cited work does not establish whether these graph instances are formally disjoint, and further evidence is needed to assess the generality of this design.

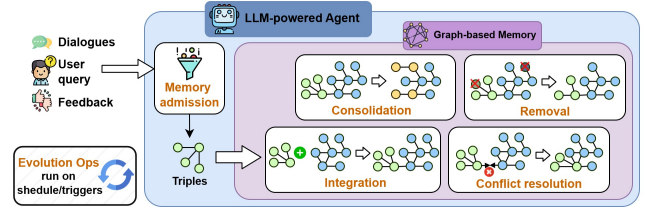


Fig. 3. Evolution of graph-based personalized memory.

Nevertheless, this pattern may be useful for highly specific use cases where memory can be stored and retrieved more efficiently because its location is known in advance.

C. Discussion

Representation determines which user evidence a system retains and which dependencies later lifecycle operations can use. Evidence-rich designs preserve provenance and support correction, while fact- and abstraction-centered designs provide compact user models but may discard context or propagate extraction errors. Hierarchical and multi-store representations also require consistency across levels or substrates. Lifecycle metadata can distinguish an active state from historical evidence, but its presence does not ensure that updates propagate correctly. Thus, representation is best understood as a substrate for memory evolution and retrieval rather than as a guarantee of personalization quality.

IV. MEMORY EVOLUTION

Memory evolution maintains a usable active user model while preserving historical evidence where needed. As new evidence arrives, the system decides whether to admit it, integrate it into the existing structure, resolve conflicts with prior states, consolidate accumulated memory, and remove outdated records. This section examines these five evolution operations (illustrated in Fig. 3).

A. Memory Admission

Admission decides whether incoming evidence becomes persistent memory and at what functional type or granularity. In KGT [35], user feedback is treated as a trigger to add or replace factual triples in its graph. MemORAI [21] applies selective filtering and compression to retain user-persona-relevant episodic content, facts, and preferences while compressing less informative discourse. MemGuard [40] assigns each memory a functional role (e.g., fact, procedural instruction, constraint) at write time and preserves boundaries among functionally distinct memory types. Admission protocol is important as over-admission risks embedding transient claims as durable user state, while under-admission can discard evidence that later interactions require.

B. Memory Integration

After admission, integration attaches evidence to the existing graph structure at a local write scale. Append-and-link preserves provenance to the originating turn, session, or chunk, as in LiCoMemory [10]. To reduce redundancy when

admitting memory, Mem0^s [31] reuses semantically matched nodes and invalidates obsolete edges, while HingeMem [36] consolidates near-duplicate boundary hyperedges. GAM [22] uses buffered attachment, writing new utterances into an active event graph before later consolidation into its global graph. This ensures GAM only consolidates semantically complete units and reduces semantic drift in the main graph.

C. Conflict Resolution

Conflict resolution addresses incoming evidence that changes the validity or scope of a stored record. Three distinct conflict types apply: (1) temporal supersession, where a newer state replaces an earlier one; (2) user correction, where stored facts or inferences are identified as wrong; and (3) contextual coexistence, where apparently contradictory states remain valid in different contexts. These types should not be collapsed into a single newest-wins rule.

AriadneMem [41] handles temporal conflicts by merging static duplicates and preserving state transitions as temporal edges. More generally, conflict handling can overwrite an earlier value, mark it inactive while retaining its evidence as in GRAVITY [42], or preserve successive states as queryable versions like ZEP [16].

D. Consolidation

Consolidation reorganizes accumulated memory after multiple writes. GAM [22] promotes completed event buffers into a topic-level structure while archiving detailed event graphs, serving abstraction and archival functions. All-Mem [43] applies gated SPLIT, MERGE, and UPDATE topology edits while preserving immutable evidence, serving deduplication, restructuring, and archival functions. Consolidation therefore serves abstraction, deduplication, restructuring, and archival, although individual systems implement different subsets. When observations are promoted into higher-level abstractions, errors in source evidence may persist into derived representations, and provenance links can support later correction without guaranteeing it.

E. Memory Removal

Memory removal takes two distinct forms: forgetting and deleting. Forgetting reduces a memory’s exposure, resolution, weight, or retrieval likelihood while some representation remains stored. ScrapMem [44] applies Optical Forgetting, which progressively reduces the resolution of older scrapbook pages, suppressing low-value detail while lowering storage cost.

Deletion changes what remains stored, active, or recoverable. Kumiho [45] uses versioned belief revision: immutable revisions are retained, mutable tags indicate the active state, and deprecated memories are excluded from normal retrieval unless explicitly requested. This design retains full progression history for auditing.

Removing source evidence can require revising or removing derived facts, summaries, personas, or higher-level abstractions produced from it to prevent the active user model from

becoming internally inconsistent: a profile claim may remain active after the interaction supporting it has been removed. The reliability and generality of dependency propagation across personalized memory systems remain open evaluation questions.

F. Discussion

Implemented systems connect evolution operations across different timescales. For example, GAM [22] buffers events before consolidation, and All-Mem [43] periodically edits its topology while retaining immutable evidence. AriadneMem [41] records temporal state transitions while Kumiho [45] separates current and historical revisions through mutable tags. These implementations link incoming evidence, structural reorganization, and active-state selection, but they do not establish one universally preferable lifecycle design.

V. MEMORY RETRIEVAL

Retrieval maps a current request to a bounded subset of the user’s persistent memory graph. We group retrieval methods into three mechanisms that may be combined: (1) similarity-based retrieval provides initial candidate generation or graph anchoring; (2) structure-based retrieval expands evidence through relations, abstraction levels, or relational views; and (3) adaptive and agentic retrieval controls query interpretation, expansion, or compression according to query intent or intermediate evidence. Their shared objective is to recover the user-specific evidence needed for the current response without exposing unrelated personal history.

A. Similarity-Based Retrieval

Similarity-based retrieval covers lexical, keyword, and dense-vector matching over memory stores. In graph memory, it commonly generates initial candidates or maps a query to graph anchors [27]. EMG-RAG [20] applies this selection problem to an editable graph of smartphone memories, retrieving user-specific records for downstream personalized tasks. Flat top- k selection can return correlated or near-duplicate interaction fragments [46], while semantic proximity alone may miss evidence connected through temporal, causal, entity, or abstraction relations.

B. Structure-Based Retrieval

Structure-based retrieval uses graph organization to expand or refine initial candidates. For personalized memory, this expansion connects a current request to user-specific evidence distributed across relations, abstraction levels, or temporal views. GAM [22] first retrieves topic nodes and then accesses archived event graphs when finer detail is required. MAGMA [47] represents each memory item across semantic, temporal, causal, and entity graphs and selects the view that best matches query intent.

Associative links provide another retrieval signal. AlsoMem [48] anchors dialogue utterances to automatically extracted clues in an associative memory graph and combines relevance, importance, and temporal alignment for candidate

ranking. xMemory [46] first selects complementary groups and memory components, then expands to source segments or messages when additional evidence is needed. Structural retrieval can recover user evidence distributed across sessions or abstraction levels, but uncontrolled expansion adds irrelevant nodes, unsupported paths, and context cost.

C. Adaptive and Agentic Retrieval

Adaptive and agentic retrieval changes query interpretation, graph exploration, stopping, routing, or compression according to the request and intermediate evidence. In personalized memory, this control helps distinguish the current user state from historical evidence and limits retrieval to the detail needed by the present request. MRAgent [33] uses a Cue-Tag-Content graph to iteratively explore and prune retrieval paths according to intermediate evidence, adapting memory access during inference.

Adaptive control also governs routing and compression. PRISM [49] combines intent-sensitive path selection with evidence compression to retain high-utility spans under a context budget. HingeMem [36] controls both which element-indexed routes to activate and how deeply to retrieve according to query demands. APEX-MEM [50] resolves conflicting or evolving information at retrieval time through a multi-tool retrieval agent.

D. Discussion

These three mechanisms can be combined in a retrieval pipeline: similarity anchoring identifies candidate entry points; structural expansion recovers distributed user evidence across relations, levels, or views; and adaptive control selects or compresses results under a context window.

Graph structure does not automatically improve retrieval since controlled comparisons show that results can depend on foundational system settings beyond graph structure [51]. Graph retrieval is therefore useful only when the representation exposes meaningful user relations, evolution maintains their validity, and retrieval returns relevant evidence.

VI. EVALUATION

Evaluation in personalized memory is still mostly downstream: benchmarks typically test whether a memory system answers correctly, retrieves supporting evidence, verifies updated user state, or reduces context cost over long histories. Since each benchmark targets different memory capabilities, Table II summarizes representative personalized-memory benchmarks and their evaluation metrics, adapting the metric categories discussed in [5].

The dominant benchmark family still tests long-horizon conversational recall. LoCoMo [52] is a common evaluation benchmark for graph-memory papers, including production-ready memory with a graph variant [31], topology-aware graph reasoning [41], learned traversal [53], multi-scale user memory [54], and graph-native versioned memory [45]. LongMemEval [55] evaluates longer histories, knowledge updates, single-session preference questions, temporal reasoning, and

abstention. These benchmarks are valuable because they force retrieval under multi-session noise, but they mostly observe whether the final answer is correct rather than whether the memory graph itself is correct.

Personalization-specific benchmarks shift the target from factual recall to user-state modeling. PersonaMem [56] tests dynamic user-profile and preference tracking, while PersonaMem-v2 [9] scales this setting to implicit personas, longer contexts, larger preference coverage, and agentic memory and therefore puts personalized memory systems to a new challenge. RealMem [57] extends personalization into project-oriented scenarios where goals, schedules, and dependencies evolve across sessions. PerLTQA [58] is useful because it organizes personal long-term memory into semantic memory, including profiles and social relationships, and episodic memory, including events and dialogues.

A smaller set of benchmarks tests requirements that are especially important for graph-based memory. EngramBench [59] isolates whether structured graph retrieval helps across persona spaces, temporal traces, and cross-space associations. EvoMemBench [60] evaluates self-evolving memory across in-episode/cross-episode and knowledge/execution settings, while EvoArena [61] stresses persistent environment change and preference-evolution chains. ActMemEval [62] tests whether retrieved memory supports implicit constraint and counterfactual reasoning, while ATM-Bench [63] adds multimodal, multi-source personal evidence. StructMemEval [64] asks whether memory is organized correctly.

Across these benchmarks, reported metrics fall into five groups:

- **Answer quality:** F1, exact match, multiple-choice accuracy, task success, and LLM-as-judge score.
- **Evidence retrieval:** Recall@k, NDCG@k, answer-turn retrieval accuracy, evidence-location accuracy, and multimodal evidence recall.
- **User-state modeling:** preference-category accuracy, profile or persona consistency, and correctness or helpfulness of personalized responses.
- **Memory evolution and structure:** update success, step or chain accuracy under changing environments, forgetting/update outcomes, structure-requiring task accuracy, and failure-attribution accuracy.
- **Efficiency:** retrieved-context size, token usage, latency, construction cost, query-time cost, and storage reduction.

VII. CHALLENGES AND FUTURE DIRECTIONS

Personalized graph memory remains a young but promising research frontier. Its central challenge is not merely storing user information in graph form but building a long-term, trustworthy, and actionable model of an individual user, which in turn motivates several future directions.

Scalable Lifelong Personal Memory. Personalized agents should support years of interaction rather than isolated sessions or short benchmark histories. Future work should build lifelong graph memories that grow with the user, consolidate

TABLE II

REPRESENTATIVE PERSONALIZED-MEMORY BENCHMARKS AND EVALUATION SIGNALS. ✓: EXPLICIT; △: PARTIAL/INDIRECT. ANS.: ANSWER QUALITY; RET.: EVIDENCE RETRIEVAL; USER: USER-STATE MODELING; EVOL.: EVOLUTION OVER TIME; MEMORG.: MEMORY ORGANIZATION; COST: EFFICIENCY; MM: MULTIMODAL EVIDENCE.

Benchmark	Focus	Setting	Ans.	Ret.	User	Evol.	MemOrg.	Cost	MM
<i>Long-horizon recall and retrieval</i>									
LoCoMo [52]	Long conversational memory	Real text-image conversations	✓	✓	△	–	–	–	✓
LongMemEval [55]	Long interactive memory	Long chat histories	✓	✓	△	△	–	△	–
<i>Personalization and user-state modeling</i>									
PersonaMem [56]	Dynamic user profiling	Simulated chats	✓	–	✓	✓	–	–	–
PersonaMem-v2 [9]	Implicit persona inference	Long persona chats	✓	–	✓	✓	–	✓	✓
RealMem [57]	Project-oriented memory	Multi-session projects	✓	✓	✓	✓	–	✓	–
PerLTQA [58]	Personal long-term QA	Semantic/episodic memory	✓	✓	✓	–	–	–	–
<i>Graph structure and evolution</i>									
EngramaBench [59]	Graph-vs-context memory	Structured retrieval	✓	✓	△	–	✓	✓	–
EvoMemBench [60]	Self-evolving memory	Episodic tasks	✓	–	–	✓	–	✓	–
EvoArena [61]	Environment evolution	Persistent environments	✓	–	✓	✓	–	✓	△
ActMemEval [62]	Action memory	Constraint reasoning	✓	✓	△	–	△	✓	–
ATM-Bench [63]	Multimodal personal evidence	Referential QA	✓	✓	✓	✓	–	△	✓
StructMemEval [64]	Memory structure	Structure-requiring tasks	✓	△	–	–	✓	–	–

recurring evidence, preserve historical context, adapt to preference drift, and at the same time, maintain accurate and efficient retrieval as memory grows. Achieving this may require new memory paradigms or architectures [65].

Multimodal Personal Memory Graphs. Current systems mostly construct memory from text, yet real personalization is grounded in multimodal evidence, including speech, images, documents, locations, actions, and interaction traces. Future work should develop graph memories that connect these heterogeneous signals while preserving their source modality, reliability, and temporal context.

Causal and Counterfactual User Modeling. Most existing memories are associative, recording what the user said or did. Stronger personalization requires understanding why a preference holds, when it changes, and what would happen under different contexts. Future graph memories should support causal and counterfactual reasoning over user state, distinguishing stable traits from temporary constraints or accidental observations.

REFERENCES

- [1] J. Luo *et al.*, “Large language model agent: A survey on methodology, applications and challenges,” *arXiv preprint arXiv:2503.21460*, 2025.
- [2] Y. Xu *et al.*, “Personalized generation in large model era: A survey,” in *Proc. of ACL*, 2025, pp. 24 607–24 649.
- [3] OpenClaw: Personal ai assistant. Software repository. OpenClaw. [Online]. Available: <https://github.com/openclaw/openclaw>
- [4] Hermes Agent: The agent that grows with you. Software repository. Nous Research. Accessed: Jun. 19, 2026.
- [5] Y. Xu, Q. Chen, Z. Ma, D. Liu, W. Wang, X. Wang, L. Xiong, and W. Wang, “Toward personalized LLM-Powered agents: Foundations, evaluation, and future directions,” *arXiv preprint arXiv:2602.22680*, 2026.
- [6] Z. Qiu, L. Luo, S. Pan, and A. W.-C. Liew, “Reasoning over user preferences: Knowledge graph-augmented LLMs for explainable conversational recommendations,” in *Proc. of ICDM*, 2025, pp. 1495–1504.
- [7] Z. Qiu, Y. Tao, S. Pan, and A. W.-C. Liew, “Knowledge graphs and pretrained language models enhanced representation learning for conversational recommender systems,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 4, pp. 6107–6121, 2025.
- [8] C. Yang, C. Zhou, Y. Xiao, S. Dong, L. Zhuang, Y. Zhang, Z. Wang, Z. Hong, Z. Yuan, Z. Xiang, S. Chen, H. Zhou, Q. Zhang, N. Liu, J. Su, X. Wang, Y. Chang, and X. Huang, “Graph-based agent memory: Taxonomy, techniques, and applications,” *arXiv preprint arXiv:2602.05665*, 2026.
- [9] B. Jiang, Y. Yuan, M. Shen, Z. Hao, Z. Xu, Z. Chen, Z. Liu, A. R. Vijjini, J. He, H. Yu, R. Poovendran, G. W. Wornell, L. H. Ungar, D. Roth, S. Chen, and C. J. Taylor, “Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory,” *arXiv preprint arXiv:2512.06688*, 2025.
- [10] Z. Huang, Z. Tian, Q. Guo, F. Zhang, Y. Zhou, D. Jiang, and X. Zhou, “Licomemory: Lightweight and cognitive agentic memory for efficient long-term reasoning,” in *Findings of ACL*, 2026, pp. 36 842–36 858.
- [11] Z. Xu, S. Zhang, E. Poyraz, Y. Li, Y. Jin, X. Lu, X. Gu, K. Ramgopal, P. K. Bodigutla, and X. Wang, “Hierarchical long-term semantic memory for linkedin’s hiring agent,” in *Proc. of KDD*, 2026, pp. 8334–8345.
- [12] R. Yuan, S. Sun, Y. Li, Z. Wang, Z. Cao, and W. Li, “Personalized large language model assistant with evolving conditional memory,” in *Proc. of COLING*, 2025, pp. 3764–3777.
- [13] C. Packer, V. Fang, S. G. Patil, K. Lin, S. Wooders, and J. E. Gonzalez, “MemGPT: Towards LLMs as operating systems,” *arXiv preprint arXiv:2310.08560*, 2023.
- [14] Z. Pan, Q. Wu, H. Jiang, X. Luo, H. Cheng, D. Li, Y. Yang, C.-Y. Lin, H. V. Zhao, L. Qiu, and J. Gao, “Secom: On memory construction and retrieval for personalized conversational agents,” in *Proc. of ICLR*, 2025.
- [15] W. Ma, J. Nan, W. Wu, and Y. Chen, “What deserves memory: Adaptive memory distillation for LLM agents,” in *Proc. of ACL*, 2026, pp. 34 789–34 812.
- [16] P. Rasmussen, P. Paliychuk, T. Beauvais, J. Ryan, and D. Chalef, “Zep: A temporal knowledge graph architecture for agent memory,” *arXiv preprint arXiv:2501.13956*, 2025.
- [17] S. Liang, Y. Zhang, and Y. Guo, “Personaagent with GraphRAG: Community-aware knowledge graphs for personalized LLM,” in *Proc. of ICASSP*, 2026, pp. 18 247–18 251.
- [18] W. Mao, H. Tan, S. Liu, H. Liu, Y. Xu, H. Ji, and X. Wang, “Bi-mem: Bidirectional construction of hierarchical memory for personalized LLMs via inductive-reflective agents,” *arXiv preprint arXiv:2601.06490*, 2026.
- [19] H. Xu, S. Moon, H. Liu, B. Liu, P. Shah, and P. S. Yu, “User memory reasoning for conversational recommendation,” in *Proc. of COLING*, 2020, pp. 5288–5308.
- [20] Z. Wang, Z. Li, Z. Jiang, D. Tu, and W. Shi, “Crafting personalized agents through retrieval-augmented generation on editable memory graphs,” in *Proc. of EMNLP*, 2024, pp. 4891–4906.
- [21] H. P. Van, N. M. Hieu, K. P. T. Tuan, N. L. Hai, L. V. Ngo, N. T. N. Diep, and T. Le, “MemORAI: Memory organization and retrieval via

- adaptive graph intelligence for LLM conversational agents,” in *Findings of ACL*, 2026, pp. 28 235–28 253.
- [22] Z. Wu, H. Zhang, F. Lin, W. Xu, X. Xu, Y. Chen, H. P. Zou, S. Chen, W. Zhang, X. Liu, P. S. Yu, and H. Wang, “GAM: Hierarchical graph-based agentic memory for LLM agents,” in *Proc. of ACL*, 2026, pp. 34 647–34 664.
- [23] Z. Qiu, L. Luo, Z. Zhao, S. Pan, and A. W.-C. Liew, “Graph retrieval-augmented LLM for conversational recommendation systems,” in *Proc. of PAKDD*, 2025, pp. 344–355.
- [24] L. Luo, Z. Zhao, J. Liu, Z. Qiu, J. Dong, S. Panev, C. Gong, T.-T. Vu, G. Haffari, D. Phung, A. W.-C. Liew, and S. Pan, “G-reasoner: Foundation models for unified reasoning over graph-structured knowledge,” in *Proc. of ICLR*, 2026.
- [25] Y. Liu, G. Zhang, K. Wang, S. Li, S. Pan, and B. An, “Graph-augmented large language model agents: Current progress and future prospects,” *IEEE Intell. Syst.*, vol. 41, no. 2, pp. 45–55, 2026.
- [26] Z. Zhang *et al.*, “A survey on the memory mechanism of large language model based agents,” *arXiv preprint arXiv:2404.13501*, 2024. [Online]. Available: <https://arxiv.org/abs/2404.13501>
- [27] Y. Wu, T. Lin, Y. Zhou, F. Zhang, Q. Guo, X. Zhou, S. Wang, X. Liu, Y. Ma, and Y. Fang, “Memory in the llm era: Modular architectures and strategies in a unified framework,” 2026.
- [28] J. Luo *et al.*, “From storage to experience: A survey on the evolution of LLM agent memory mechanisms,” *arXiv preprint arXiv:2605.06716*, 2026.
- [29] S. K. Paul, S. Sharma, and N. Sareen, “GAAMA: Graph augmented associative memory for agents,” *arXiv preprint arXiv:2603.27910*, 2026.
- [30] Y. Wu, Y. Zhang, S. Liang, and Y. Liu, “SGMem: Sentence graph memory for long-term conversational agents,” *arXiv preprint arXiv:2509.21212*, 2025.
- [31] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, “Mem0: Building production-ready AI agents with scalable long-term memory,” in *Proc. of ECAI*, 2025, pp. 2993–3000.
- [32] Y. Song, C. Lyu, P. Zhang, S. Brunswicker, N. D. Dutt, and A. M. Rahmani, “DEMENTIA-PLAN: An agent-based framework for multi-knowledge graph retrieval-augmented generation in dementia care,” *arXiv preprint arXiv:2503.20950*, 2025.
- [33] S. Ji, Y. Li, and B. Hooi, “Memory is reconstructed, not retrieved: Graph memory for LLM agents,” *arXiv preprint arXiv:2606.06036*, 2026.
- [34] J. Cui, K. Pu, and T. Grossman, “LOOM: Personalized learning informed by daily LLM conversations toward long-term mastery via a dynamic learner memory graph,” *arXiv preprint arXiv:2511.21037*, 2025.
- [35] J. Sun, Z. Du, and Y. Chen, “Knowledge graph tuning: Real-time large language model personalization based on human feedback,” *arXiv preprint arXiv:2405.19686*, 2024.
- [36] Y. Zhong, Y. Gao, and H. Wang, “Hingemem: Boundary guided long-term memory with query adaptive retrieval for scalable dialogues,” in *Proc. of WWW*, 2026, pp. 3576–3587.
- [37] J. Yue, C. Hu, J. Sheng, Z. Zhou, W. Zhang, T. Liu, L. Guo, and Y. Deng, “Hypermem: Hypergraph memory for long-term conversations,” in *Proc. of ACL*, 2026, pp. 35 237–35 254.
- [38] J. Yu, Y. Fang, X. Liu, and Y. Ma, “H-mem: A novel memory mechanism for evolving and retrieving agent memory via a hybrid structure,” *arXiv preprint arXiv:2605.15701*, 2026.
- [39] J. Ye, X. Li, X. Yang, C. Huang, L. Nie, L. Yao, and D. Zhan, “Memweaver: Weaving hybrid memories for traceable long-horizon agentic reasoning,” in *Findings of ACL*, 2026, pp. 12 928–12 956.
- [40] H. Ha, J. Kim, C. Qian, J. Liu, W. M. Campbell, Y. Wu, Y. Zhang, K. McKeown, D. Hakkani-Tur, and H. Ji, “Memguard: Preventing memory contamination in long-term memory-augmented large language models,” *arXiv preprint arXiv:2605.28009*, 2026.
- [41] W. Zhu, X. Chen, Z. Wang, J. Wang, X. Dong, M. Huang, R. Cai, H. Sang, H. Wang, P. Qiu, Y. Deng, P. Tiwari, B. H. Rappazzo, and Y. Wang, “Ariadnemem: Threading the maze of lifelong memory for LLM agents,” *arXiv preprint arXiv:2603.03290*, 2026.
- [42] Y. Sun, B. Cao, D. Fang, L. Su, and W. Lam, “GRAVITY: Architecture-agnostic structured anchoring for long-horizon conversational memory,” *arXiv preprint arXiv:2605.01688*, 2026.
- [43] C. Lv, H. Chang, Y. Guo, S. Tao, and S. Zhou, “All-mem: Agentic lifelong memory via dynamic topology evolution,” *arXiv preprint arXiv:2603.19595*, 2026.
- [44] J. Chang and Y. Ren, “Scrapmem: A bio-inspired framework for on-device personalized agent memory via optical forgetting,” *arXiv preprint arXiv:2605.03804*, 2026.
- [45] Y. B. Park, “Graph-native cognitive memory for AI agents: Formal belief revision semantics for versioned memory architectures,” *arXiv preprint arXiv:2603.17244*, 2026.
- [46] Z. Hu, Q. Zhu, H. Yan, Y. He, and L. Gui, “Beyond RAG for agent memory: Retrieval by decoupling and aggregation,” *arXiv preprint arXiv:2602.02007*, 2026.
- [47] D. Jiang, Y. Li, G. Li, and B. Li, “MAGMA: A multi-graph based agentic memory architecture for AI agents,” in *Proc. of ACL*, 2026, pp. 36 848–36 865.
- [48] K. Zhang, X. Zhang, E. Ahmed, H. Jiang, C. Kumar, K. Sun, Z. Lin, S. Sharma, S. Oraby, A. Colak, A. Aly, A. Kumar, X. Liu, and X. L. Dong, “Assomem: Scalable memory QA with multi-signal associative retrieval,” *arXiv preprint arXiv:2510.10397*, 2025.
- [49] J. Peng, Z. Wan, W. Liu, and Q. Sun, “PRISM: Pareto-efficient retrieval over intent-aware structured memory for long-horizon agents,” *arXiv preprint arXiv:2605.12260*, 2026.
- [50] P. Banerjee, M. Moshtaghi, S. Subramanian, A. Misra, and A. Chadha, “APEX-MEM: Agentic semi-structured memory with temporal reasoning for long-term conversational AI,” in *Proc. of ACL*, 2026, pp. 16 470–16 489.
- [51] S. Hu, Y. Wei, J. Ran, Z. Yao, X. Han, H. Wang, R. Chen, and L. Zou, “Does memory need graphs? a unified framework and empirical analysis for long-term dialog memory,” in *Proc. of ACL*, 2026, pp. 26 758–26 782.
- [52] A. Maharana, D.-H. Lee, S. Tulyakov, M. Bansal, F. Barbieri, and Y. Fang, “Evaluating very long-term conversational memory of LLM agents,” in *Proc. of ACL*, 2024, pp. 13 851–13 870.
- [53] D. Jiang, Y. Li, G. Li, Q. Li, and B. Li, “HAGE: Harnessing agentic memory via RL-Driven weighted graph evolution,” *arXiv preprint arXiv:2605.09942*, 2026.
- [54] A. Tian, Y. Lu, X. Fan, C. Wang, L. Zhou, Y. Zhang *et al.*, “RGMem: Renormalization group-inspired memory evolution for language agents,” *arXiv preprint arXiv:2510.16392*, 2025.
- [55] D. Wu, H. Wang, W. Yu, Y. Zhang, K.-W. Chang, and D. Yu, “Longmemeval: Benchmarking chat assistants on long-term interactive memory,” in *Proc. of ICLR*, 2025.
- [56] B. Jiang, Z. Hao, Y.-M. Cho, B. Li, Y. Yuan, S. Chen, L. H. Ungar, C. J. Taylor, and D. Roth, “Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale,” *arXiv preprint arXiv:2504.14225*, 2025.
- [57] H. Bian, Z. Yao, S. Hu, Z. Xu, S. Zhang, Y. Guo, Z. Yang, X. Han, H. Wang, and R. Chen, “Realmem: Benchmarking LLMs in real-world memory-driven interaction,” in *Findings of ACL*, 2026, pp. 14 349–14 365.
- [58] Y. Du, H. Wang, Z. Zhao, B. Liang, B. Wang, W. Zhong, Z. Wang, and K.-F. Wong, “PerLTQA: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering,” in *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, 2024, pp. 152–164.
- [59] J. Acuna, “Engramabench: Evaluating long-term conversational memory with structured graph retrieval,” *arXiv preprint arXiv:2604.21229*, 2026.
- [60] Y. Wang, Z. Zhang, M. Chi, K. Yu, Y. Li, M. Peng, B. Tong, C. Zhang, Y. Zhou, and J. Li, “Evomembench: Benchmarking agent memory from a self-evolving perspective,” *arXiv preprint arXiv:2605.18421*, 2026.
- [61] J. Xu, Q. Li, J. Wu, Y. Lan, S. S. Li *et al.*, “Evoarena: Tracking memory evolution for robust LLM agents in dynamic environments,” *arXiv preprint arXiv:2606.13681*, 2026.
- [62] X. Zhang, Z. Sun, C. Yang, Y. Jin, Y. Zhang, and W. Hu, “Actmem: Bridging the gap between memory retrieval and reasoning in LLM agents,” *arXiv preprint arXiv:2603.00026*, 2026.
- [63] J. Mei, J. Chen, G. Yang, X. Hou, M. Li, and B. Byrne, “According to me: Long-term personalized referential memory QA,” *arXiv preprint arXiv:2603.01990*, 2026.
- [64] A. Shutova, A. Olenina, I. Vinogradov, and A. Sinitsin, “Evaluating memory structure in LLM agents,” *arXiv preprint arXiv:2602.11243*, 2026.
- [65] R. W. H. Quek, S. Lee, A. W. L. Leong, A. Verma, A. Prakash, N. F. Chen, B. K. H. Low, D. Rus, and A. Solar-Lezama, “MeMo: Memory as a Model,” *arXiv preprint arXiv:2605.15156*, 2026.