LIFEBENCH: A BENCHMARK FOR LONG-HORIZON MULTI-SOURCE MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling user memory is essential for building personalized AI agents. However, existing benchmarks mainly focus on sparse user information explicitly stated in conversational narratives. In contrast, real-world information is scattered across diverse digital sources, such as messages, calendars, and photos. Such observations form a dense and evolving data stream that only indirectly reveals underlying user preferences and habits. This mismatch limits the effective evaluation of agent memory systems. However, collecting real user data remains prohibited due to privacy concerns. This paper introduces LIFEBENCH, a simulation framework that combines *hierarchical outline planning* for consistency control with *Objective Agents* for grounding generated data in real-world information. Evaluation shows that LIFEBENCH better captures real-world preference shifts than existing memory benchmarks. In addition, its Objective-Agent mechanism enables digital traces that resemble human mobility patterns in real-world data without supervised training. We further synthesize a moderate-sized dataset and evaluate state-of-the-art memory systems. Results show that LIFEBENCH more clearly exposes performance gaps among current systems. Deeper analysis reveals insights across data distributions and settings, including, e.g., *sparse vs. dense data distributions*.

1 INTRODUCTION

To understand user preferences and provide tailored assistance, personalized agents need to build user memory from fragmented signals (e.g., phone apps and messages). Existing long-term memory benchmarks cover only part of this setting. As shown in Table 1, LoCoMo (Maharana et al., 2024), LongMemEval (Wu et al., 2025), PerLTQA (Du et al., 2024), and PersonaMem-v2 (Jiang et al., 2025) are built primarily around conversational histories. Mem-PAL (Huang et al., 2025a) additionally incorporates application logs, but focuses on summary and response generation, with limited evaluation of reasoning capabilities (see Table 10). Most existing datasets reflect sparse user activities over long periods, rather than dense daily activities indirectly observed through digital traces such as phone data (Figure 1 compares conversation-centric view with multi-source data view).

Constructing such a dataset that faithfully reflects real-world user life is challenging. Authentic longitudinal records are privacy-sensitive and rarely capture mobility, schedules, health, and device use together. Existing synthesis methods typically target a single modality or short time window (Xiong et al., 2024; Huang et al., 2025b; Ju et al., 2025; Liu et al., 2025), failing to capture the diverse

Table 1: Comparison with existing memory benchmarks.

Benchmark	Domain	Artifacts	Life events	#Items	Context depth	Realism-aware synthesis	Dynamic indexing	Target scenario
PerLTQA	Personal	Chats	Sparse	8,593	∼1M	✗	✗	Personalized; companion agents
LoCoMo	Personal	Chats	Sparse	7,512	∼16.6K	✗	✗	Long-term social chatbots
LongMemEval	Personal	Chats	Sparse	500	115K/1.5M	✗	✗	Memory-enabled chat assistants
Mem-PAL	Personal	Chats, logs	Sparse	826	∼55K	✗	✗	Personalized service assistants
PersonaMem-v2	Personal	Chats	✗	20K+	32K–128K	✗	✗	Personalized chatbot responses
LIFEBENCH (ours)	Personal	Chats, apps	Dense	3,380	1.31M	✓	✓	Phone assistants; User modeling

Notes. Life events mark explicit synthesized personal-life events; context depth is the average number of tokens per user history; and dynamic indexing means that memory addition and querying are chronologically interleaved. Realism-aware synthesis means that the generation process actively targets real-world priors and grounding. Target scenario summarizes the tasks in the primary evaluation setting.

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

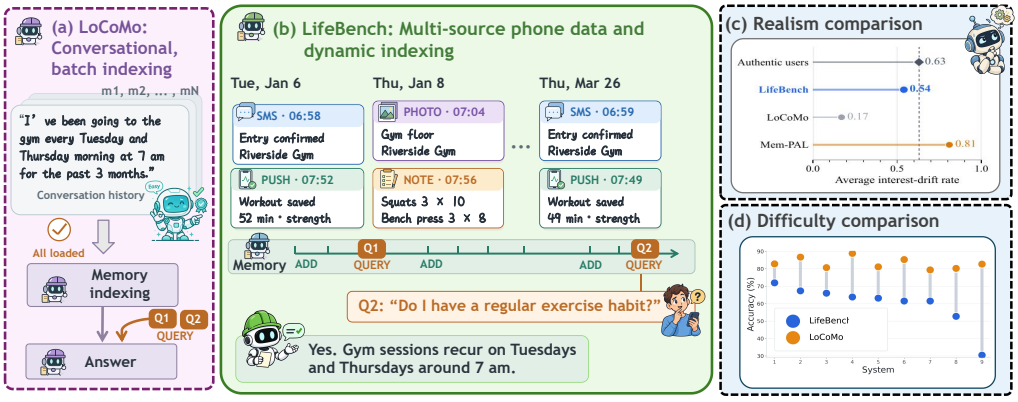

Figure 1: Comparison of LoCoMo and LifeBench. (a) LoCoMo uses a conversation-centric view with batch-indexing evaluation. (b) LifeBench requires memory systems to process fragmented mobile data injected in real time. (c) User interest-drift rates across benchmarks. (d) Memory systems exhibit greater performance variance on LifeBench.

experiences represented in multi-source personal memory. For agent memory evaluation, existing benchmarks also lack explicit realism consideration, and their synthesis frameworks are generally not publicly available to facilitate further research.

This paper introduces LIFEBENCH, a framework inspired by the partonomic event structure (Zacks & Tversky, 2001; Zacks et al., 2001) for data synthesis. Specifically, it adopts hierarchical event generation, first synthesizing coherent life events and then decomposing them to daily events and digital observations. This approach helps maintain consistency among significant events over long time horizons. We also integrate real-world priors to ensure behavioral diversity, followed by a dual-agent simulator in which an objective agent grounds the synthetic data in real-world information.

We evaluate the synthesized data for realism from multiple aspects, including preference shifts (see Figure 1 (c)) and mobility patterns (see Section 2), by comparing them with real-world data. Building on this data, we construct a comprehensive QA benchmark for evaluating memory agents. Unlike previous benchmarks, we focus on dynamic memory digestion and evaluation: data are digested daily as they arrive, and QA is evaluated at different timestamps, rather than indexing a complete year-long history in batch before evaluation. We evaluate nine contemporary memory systems and find that LifeBench better distinguishes their respective strengths and weaknesses (see Figure 1 (d)).

Our contributions are threefold: (1) a controllable, realism-aware framework for synthesizing coherent year-long lives and heterogeneous digital traces; (2) a comprehensive benchmark for evaluating dynamic indexes in real-world mobile agent memory scenarios; and (3) a unified evaluation and diagnostic analysis of nine representative memory systems. Our contribution extends beyond the released benchmarks: our open-source synthesis framework enables other research teams to generate data and evaluate memory systems under diverse settings, such as different user demographics or rare events, by modifying real-world priors and objective-agent APIs.

2 LIFEBENCH: MEMORY SIMULATION FRAMEWORK

As shown in Figure 2, we synthesize personal lives with DeepSeek-V4-Pro through four stages: persona and context construction, hierarchical long-term planning, day-level simulation, and rendering into heterogeneous phone records.

2.1 SYNTHESIS PIPELINE ARCHITECTURE

2.1.1 PERSONA SYNTHESIS

Base personas are constructed by sampling attributes from multiple sources and refining them with LLMs. We sample background attributes (e.g., ethnicity, marital status, occupation, and health sta-

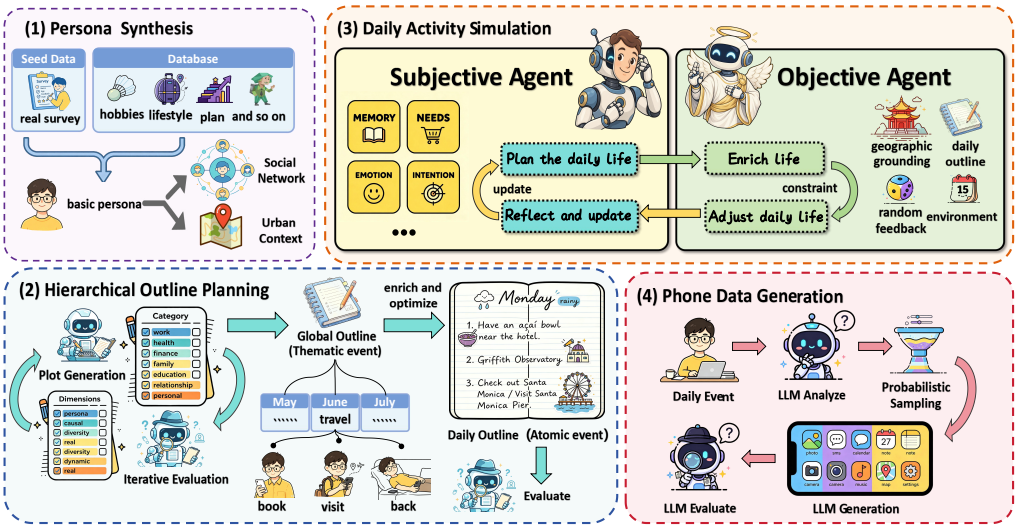

Figure 2: The LIFEBENCH data synthesis pipeline architecture and key algorithmic components.

tus) from anonymized survey data[1] to form seed personas, and sample habits, plans, personality, and skills from PersonaHub (Ge et al., 2024) for enrichment. The LLM then reconciles these attributes for consistency and expands the persona with lifestyle details and personal growth trajectories.

Social network. Leveraging base personas, LLMs infer and generate social circles for each individual, and define the background, shared information, specific members and their relational networks within each circle. The resulting overall social network comprises approximately 20–30 associated individuals, each equipped with a detailed role profile.

Personalized urban landscape. We invoke the map-service API[2] to anchor personas to real urban environments. Residential and work locations are retrieved according to the administrative regions in the base persona and social network. The candidates are selected by ranking based on geographic fitness, completeness, commuting feasibility, and travel preferences. We then search points of interest near the residence (e.g., supermarkets) as frequently visited places for the persona.

2.1.2 HIERARCHICAL OUTLINE PLANNING

This work uses coarse-to-fine hierarchical planning, first outlining year-long storylines and major events, then decomposing them into monthly and daily events.

Global Outline Generation. To ensure narrative coherence, we first construct a year-long *global plot* that serves as the individual's life narrative. We predefine about 50 candidate plots across seven categories: work, health, finance, relationships, family, education, and personal life. The LLM selects up to 10 persona-tailored plots, which the generation agent integrates into a coherent outline. A critic agent evaluates the outline for persona alignment, temporal plausibility, cross-theme interactions, event density, and causal continuity, with feedback iteratively used to refine the outline. For controlled synthesis, the framework supports priority labels that allow users to weight specific plots or directly converse with the generation agent to inject custom plotlines.

Thematic Event Decomposition. Guided by the plot outline, the LLM autoregressively adjusts and instantiates events at the monthly level. A thematic event is assigned to the month containing its start date, though it may span multiple months. The system integrates persona evolution, historical context, and the global outline to generate concrete monthly events, with each month's analysis propagated to the next. These thematic events are recursively decomposed into hierarchical event trees until reaching atomic events lasting less than one day, forming a complete event partition tree.

[1] https://www.cnsda.org/index.php
[2] https://lbs.amap.com/

Atomic Event Optimization. Atomic events capture short-term behaviors shaped by emotions, habits, and living environments. The LLM refines them within two-week windows using persona and historical context, adding routine behaviors such as exercise and adapting events to factors such as weather and mood. It then summarizes each month and checks global consistency, routing detected issues back to the corresponding months for revision. This coarse-to-fine and fine-to-coarse process ensures both global narrative coherence and local temporal consistency.

2.1.3 DUAL-AGENT DAILY ACTIVITY SIMULATION

To generate these daily activities, we adopt a dual-agent simulation framework: a subjective agent plays a human role to produce plausible activities, while an objective agent verifies and grounds these activities in real-world data to ensure fidelity and consistency.

Subjective Agent. The Subjective Agent is an LLM that simulates human behavior to generate daily activities conditioned on atomic events and the persona's daily contextual attributes. Persona attributes include long-term memory, short-term memory, episodic memory, affective states, intentions and needs, and rolling behavioral statistics. Full details are provided in the appendix B.1.

Objective Agent. The Objective Agent enforces physical and temporal constraints. Leveraging a geographic allocation algorithm built upon rules and persona, together with improved exploration-and-preferential-return (EPR) (Song et al., 2010) and gravity-based mobility models (Zipf, 1946; Wilson, 1971), it determines real-world locations, transportation modes and commuting durations for each activity. It then coordinates and validates the itinerary to guarantee the feasibility of daily plans. Meanwhile, it introduces supplementary events and external environmental feedback (e.g., social invitations, traffic jams). Full details are provided in the appendix B.2.

2.1.4 PHONE DATA GENERATION.

Phone data provides noisy, incomplete observations of underlying user activities, events, and habits, posing a unique challenge for memory-system benchmarking. It includes contacts, mobile records (SMS, calls, push notifications, photos, notes, calendar, and agent chats), and noise data. Contacts are generated from the persona's social network. For each activity, an LLM estimates the likelihood of different mobile operations, samples operations accordingly, and generates the corresponding records. We additionally generate event-irrelevant noise, such as recommended ads, video suggestions, and trivial device alerts, which do not conflict with the user's life events.

2.2 QUESTION CATEGORIES AND GENERATION

As shown in Figure 3, our taxonomy comprises five high-level capability categories and nine fine-grained question types. Unlike existing benchmarks, which may infer persistent preferences or habits from salient events (e.g., starting to play badminton), our taxonomy leverages the dense, continuous histories in LIFEBENCH. In particular, **Non-declarative (ND)** questions require summarizing daily traces to infer interests or habits, and their development over time. Answers thus depend on accumulated behavioral evidence rather than single declarative events. Appendix B.3 provides design principles and examples for all types. Questions are LLM-generated following manually designed rules, then manually filtered; only those confirmed correct by humans are retained.

2.3 SYNTHESIS DATA STATISTICS

We generate two datasets: LIFEBENCH-BEIJING, with five Beijing personas with 295 person-days, including 265 complete days for mobility evaluation; and LIFEBENCH-GENERAL, with ten diverse personas[3] and year-long life streams, totaling 3,656 person-days, 57,486 events, 47,409 phone records, and 3,380 QA pairs. Each persona averages 1.31M phone-record tokens. Beijing is selected for LIFEBENCH-BEIJING because public real-world trajectories and synthetic baselines are available for comparison. Detailed statistics are provided in Appendix C.

[3]The same number of users as LoCoMo.

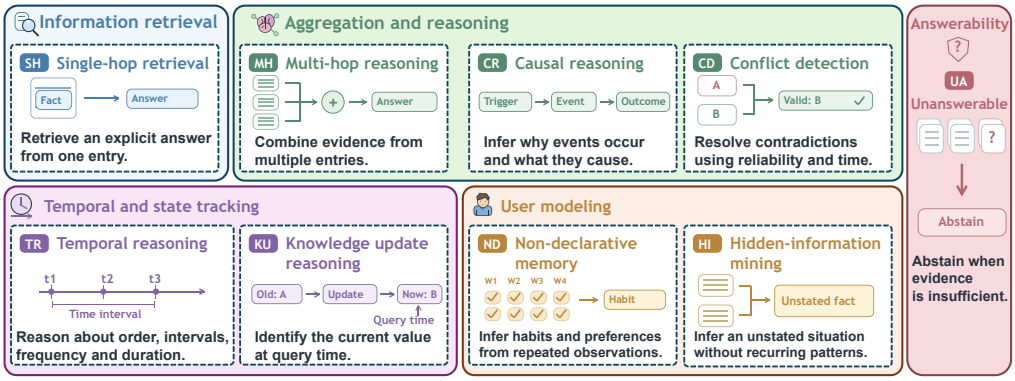

Figure 3: Nine question types: Single-hop Retrieval (SH), Multi-hop Reasoning (MH), Causal Reasoning (CR), Conflict Detection (CD), Temporal Reasoning (TR), Knowledge Update (KU), Non-declarative Memory (ND), Hidden-information Mining (HI), and Unanswerable (UA). This taxonomy covers the most comprehensive set of question types. (see Appendix B.3).

2.4 Data Evaluation

We evaluate LifeBench data from four complementary perspectives: population-level behavior, mobility, automated evaluation, and human evaluation.

Population-level behavior. We sample 90-day preference trajectories from LifeBench-General and Mem-PAL and compare them with real interest dynamics from OmniBehavior (Chen et al., 2026). The LoCoMo trajectory and statistics are taken directly from the OmniBehavior paper. Figure 4 shows that real users maintain interests that persist, emerge, decay, and recur. LoCoMo shows limited change, whereas Mem-PAL shows excessive turnover. We quantify day-to-day change using the average Jaccard distance between interest-category sets on consecutive valid days (Appendix D.1). As shown in Figure 1(c), LifeBench (0.54) is closest to the real-user reference (0.63), compared with LoCoMo (0.17) and Mem-PAL (0.81).

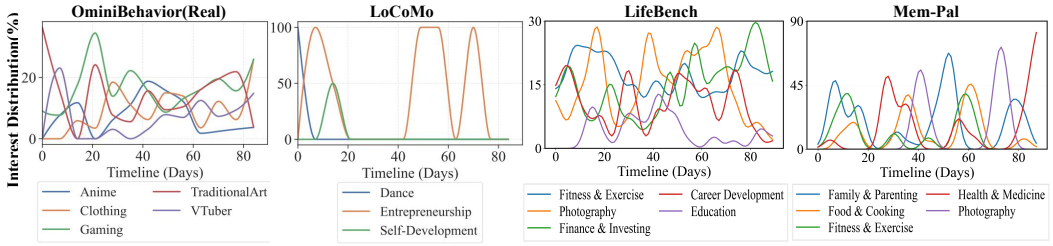

Figure 4: Long-term interest trends of real and synthesized users.

Mobility evaluation. We use a sample of 265 complete days from LifeBench-Beijing for comparison. LifeBench-Beijing, WorldMove (Yuan et al., 2026), and CoPB (Shao et al., 2024) are compared against three real-world references—ChinaMobile cellular locations, Tencent POI stays, and GeoLife GPS traces—on a common ∼1-km Beijing grid with 48 half-hour slots per day (Zheng et al., 2010; Yuan et al., 2026; Shao et al., 2024). We evaluate jump length, daily travel distance, radius of gyration, dwell time, and daily visited locations using Jensen–Shannon

Table 2: Mobility evaluation against GeoLife (JSD; lower is better).

Metric	LifeBench-Beijing	WorldMove	CoPB
Jump length Δr	**0.0618**	0.3696	0.2726
Daily travel distance r_w	**0.2194**	0.3166	0.2342
Radius of gyration r_g	**0.0828**	0.2663	0.1617
Dwell time Δt	0.1128	**0.0663**	0.1937
Daily visited locations S_d	0.2025	**0.1124**	0.2645

Note. Means over 50 equal-size 265-user-day samples.

divergence (JSD). Table 2 reports results on five GeoLife metrics, with all three references provided in Appendix D.2. LIFEBENCH-BEIJING achieves the lowest GeoLife JSD for jump length, daily distance, and radius of gyration, and the lowest ChinaMobile JSD for jump length and daily distance. WorldMove more closely matches dwell time and daily visited locations on these references. Notably, unlike WorldMove and CoPB, LIFEBENCH-BEIJING requires no training on real trajectories while achieving competitive performance across individual mobility metrics.

Automated evaluation is conducted on all users of LIFEBENCH-GENERAL using three metrics. To measure *relation consistency* (%), we build a knowledge base from each persona's initial profile, then check events chronologically for consistency with evolving interpersonal relationships. *Normalized Shannon entropy* measures coverage of the event-type distribution relative to its maximum possible entropy. Finally, *Simpson diversity index* measures its evenness through the probability that two sampled events belong to different categories. The results in Table 3 show a relation-consistency score of 95.2%, indicating that most generated events respect the evolving social relationships. Normalized Shannon entropy of 0.75 and Simpson diversity of 0.78 show substantial event-type variety.

Table 3: Automatic and human evaluation of LifeBench.

Automatic		Human	
Metric	Score	Metric	Score
Relation consistency	95.2%	Daily-activity validity	98.8%
Normalized Shannon entropy	0.75	Phone-record validity	95.38%
Simpson diversity index	0.78	Diversity score (/10)	7.67

Human evaluation. For human evaluation, we sample two consecutive months from three users and assign two reviewers to each user's activities and phone records. *Daily-activity validity* measures temporal, content, location, and participant coherence, while *phone-record validity* assesses consistency with the corresponding activity and content plausibility. A 10-point *diversity score* evaluates inter- and intra-user variability, event balance, fine-grained behavioral diversity, and realistic complexity. Table 3 summarizes the results. The synthesized data achieves high validity for both daily activities and phone records (98.8% and 95.38%), with minor issues mainly caused by repeated events or step jumps from hallucinations. No major logical inconsistencies are observed. The data also shows strong inter- and intra-user diversity, though fine-grained behavioral diversity and non-stereotypical representation remain areas for improvement.

3 EVALUATING MEMORY SYSTEMS ON LIFEBENCH

3.1 MEMORY SYSTEMS AND EXPERIMENTAL SETUP

We evaluate nine long-term memory systems on LIFEBENCH and LoCoMo. They span five memory designs. *Extraction-based methods*, including Mem0 (Chhikara et al., 2025) and Hindsight (Latimer et al., 2025), extract entities or facts related to when, what, who, and where. *Hierarchical abstraction methods*, such as EverMemOS (Hu et al., 2026) and MemU (MemU), organize data into memory units using summarization or clustering. *Operating-system-style methods*, such as MemOS (Li et al., 2025), manage memory units like cache and storage in OS. *Graph-centered methods*, including Cognee (cog, 2026), Zep (Rasmussen et al., 2025), and GraphRAG (Edge et al., 2024), organize memories into graphs. Finally, *hybrid methods*, such as MindMemOS (Liang et al., 2026), combine multiple memory designs. Appendix E.1 details memory systems.

Unless otherwise stated, we use DeepSeek-V4-Flash as the common answer model and LLM judge, Qwen3-Embedding-4B for embeddings, and Qwen3-Reranker for reranking. Each query is judged three times by majority vote. The LLM judge achieves 93.3% agreement with human annotations (Appendix E.2), validating its effectiveness. We enable each system's strongest available reasoning or retrieval mode and use top-$k = 40$ when a fixed cutoff is supported, otherwise using its native adaptive retrieval. Client-side and local components run on a machine with 16 GB RAM and an AMD Ryzen 5 4600H, while hosted services run on provider-managed infrastructure.

3.2 MAIN RESULTS

As shown in Table 4, under the default DeepSeek-V4-Flash backbone, Hindsight (71.98%), MindMemOS (67.37%), and EverMemOS (66.04%) achieve the three highest LifeBench Micro accuracies. MindMemOS leads on Conflict Detection with 84.67%, but its lower Macro accuracy (55.89%)

Table 4: Accuracy (%) on LifeBench, where Micro is the accuracy over all questions, whereas Macro is the arithmetic mean of the nine question types. Gold Evidence† directly provides the raw golden evidence to the answer model. LoCoMo is for reference, which excludes adversarial questions.

Base LLM	Memory system	LifeBench											LoCoMo Overall
		SH	MH	TR	ND	KU	CR	CD	HI	UA	Micro	Macro	
DeepSeek-V4-Flash	Mem0	78.09	41.41	32.45	48.48	75.84	52.55	78.07	32.47	55.73	61.54	55.01	85.32
	Cognee	71.69	37.07	27.36	45.44	72.49	43.16	68.63	30.93	**92.14**	63.17	54.32	81.12
	Hindsight	**83.43**	**53.98**	44.34	**58.75**	**85.09**	**67.83**	80.66	**49.48**	75.56	**71.98**	**66.57**	82.83
	MemU	60.34	23.87	17.92	23.19	47.30	30.29	62.26	23.20	90.77	52.75	42.13	80.26
	MemOS	72.94	32.91	26.98	33.65	71.47	40.21	68.63	35.57	88.89	61.51	52.36	79.40
	EverMemOS	71.76	48.55	38.30	54.75	81.23	59.25	67.45	34.02	85.30	66.04	60.07	80.69
	MindMemOS	79.83	42.04	42.45	35.36	69.67	38.07	**84.67**	26.80	84.10	67.37	55.89	86.70
	Zep	77.78	49.46	**46.23**	43.92	73.78	53.89	83.25	39.69	54.53	63.85	58.06	**88.84**
	GraphRAG	22.10	12.93	12.83	19.77	16.97	15.01	11.08	10.31	88.38	30.50	23.26	82.72
	Gold Evidence†	98.14	94.85	94.91	93.73	96.14	94.10	98.11	93.30	100.00	97.28	95.92	–
GLM-5.2	Hindsight	**85.60**	53.62	45.85	56.65	83.55	**66.76**	**80.90**	**40.72**	**84.79**	**74.41**	**66.49**	–
	EverMemOS	79.64	**58.95**	**59.25**	**58.75**	**84.83**	57.64	78.54	**40.72**	71.45	70.95	65.53	–
Qwen3.8-MAX	Hindsight	**83.30**	50.72	38.87	**54.94**	83.29	**64.88**	**82.08**	**44.85**	85.30	**72.31**	**65.36**	–
	EverMemOS	73.49	**55.06**	**55.09**	54.37	**86.38**	55.50	72.17	43.81	**86.84**	69.32	64.75	–

reveals substantial variation across capabilities. Zep achieves the best Temporal Reasoning accuracy (46.23%) and the highest LoCoMo score, yet reaches only 63.85% Micro accuracy on LifeBench. This contrast shows that strong performance on a conventional conversational benchmark does not necessarily transfer to the data with different characteristics of LIFEBENCH. *All memory systems continue to struggle with the most difficult question types, particularly Temporal ($\leq$ 46.23%), Hidden-information ($\leq$ 49.48%), and Non-declarative questions ($\leq$ 58.75%).*

Backbone choice has a clear but category-dependent effect. GLM-5.2 achieves the highest Micro accuracy for both Hindsight (74.41%) and EverMemOS (70.95%), outperforming Qwen3.8-Max and DeepSeek-V4-Flash overall. Its gains are not uniform, however: GLM-5.2 substantially improves EverMemOS on Single-hop, Multi-hop, Temporal, and Non-declarative questions, whereas Hindsight with DeepSeek-V4-Flash remains strongest on Causal (67.83%) and Hidden Information (49.48%). Table 25 in the Appendix shows Hindsight and EverMemOS across different LLMs. *These results suggest that different backbones offer distinct category-level advantages rather than uniformly improving every reasoning capability.*

3.3 RETRIEVAL CAPABILITY ANALYSIS

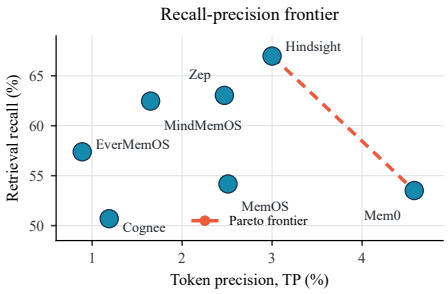

Figure 5: Retrieval recall vs. token precision (TP). TP = tokens in relevant sentences / final recall tokens.

Table 5: Mean tokens per memory unit and unit counts across retrieval stages. R1/R2 denote two agentic retrieval rounds; output units are mean final unit counts.

System	Tokens per unit	Retrieval-capacity trace	Avg. final return units
Mem0	76.85	320 → RRF → 40 → rerank	40.00
Cognee	291.56	500 → triplet-importance score	89.77
Hindsight	103.68	900 → RRF → 300 → rerank	37.67
MemOS	105.80	480 → combine → 80 → rerank	39.52
EverMemOS	815.50	R1/R2: 60 → RRF → 20 each; merge 40 → rerank	19.64
MindMemOS	1425.40	R1/R2: 60 → RRF → 10 each; merge 20 → rerank	15.28
Zep	287.38	560 → RRF → 160 → rerank	40.00

We evaluate retrieval separately to explain differences in QA performance. During consolidation, each system converts raw phone and activity records into its own *memory units*, such as atomic facts in Mem0, source-linked facts in Hindsight, or graph entities/relations in Cognee. Since gold evidence is available only at the raw-data level, we use DeepSeek-V4-Pro to semantically align memory units with the gold evidence.

Figure 5 shows the recall–precision trade-off, while Table 5 summarizes memory-unit length and counts across retrieval stages. For example, Hindsight retrieves 900 units from four retrieval methods, reduces them to 300 via lightweight RRF fusion (Cormack et al., 2009), and reranks them with Qwen3-Reranker, retaining about 37.67 units on average for generation. Token counts include only content-bearing tokens, excluding structural tokens such as JSON syntax. GraphRAG is excluded because its cloud-based API does not expose intermediate memory units.

The results show a *granularity–coverage trade-off*: Hindsight achieves the highest recall with moderate retrieval volume by combining semantic, lexical, graph, and temporal signals. Cognee returns the most units (89.77), mainly entities and relations from extensive graph traversal, but localizes evidence poorly. Although Hindsight's recall advantage aligns with its end-to-end accuracy in Table 4, this correspondence does not hold across other systems, highlighting the complex relationship between retrieval and end-to-end QA.

3.4 Beyond Retrieval Sufficiency: Locating Answer Failures

Table 6: Stage-wise accuracy on the shared 183-question subset. Stage loss (in %) is relative to the preceding condition; total loss is from *Gold-Raw* to *Normal Retrieval*.

	Hindsight		EverMemOS	
Condition	Acc.	Loss	Acc.	Loss
Gold-Raw	96.72	–	96.72	–
Gold-Unit	94.54	2.18	83.61	13.11
Normal Retrieval	67.76	26.78	58.47	25.14
Total loss	–	**28.96**	–	**38.25**

To quantify the impact of memory consolidation and retrieval on final QA, we compare three conditions: (1) *Gold-Raw*, which provides original gold evidence; (2) *Gold-Unit*, which provides system-generated units linked to the gold evidence; and (3) *Normal Retrieval*, which uses the non-controlled pipeline. *Gold-Unit* preserves each system's normal unit format. We analyze Hindsight and EverMemOS, two strong systems representing extraction-based and abstraction-based consolidation. From *Gold-Raw* to *Gold-Unit*, accuracy drops by 2.18 points for Hindsight and 13.11 points for EverMemOS, indicating greater information loss. EverMemOS passes only compressed Episodes, whereas Hindsight links *MemoryFacts* to source chunks, preserving more raw information. Both systems incur further substantial losses under *Normal Retrieval*. *Overall, retrieval is the main source of loss, while raw-data linking and source fidelity also matter.*

3.5 Robustness under Dynamic Memory Indexing

To assess sensitivity to streaming data, we compare three representative systems—GraphRAG, Hindsight, and EverMemOS—under dynamic (our default) and batch indexing, for which all data is ingested before QA evaluation. Batch indexing changes only the query-execution schedule: each question retains its original content, semantic query timestamp, evidence cutoff, and gold answer (see Appendix F.2). Table 7 shows a clear architecture-dependent effect. Relative to batch indexing, dynamic indexing reduces accuracy by 12.71 points for GraphRAG, 3.04 points for EverMemOS, and only 0.12 points for Hindsight. These differences stem from how the systems organize and retrieve memories. Hindsight combines multiple indexing views (temporal, dense, sparse, and graph) and fuses their retrieval results before reranking a large candidate set (300 memory units, see Table 5), which helps mitigate the challenges of indexing streaming data (Appendix E.7). In contrast, GraphRAG's graph-based indexing and EverMemOS's clustering are both sensitive to streaming data (Edge et al., 2024; Hu et al., 2026). EverMemOS reranking stage (Table 5), however, may partly mitigate these effects. It is noted that although reranking helps mitigate these effects, it shifts the burden to the more expensive ranking step. This highlights the need for native indexing strategies (Xu et al., 2025a; Mohoney et al., 2024) that make retrieval more robust to streaming data.

Table 7: Micro QA accuracy (%) under batch and dynamic indexing on the full benchmark.

System	Batch Indexing	Dynamic Indexing
GraphRAG	43.21	30.50
Hindsight	72.10	71.98
EverMemOS	69.08	66.04

3.6 Memory Systems under Different Data Distribution

LifeBench is temporally dense by default. To study the effect of temporal density, we construct dense and sparse variants with the same questions and supporting evidence, varying only the temporal distribution of distractors. Using traces from three users—Sun Yuwei, Yu Xiaowei, and Feng Haoran—we retain 453 questions and 3,965 evidence items in both variants. After removing irrelevant records, we sample non-evidence records as distractors: the dense variant samples records close in time to the evidence, while the sparse variant samples uniformly across each user's year-long trace. The variants are closely matched in scale, containing 11,404 and 11,401 records (7,439 and 7,436 distractors), respectively, with approximately 1.07M tokens each.

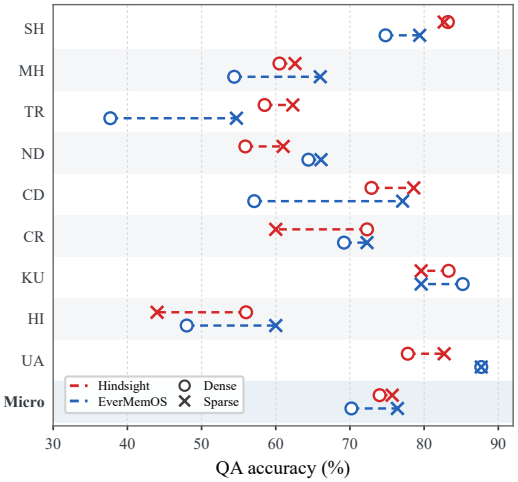

Figure 6: Dense versus sparse-context QA accuracy (%) across all question categories and Micro for 453 questions from three users.

Figure 6 shows that Hindsight outperforms EverMemOS in micro accuracy in the dense setting, broadly consistent with Table 4, but the gap narrows in the sparse setting. Notably, EverMemOS substantially outperforms Hindsight on causal (CR) and hidden-information (HI) questions. *This suggests that, for these question types, the effectiveness of the memory system depends on whether the memory organization method fits the underlying data distribution.* Specifically, these question types require reasoning about event relationships across timestamps via common sense. Hindsight primarily connects observations through semantic, keyword, entity and temporal matching. Dense observations provide richer temporal links for connecting evidence, making Hindsight more likely to retrieve relevant evidence. In the sparse setting, however, these links might become weaker, leading to the drop in CR and HI for Hindsight (compared to Dense). In contrast, EverMemOS clusters observations into memory cells (using LLMs) and merges cells into memory scenes, identifying semantic coherence topics during memory cell construction and long-range patterns during memory scene consolidation. The sparse setting may also facilitate its clustering by providing clearer groupings among observations. Together, these factors allow EverMemOS to better bridge sparse observations across timestamps, facilitating CR, HI and other question types (e.g. TR, MH). *Since real-world data density can vary substantially across users, robust memory systems may benefit from combining complementary strategies for memory retrieval and consolidation.*

4 Conclusion

We presented LifeBench, a synthesis framework for evaluating long-term memory in personalized agents under dense, heterogeneous, and temporally evolving information streams. Building on this framework, we synthesize datasets for evaluating data quality and memory-system performance. Our evaluation of nine representative memory systems shows that leaderboard positions on conversational benchmarks (e.g., LoCoMo) do not necessarily transfer to this setting. Additional takeaways include: (1) while retrieval performance is the dominant bottleneck, linking memory units to raw data is also important for preventing information loss; (2) robustness to dynamic indexing depends on memory structure design and the retrieval pipeline—for example, graph-based indexing degrades substantially under streaming ingestion, while reranking a larger candidate set can mitigate retrieval degradation caused by dynamic updating; and (3) different memory consolidation methods are sensitive to different data distributions, motivating memory management that combines the strengths of extraction-based methods such as Hindsight and abstraction-based methods such as EverMemOS.

Limitation: our synthesis follows a realism-aware design, and our data are evaluated against real-world data. The evaluation cannot cover all aspects of life; thus, we do not claim that LifeBench fully solves this problem. We will, however, publish our code to facilitate future research.

AI USE STATEMENT

We used large language models as components of the LifeBench construction and evaluation pipeline to synthesize persona profiles, event plans, daily activities, and phone records; generate benchmark questions following manually designed rules; and perform automated answer evaluation and semantic alignment of retrieved units with supporting evidence. The procedures and model configurations are described in the main text and appendix. Human oversight included manual filtering, refinement, and validation of generated questions, sampled audits of synthetic records, and validation of the LLM judge against human annotations.

Generative AI was not used to conceive the framework, formulate research hypotheses, design experiments, implement the core methods, nor interpret experimental results. We additionally used generative AI to assist with English translation and language editing, plotting code, figure preparation, and LaTeX formatting. The authors reviewed the AI-assisted materials and take responsibility for the final manuscript, data, code, and conclusions.

REFERENCES

Cognee: Knowledge-graph memory for AI agents, 2026. https://github.com/topoteretes/cognee.

Richard C Atkinson and Richard M Shiffrin. Human memory: A proposed system and its control processes. In *Psychology of learning and motivation*, volume 2, pp. 89–195. Elsevier, 1968.

Alan Baddeley. Working memory: Theories, models, and controversies. *Annual Review of Psychology*, 63:1–29, 2012.

Jiawei Chen, Ruoxi Xu, Boxi Cao, Ruotong Pan, Yunfei Zhang, Yifei Hu, Yong Du, Tingting Gao, Yaojie Lu, Yingfei Sun, Xianpei Han, Le Sun, Xiangyu Wu, and Hongyu Lin. Towards real-world human behavior simulation: Benchmarking large language models on long-horizon, cross-scenario, heterogeneous behavior traces. *arXiv preprint arXiv:2604.08362*, 2026.

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

Martin A Conway and Christopher W Pleydell-Pearce. The construction of autobiographical memories in the self-memory system. *Psychological review*, 107(2):261, 2000.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009. doi: 10.1145/1571941.1572114.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai 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)*, pp. 152–164, August 2024.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. Evermemos: A self-organizing memory operating system for structured long-horizon reasoning. *arXiv:2601.02163*, 2026.

Zhaopei Huang, Qifeng Dai, Guozheng Wu, Xiaopeng Wu, Kehan Chen, Chuan Yu, Xubin Li, Tiezheng Ge, Wenxuan Wang, and Qin Jin. Mem-pal: Towards memory-based personalized dialogue assistants for long-term user-agent interaction, 2025a. URL https://arxiv.org/abs/2511.13410.

Zihan Huang, Tong Li, and Yong Li. Appgen: Mobility-aware app usage behavior generation for mobile users. *IEEE Transactions on Mobile Computing*, 2025b.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, et al. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*, 2025.

Chenlu Ju, Jiaxin Liu, Shobhit Sinha, Hao Xue, and Flora Salim. Trajllm: A modular llm-enhanced agent-based framework for realistic human trajectory simulation. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 2847–2850, 2025.

Chris Latimer, Nicoló Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects, 2025. URL https://arxiv.org/abs/2512.12818.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, et al. Memos: A memory os for ai system. *arXiv:2507.03724*, 2025.

Kaichao Liang, Yuqi Cui, Hao Kong, Xinyuan Huang, Guohaotian Hou, Qingcan Kang, Liang Chen, Yiyang Yin, Ke Ye, Jiaquan Guo, Da Chen, Lingan Zeng, Yixing Peng, Rong Yao, Shixiong Kai, and Mingxuan Yuan. Mindmemos: A portable and self-evolving memory operating layer for ai agents, 2026. URL https://arxiv.org/abs/2608.12428.

Qi Liu, Can Li, and Wanjing Ma. Gatsim: Urban mobility simulation with generative agents. *arXiv preprint arXiv:2506.23306*, 2025.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *The 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 13851–13870, 2024.

MemU. https://github.com/NevaMind-AI/memU.

Jason Mohoney, Anil Pacaci, Shihabur Rahman Chowdhury, Umar Farooq Minhas, Jeffery Pound, Cedric Renggli, Nima Reyhani, Ihab F Ilyas, Theodoros Rekatsinas, and Shivaram Venkataraman. Incremental ivf index maintenance for streaming vector search. *arXiv preprint arXiv:2411.00970*, 2024.

Ilan Naiman, N Benjamin Erichson, Pu Ren, Michael W Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via koopman vaes. In *International conference on learning representations*, volume 2024, pp. 50547–50573, 2024.

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

Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-László Barabási. Returners and explorers dichotomy in human mobility. *Nature communications*, 6(1):8166, 2015. doi: 10.1038/ncomms9166.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

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

Richard M. Ryan and Edward L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1):68–78, 2000. doi: 10.1037/0003-066X.55.1.68.

Chenyang Shao, Fengli Xu, Bingbing Fan, Jingtao Ding, Yuan Yuan, Meng Wang, and Yong Li. Chain-of-planned-behaviour workflow elicits few-shot mobility generation in llms. *arXiv preprint arXiv:2402.09836*, 2024.

Filippo Simini, Gianni Barlacchi, Massimilano Luca, and Luca Pappalardo. A deep gravity model for mobility flows generation. *Nature Communications*, 12(1):6576, 2021. doi: 10.1038/s41467-021-26752-4.

Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. Modelling the scaling properties of human mobility. *Nature Physics*, 6(10):818–823, 2010. doi: 10.1038/nphys1760.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.

Wang-Chiew Tan, Jane Dwivedi-Yu, Yuliang Li, Lambert Mathias, Marzieh Saeidi, Jing Nathan Yan, and Alon Y Halevy. Timelineqa: A benchmark for question answering over timelines. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 77–91, 2023.

Endel Tulving. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53:1–25, 2002. doi: 10.1146/annurev.psych.53.100901.135114.

Alan Geoffrey Wilson. A family of spatial interaction models, and associated developments. *Environment and Planning A*, 3(1):1–32, 1971.

Wendy Wood and David T Neal. A new look at habits and the habit-goal interface. *Psychological review*, 114(4):843, 2007.

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

Gang Xiong, Zhishuai Li, Meihua Zhao, Yu Zhang, Qinghai Miao, Yisheng Lv, and Fei-Yue Wang. Trajsgan: A semantic-guiding adversarial network for urban trajectory generation. *IEEE Transactions on Computational Social Systems*, 11(2):1733–1743, 2024. doi: 10.1109/TCSS.2023.3235923.

Haike Xu, Magdalen Dobson Manohar, Philip A. Bernstein, Badrish Chandramouli, Richard Wen, and Harsha Vardhan Simhadri. In-place updates of a graph index for streaming approximate nearest neighbor search, 2025a. URL https://arxiv.org/abs/2502.13826.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 2025b.

Yuan Yuan, Yuheng Zhang, Jingtao Ding, et al. Worldmove, a global open data for human mobility. *Scientific Data*, 13:549, 2026. doi: 10.1038/s41597-026-06555-2.

Jeffrey M. Zacks and Barbara Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127(1):3–21, 2001. doi: 10.1037/0033-2909.127.1.3.

Jeffrey M. Zacks, Barbara Tversky, and Gowri Iyer. Perceiving, remembering, and communicating structure in events. *Journal of Experimental Psychology: General*, 130(1):29–58, 2001. doi: 10.1037/0096-3445.130.1.29.

Jiarui Zhang. Guided profile generation improves personalization with large language models. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 4005–4016, 2024.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, Zhongzhi Li, Yingying Zhang, Le Song, and Qianli Ma. LifelongAgentBench: Evaluating LLM agents as lifelong learners. *arXiv preprint arXiv:2505.11942*, 2025.

Yu Zheng, Xing Xie, Wei-Ying Ma, et al. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024.

George K. Zipf. The p1 p2/d hypothesis: On the intercity movement of persons. *American Sociological Review*, 11(6):677–686, 1946. doi: 10.2307/2087063.

A Related Work

Agent Memory Existing LLM-agent memory research stems from three interrelated directions. The first benchmarks memory in long-term human–agent interactions: PerLTQA tests personalized QA on dialogue histories (Du et al., 2024), Mem-PAL combines dialogues and application logs (Huang et al., 2025a), and LoCoMo and LongMemEval cover diverse memory-reasoning tasks across multi-session conversations (Maharana et al., 2024; Wu et al., 2025). These benchmarks underpin memory systems including A-MEM, Zep, Mem0 and Hindsight (Xu et al., 2025b; Rasmussen et al., 2025; Chhikara et al., 2025; Latimer et al., 2025). Still, they omit densely interwoven real-world events, center mainly on semantic and episodic memory, and assume user information is fully explicit in dialogue. LifeBench mitigates these gaps for mobile-agent scenarios; Table 1 details comparisons with prior datasets. The second line studies memory for task-oriented agents. Long-horizon tasks demand hierarchical storage, trajectory compression, state tracking and selective retrieval to retain decision-critical experience under context limits, as measured by multi-step benchmarks like WebArena (Packer et al., 2023; Sumers et al., 2023; Zhou et al., 2024). The third explores continual agent memory across task sequences, where LifelongAgentBench assesses knowledge retention, transfer and anti-forgetting capabilities (Zheng et al., 2025).

Personal Data Synthesis Human behavioral traces support applications ranging from personalized services to urban planning and public-health research, yet privacy concerns restrict their collection and sharing. Synthetic data provide a controllable, lower-risk alternative. Early approaches used statistical methods (Zhang et al., 2017) or generative models such as GANs and VAEs (Xiong et al., 2024; Naiman et al., 2024), while recent LLM-based methods generate richer textual and structured personal data (Zhang, 2024; Ju et al., 2025). Most existing work, however, models isolated aspects of behavior—such as user profiles (Zhang, 2024), mobility trajectories (Yuan et al., 2026; Shao et al., 2024; Ju et al., 2025), or app-use sequences (Huang et al., 2025b)—rather than a coherent, user-centric memory. Broader efforts such as TimelineQA and Generative Agents extend the scope to personal timelines and simulated lives, but may rely on rigid templates, lack grounding in realistic temporal and geographic contexts, or cover only short time spans (Tan et al., 2023; Park et al., 2023). In contrast, LifeBench requires no training on real trajectories and simulates coherent year-long lives with persistent identities and cross-source dependencies.

B Data Synthesis Extensions

B.1 Subjective-Agent Simulation

B.1.1 Subjective-Agent Human Simulation Component Design

The Subjective Agent represents a simulated person through seven components that jointly form the environment for daily thinking. These components include both persistent states and query-conditioned memory views. Each is defined by its content, its role in simulation, and its state-transition or retrieval rule. This separation prevents transient affect from altering the immutable self-concept while still allowing recent experiences, unresolved intentions, and behavioral regularities to influence subsequent days. Table 8 summarizes this complete deliberation context.

Psychological basis and operational definitions. The separation of durable knowledge from currently available context is motivated by distinctions between short-term and long-term memory (Atkinson & Shiffrin, 1968) and accounts of working memory (Baddeley, 2012). Episodic memory concerns personally experienced events, whereas semantic memory concerns general knowledge (Tulving, 2002); this distinction describes memory content and is not an additional retention timescale. Accordingly, our long-term summary can contain both personal facts and salient episodes, while episodic retrieval supplies particular past experiences. The component named short-term memory is an engineering view of recent and periodic history: its multi-day window does not model the duration or capacity of human short-term memory. Window lengths, retrieval counts, and daily rewriting rules are simulation choices.

The seven components are organized by their temporal scope and functional role. The self-concept specifies who the person is. The interaction between personal knowledge and current goals is consistent with the self-memory system account of autobiographical remembering (Conway & Pleydell-

Table 8: Components of the Subjective Agent's daily deliberation context.

Component	Information provided	Role in simulation	Transition or retrieval rule
Self-concept	Basic identity, work, family, other life circumstances, and ordinary schedule	Stable identity anchor for all generated decisions	Initialized once and held fixed throughout the simulation
Long-term memory	profile_changes, persistent_patterns, key_memories, and period_summary	Preserves durable changes, habits, relationships, consequential experiences, and the current life phase	Re-consolidated through a bounded end-of-day rewrite; the pattern and key-memory lists contain at most 20 entries each
Short-term memory	Complete activity records from the five most recent days, the same weekday in the previous week, and the same calendar day in the previous month	Supplies recent and periodic temporal context	Refresh short-term memory according to the current time
Episodic memory	Historical daily activity records semantically similar to the current day's atomic events	Supplies analogous experiences that may fall outside the short-term window	Retrieves the three nearest historical daily activity records by embedding similarity
Affective state	Current thoughts, emotion, fatigue, and immediate plans	Carries psychological and physiological continuity into the next decision	Summarize changes based on the current event and update
Intentions and needs	Unfinished goals, pending tasks, and current social, leisure, and physiological needs	Carries unresolved commitments and current demands across days	Regenerated from realized outcomes at day end and supplied to the next day
Behavioral statistics	Rolling 30-day measures such as trip signatures, long-distance travel, number of social activities and evening outings	Constrains schedule generation to preserve habits without forcing repetition	Appends measures computed from the realized itinerary and discards records outside the 30-day window

Pearce, 2000). In our implementation, the initial self-concept is held fixed as a reference profile. For a 34-year-old backend engineer in Hangzhou who commutes by electric bicycle, lives with a teacher spouse, and helps care for a five-year-old daughter, this profile anchors the initial identity and circumstances. Subsequent experiences may be recorded as profile_changes in long-term memory, but they do not overwrite the original self-concept.

Long-term memory retains information expected to remain useful across many days. Its four fields distinguish changes in the person's circumstances, repeatedly observed patterns, dated consequential experiences, and a compact summary of the current life phase. Rather than continually appending text, the agent re-consolidates this object after each realized day. Thus, repeated late work during a project deadline may become a persistent pattern, while a successful release may replace a less salient project event among the bounded key memories.

Short-term memory and episodic memory provide complementary views of the activity history. The short-term view is temporal: it contains the five most recent realized days and, when available, the same weekday in the previous week and the same calendar day in the previous month. The episodic view uses semantic similarity for retrieval rather than maintaining independently updated state: the current day's atomic events query the historical activity repository, which returns the three nearest daily activities.

The remaining components capture the person's mutable condition and behavioral regularities. The affective state represents current thoughts, emotion, fatigue, and immediate plans, whereas intentions and needs preserve unfinished commitments and current social, leisure, and physiological demands. Behavioral statistics provide a separate numerical view of realized mobility over the rolling

30-day window. The first two components are updated through reflection on realized experience; the third is computed deterministically from the final itineraries.

Keeping needs explicit is consistent with motivational accounts in which psychological needs shape behavior; self-determination theory identifies autonomy, competence, and relatedness as basic psychological needs (Ryan & Deci, 2000). Our needs field is broader, also recording physiological and practical demands, and does not implement a fixed ordering of needs. Separately retaining behavioral regularities is motivated by research on habits as learned responses to recurring contexts that interact with current goals (Wood & Neal, 2007). The rolling statistics summarize observed repetition for schedule conditioning.

B.1.2 Subjective-Agent Workflow and Component Transitions

Simulation-start inputs and schedule generation. Before generating day t, the Subjective Agent receives the seven components described above and the current day's atomic events and environmental conditions such as date and weather. The agent first decides how to realize the required atomic events, using retrieved experiences as precedents and pending intentions as cross-day obligations. It then fills open periods with routine, social, leisure, and recovery activities that are consistent with the persona's current needs and recent behavior. The output is a coarse subjective schedule.

Behavioral statistics provide deterministic factual constraints alongside the agent's semantic reasoning. They are computed only from coordinates, durations, distances, and activity sequences in realized itineraries. For example, an uncommon evening restaurant visit slightly changes the rolling 30-day trip-signature distribution and evening-outing count. These statistics can modulate the likelihood of similar future schedules without requiring the model to infer numerical mobility properties from prose. This mechanism balances stable personal habits with day-to-day variation.

Simulation-end reflection and component transitions. After the Objective Agent returns the validated itinerary for day t, the Subjective Agent reflects on the realized outcome rather than the initial plan. It refreshes the affective state, including current thoughts, emotion, fatigue, and immediate plans; rewrites intentions and needs to retain unfinished commitments and incorporate newly created demands; adds the realized day to the activity history used by short-term memory and episodic retrieval; and re-consolidates long-term memory. Independently, a deterministic process computes the day's behavioral measures from the final itinerary and advances the rolling 30-day window.

These transitions produce the persistent state used to construct the deliberation context for day $t + 1$. Each component has a distinct temporal scope: self-concept remains fixed; long-term memory changes through bounded consolidation; short-term memory advances with time; episodic retrieval varies with the query; affective state, intentions, and needs reflect realized outcomes; and behavioral statistics follow a fixed rolling window. Thus, the simulated person can evolve over a year while preserving stable attributes and behavioral regularities that keep the trajectory coherent.

B.2 Objective-Agent Geographic Allocation Algorithm Extension

The Objective Agent converts the Subjective Agent's schedule into a geographically grounded and temporally feasible trajectory. This conversion is separated from schedule generation: the LLM describes the intended place and mobility conditions, whereas the geographic allocator resolves them to registered or map-retrieved entities, selects alternatives, and constructs travel legs. The separation prevents the LLM from inventing plausible-sounding venues or coordinates and makes the locations, routes, and confidence assignments reproducible and auditable.

B.2.1 End-to-End Geographic Allocation Pipeline

Location-intent normalization. As illustrated in Figure 7, the allocation unit is a structured *location intent*: a machine-readable specification of where an activity should occur, rather than a free-form place description or an LLM-generated coordinate. The common schema contains five groups of attributes. (i) *Decision authority* identifies who determines the location. For example, self-chosen or external-unspecified. (ii) *Binding status* indicates whether the location is fixed, preferred, or open. A hospital named in an appointment is fixed. (iii) The *resolution instruction* specifies how the location is to be obtained. For example, the allocator may reuse the registered home,

resolve a named museum through the map service or other ways. (iv) The *eligible-place set* lists the historical or preferred entities that may participate in selection. An intent may permit two previously visited supermarket branches, for instance, without making every venue in the shopping category a valid return candidate. (v) *Geographic and mobility constraints* define conditions such as the target city, distance band, distance sensitivity, and preferred transportation mode; an example is a leisure venue in Hangzhou, 2–5 km from the preceding location, with a preference for short cycling trips. Normalization therefore converts heterogeneous LLM outputs into a finite set of executable location instructions while preserving the original activity semantics.

Instruction routing and direct resolution. The normalized location instruction determines the execution branch. If the intent contains a valid registry identifier, the allocator reuses that entity and its coordinates directly. A fixed name or address is matched against the persona's location registry and then resolved through the map service when no exact registered entity exists. Mandatory anchors such as home, work, and school follow the same identity-preserving path. A micro activity (short-range movement, e.g., walking from the south gate to the north gate of a park) without an independent location instruction inherits its macro-activity anchor when the city is consistent; otherwise, it inherits the preceding location.

Candidate-pool construction. Open, preferred, and self-selected intents enter the candidate-construction branch. The allocator first adds registered and preferred places together with the historical venues selected during location-intent generation. Thus, historical venues are screened for compatibility with the current activity before allocation, and only those explicitly included in the intent's eligible-place set enter the return-candidate pool. It then issues nearby-POI or target-city map queries derived from the activity semantics. The retrieval radius is activity dependent: routine dining and errands emphasize nearby results, whereas medical, leisure, and explicitly long-distance activities permit wider search. Cross-city intents are searched in the destination city rather than around the departure point. When an intent produces several semantic queries, the candidate budget is divided across them so that a single query cannot consume the entire pool.

Candidate responses from the registry and map service are normalized by venue name, alias, address, category, city, and coordinates and then merged. Duplicate records are collapsed using exact identifiers, normalized textual matches, or near-identical coordinates. The remaining candidates are screened for city consistency and basic semantic compatibility and stratified into local, urban, and long-distance bands. The requested band is treated as a soft preference rather than a hard filter, so a sparse retrieval result does not eliminate all feasible alternatives.

Candidate selection and travel-leg construction. After deduplication, a closed candidate set is routed to the selection strategy specified by the intent. A simple open choice uses seeded random selection; a behavior-conditioned choice combines an exploration-and-preferential-return (EPR) mechanism with a gravity-inspired destination score, drawing on established models of individual mobility and spatial interaction (Song et al., 2010; Pappalardo et al., 2015; Simini et al., 2021). For the highest-ranked candidates, the allocator retrieves map-based routes when available, checks travel time against the adjacent activities, and jointly evaluates the venue choice and its effect on the remaining daily mobility budget. The selected venue is then materialized as the final location, and its distance, duration, route, and transportation mode define the incoming travel leg.

Validation and cross-day feedback. Before finalization, an LLM-based validator checks the allocated locations and travel legs against the activity semantics, temporal order and persona constraints. The validator diagnoses an implausible assignment and emits a revised location intent, which may adjust the query, search radius, target distance band, binding strength, or selection strategy. Candidate construction, ranking, and route evaluation are then repeated once under the same deterministic allocation rules. The finalized locations are reconciled with the persistent location registry, allowing today's grounded trajectory to update the familiarity, return eligibility, and mobility profile used on subsequent days.

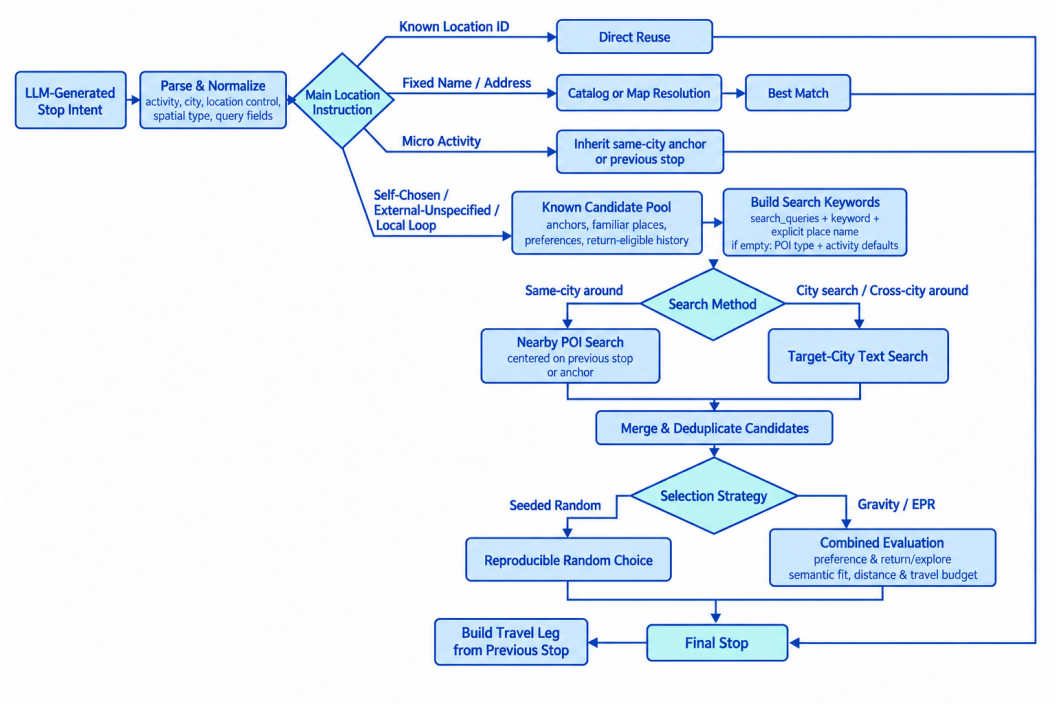

Figure 7: Intent-conditioned geographic grounding and trajectory allocation. An LLM-generated location intent is normalized and routed according to its location instruction. Known and fixed locations are reused or resolved directly; open choices construct and search candidate pools. After candidate deduplication, seeded random choice or gravity/EPR-based evaluation selects the final location, from which the travel leg is built.

B.2.2 Candidate Selection and Joint Trajectory Strategies

Personalized exploration-and-return decision. For every exploration-eligible open intent, the allocator first decides between visiting a new venue and returning to a familiar one. This branch adapts the EPR mechanism introduced for individual human mobility and its later returner–explorer extension (Song et al., 2010; Pappalardo et al., 2015). Its personalized profile is estimated from structured trajectories realized before the current day. The profile tracks the rate of new-place visits, repeated movement patterns, characteristic trip distance, venue frequency and recency, and weekday/weekend and time-of-day regularity. When little history is available, conservative population priors dominate; empirical estimates receive increasing but bounded weight as the trajectory grows.

The exploration probability for activity category a is

$$P_{\text{explore}} = \text{clip}_{[l_a,u_a]}\left(\rho\, c_a\, N^{-\gamma}\, c_{\text{novel}}\, f_{\text{fatigue}}\right),$$

where ρ is the persona's baseline exploration tendency, γ controls exploration decay as eligible return locations grow, N is the number of such locations, c_a is an activity-specific adjustment, c_{novel} represents the current preference for novelty, and f_{fatigue} increases exploration pressure after several repetitive days. The activity-specific bounds $[l_a, u_a]$ prevent both perpetual exploration and complete collapse to familiar venues. When the return set is empty, the decision is routed to exploration.

Preferential return and gravity-based exploration. If the EPR decision is to return, eligible familiar venue j receives

$$w_j^{\text{return}} = v_j^{\eta}\, r_j\, t_j\, s_j,$$

where v_j is its prior visit count, η controls preferential reinforcement, r_j discounts stale visits, t_j rewards consistency with the persona's historical day-type and time-of-day patterns, and s_j measures semantic compatibility with the current activity. This factorization allows a frequently visited place to remain likely without making it universally eligible or ignoring recency and context.

If the decision is to explore, map-retrieved candidate j is evaluated by a gravity-inspired destination-choice score (Simini et al., 2021)

$$w_j^{\text{explore}} = (d_j + d_0)^{-\beta} \exp(-d_j/\kappa)\, a_j\, s_j\, b_j,$$

where d_j is its distance from the preceding location and d_0 prevents singularity at zero distance. The power-law term expresses distance deterrence, while exponential truncation controlled by κ limits frequent long trips. The attraction term a_j is derived from map evidence, s_j measures semantic compatibility, and b_j rewards agreement with the target distance or distance band. Persona-specific β and κ preserve heterogeneous mobility scales and a controlled medium- and long-distance tail.

Seeded weighted selection. Both return and exploration scores define a weighted candidate distribution. For each positive weight w_j, the allocator draws a deterministic pseudorandom value $U_j \sim \text{Uniform}(0, 1)$ from the intent-specific seed and computes

$$k_j = -\frac{\log U_j}{w_j}.$$

Sorting candidates by increasing k_j implements weighted sampling without replacement. This procedure retains behavioral variation across locations and personas while reproducing exactly the same candidate order for the same seed and candidate pool. Intents requesting an unconstrained random choice use the same seeding scheme without the EPR or gravity weights.

Budget-aware joint trajectory allocation. Selecting venues independently can make the complete itinerary infeasible because a choice changes both the current travel leg and the remaining distance and time available later in the day. Locations are therefore materialized sequentially, but each choice jointly considers venue preference, route duration, and the remaining daily mobility target. After map routes are obtained for the highest-ranked candidates, candidate j is evaluated as

$$\text{Score}(j) = \frac{w_j\, \phi_j}{1 + \tau_j/B},$$

where w_j is its semantic-mobility weight, τ_j is the route travel time, B is the applicable travel budget, and ϕ_j measures compatibility with the remaining target ranges for total daily distance and number of distinct locations. The budget is conditioned on the activity type, the time gap between adjacent activities, persona-specific commuting expectations, and mobility willingness. Cross-city activities use their explicit long-distance constraints rather than ordinary commute limits.

Among candidates satisfying the temporal and geographic constraints, the allocator selects the highest joint score. If no candidate satisfies the budget, the allocator returns an explicit infeasibility signal to the downstream coordinator, which may revise the location intent or the surrounding schedule before allocation is attempted again.

Transportation and provenance strategies. Transportation mode is selected from the intent's semantic hint, route distance, time budget, and persona constraints. Walking and cycling are prioritized for short trips, public transport for intermediate trips, and public transport or driving for longer trips, subject to vehicle ownership and habitual mode use. Map-derived durations are stored separately from speed-based estimates, with lower confidence assigned to the latter. Activities that return to their origin, are represented as local loops rather than verified point-to-point routes.

Each selected location and route records its source, map-verification status, and confidence. Finally, selected venues are merged into the registry using normalized names, aliases, full addresses, and near-identical coordinates. Visit frequency and temporal span assign each venue an anchor, routine, occasional, or temporary lifecycle state. These states update its future return eligibility, closing the feedback loop between realized geographic experience and subsequent EPR decisions.

B.3 Question-Type Design Principles and Examples

This appendix expands the taxonomy introduced in Figure 3. Questions are generated after the life events and multi-source records have been instantiated. Answerable questions are linked to the evidence.

Table 9: Design principles and illustrative examples for the nine LIFEBENCH question types.

Question type	Design principle and evidence construction	Illustrative question
Single-hop Retrieval (SH)	Tests whether a system can preserve and retrieve one explicit fact without aggregation or inference.	*Which restaurant did I reserve for dinner on May 18?* One calendar or message record states the venue.
Multi-hop Reasoning (MH)	Requires combining complementary facts distributed across at least two records, dates, or sources.	*Which cities did I visit during the National Day trip?* Transport, hotel, and photo records jointly supply the complete list.
Causal Reasoning (CR)	We extract causal relationships between events to generate questions using the event decomposition tree from the Hierarchical Outline Planning process. The system is required to infer the cause or outcome of an event, rather than merely retrieving co-occurring facts.	*Why did I cancel Saturday's hike?* A weather alert, a subsequent message, and the cancelled calendar entry establish the causal chain.
Conflict Detection (CD)	Introduces incompatible claims through plans, recollections, noisy records, or later corrections. The valid answer is determined by source reliability, actual-event evidence, chronology, or an explicit correction.	*At which venue did the dinner actually take place?* An early invitation names venue A, but a later correction and geotagged photo support venue B.
Temporal Reasoning (TR)	Targets ordering, interval calculation, frequency, duration, and relations among timestamped observations. Questions are constructed so that lexical matching alone does not reveal the requested temporal relation.	*How many days after registering for the race did I complete it?* The registration and race records provide the two endpoints.
Knowledge Update (KU)	Tracks a user attribute or state that changes over time. Multiple values may all have been true, but the answer must select the value valid at the query time and must not use a later update.	*Which gym membership was active at the end of October?* Records contain the original plan, an upgrade, and a later cancellation.
Non-declarative Memory (ND)	It requires modeling the user's habits, preferences, behavioral intensity, regularity, or behavioral changes based on long-term observations of the user's daily behaviors, rather than explicit self-reports from the user, with relevant evidence spanning weeks or months.	*How did my running routine change from spring to summer?* Repeated health, calendar, and photo records reveal the shift in frequency and distance.
Hidden-information Mining (HI)	Requires inferring an unstated situational fact from one or several complementary observations. Unlike ND, the target is a specific latent fact rather than a recurring long-term pattern, and the evidence may be lexically dissimilar to the question.	*Can I go and water my neighbor's flowers recently?* The user is away from the residence in the near term, so this is not possible.
Unanswerable (UA)	Ask for a specific event detail or content that has not been mentioned.	*What gift did I buy for Li Ming on July 9?* The history mentions the birthday and shopping plan but contains no completed purchase or gift identity.

Table 10 compares the benchmarks in Table 1 using the original task definitions of PerLTQA (Du et al., 2024), LoCoMo (Maharana et al., 2024), LongMemEval (Wu et al., 2025), Mem-PAL (Huang et al., 2025a), and PersonaMem-v2 (Jiang et al., 2025). The projection is our qualitative analysis of overlapping question demands, not a claim that these benchmarks use our labels or evidence protocol. "Partial" indicates a related capability with a different output or evidence type. The last column lists only QA tasks or types without a direct LifeBench counterpart; a dash means the work declares no separate uncovered QA type.

Table 10: QA tasks in Table 1 and their qualitative projection onto LifeBench question types. Abbreviations follow Figure 3.

Benchmark	Declared QA types or tasks	LifeBench projection	QA types not covered by LifeBench
PerLTQA	Memory classification Memory retrieval Memory fusion (QA)	SH .	—
LoCoMo	Single-hop Multi-hop Temporal reasoning Open-domain knowledge Adversarial	SH, MH, TR, UA.	Open-domain knowledge.
LongMemEval	Single-session-user Single-session-assistant Single-session-preference Multi-session Knowledge-update Temporal-reasoning Abstention	SH, MH, KU, TR, UA.	—
Mem-PAL	Requirement Restatement Solution Proposal Multi-turn Dialogue Interaction	HI, ND (partial).	Requirement Restatement Solution Proposal
PersonaMem-v2	Implicit-preference QA (MCQ / open-ended)	ND, KU (partial).	—
LIFEBENCH (ours)	SH, MH, CR, CD, TR, KU, ND, HI, UA.	SH, MH, CR, CD, TR, KU, ND, HI, UA.	—

From the comparison results in Table 10, we observe that LifeBench is currently the benchmark with the most comprehensive and diverse set of question categories, capable of comprehensively evaluating various capabilities of memory systems. Furthermore, benefiting from its rich textual content and complete representation of personal life trajectories, LifeBench allows researchers to design and generate more types of QA tasks or iteratively create more challenging QA instances within existing categories without modifying the underlying dataset.

C SYNTHESIS DATA STATISTICS

C.1 LIFEBENCH-BEIJING

LIFEBENCH-BEIJING is a dedicated subset for mobility realism evaluation. LIFEBENCH-BEIJING contains five independent personas, yielding 295 exported user-days; 265 of these days include all 48 half-hour slots within the Beijing grid and are adopted in the primary mobility comparison experiments.

LIFEBENCH-BEIJING is not trained or calibrated on these real trajectories: its movements arise from persona constraints, commonsense knowledge, geographic algorithms, and simulated behavior. Table 11 summarizes the five personas, and Figure 8 shows one sampled day for each.

C.2 LIFEBENCH-GENERAL

This section reports descriptive statistics for LIFEBENCH-GENERAL. The dataset contains ten simulated individuals, each with a year-long life stream. Table 12 summarizes the scale of the personas, social graphs, daily events, questions, and observable phone traces. The persona schema exposes the core identity and background attributes listed in Tables 12 and 13.

Table 14 gives the category-level composition of the QA and phone-record layers in LifeBench. Question types are multi-label: one QA pair may belong to several categories, and the integer-valued `category` codes map one-to-one to the raw string-valued `question_type` labels. The

Table 11: The five-person LIFEBENCH-BEIJING dataset used for mobility evaluation. Home and work locations are reported at district level; for the student, the work district denotes the primary study location. "Places" counts distinct locations visited in the trajectory traces.

Name	Gender	Age	Home district	Work district	Job	Family	Places
Cheng Yining	Female	36	Chaoyang	Chaoyang	Media-company producer	Married; one preschool child; shares caregiving with her spouse	103
Chen Yutong	Female	20	Fengtai	Haidian (study)	University student	Parents run a small hardware store in Laishui; one younger brother; lives in a university dormitory during term	144
Chen Mo	Male	29	Haidian	Haidian	AI product manager	Unmarried; shares an apartment with a university classmate	96
Lin Xiaomei	Female	32	Tongzhou	Chaoyang	General-hospital emergency nurse	Married; no children; rents a two-bedroom apartment with her spouse	112
Wang Zhiqiang	Male	47	Xicheng	Xicheng	Salesperson	Married; lives with his spouse and high-school-aged child	111
Sum of person-level unique places							566
Cross-person union of unique places							560

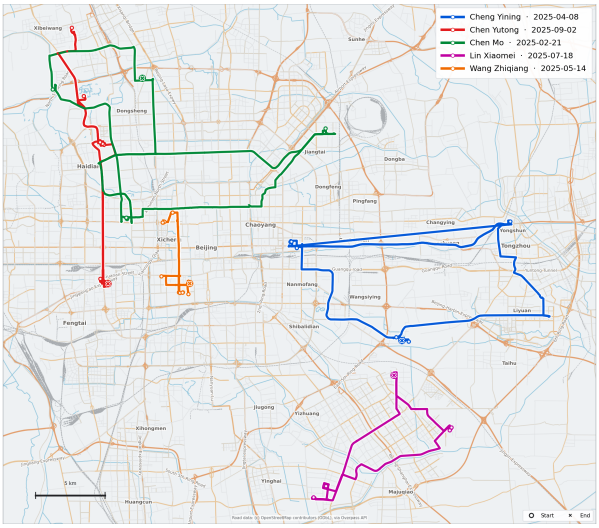

Figure 8: Randomly sampled one-day mobility trajectories for the five Beijing personas. The sampled dates are April 8, 2025 (Cheng Yining), September 2, 2025 (Chen Yutong), February 21, 2025 (Chen Mo), July 18, 2025 (Lin Xiaomei), and May 14, 2025 (Wang Zhiqiang). Colors identify personas; circles and crosses mark the start and end of each daily trajectory, respectively.

Table 12: Summary statistics of LifeBench. "Per day" averages use the 3,656 simulated person-days.

Component	Value
Simulated individuals	10
Persona tokens	12,696.6 per person
Social relationships	14.1 per person
Daily events	5,748.6 per person; 15.72 per day
QA pairs	3,380 total
Phone records	47,409 total; 12.97 per day
Phone-record tokens	1,313,336.1 per person
Data-synthesis time	7.98 h per person on average
Data-synthesis token consumption	286M tokens per person (daily-life and phone data)

table uses the normalized display names adopted throughout the paper. Consequently, the category counts sum to 5,738, exceeding the 3,380 unique QA pairs because of label overlap. Phone-record categories, by contrast, partition the 47,409 observable records.

D DATA EVALUATION DETAILS

Evaluating synthesized personal-life data is difficult without complete real-world life histories for comparison. Real-world observations typically cover only one city, population, or behavioral modality. We therefore combine four complementary evaluations: population statistics provide behavioral references, mobility metrics quantify agreement with real trajectories, automated checks scale to the full synthetic corpus, and human review covers failures requiring commonsense or long-range narrative context.

Table 13: Key persona fields for the ten simulated individuals in LifeBench. All salary values are reported in CNY per year.

Sample	Name	Gender	Age	Education	Job	Hobbies	Salary	Family
fenghaoran	Feng Haoran	Male	27	Bachelor's degree	Financial investment adviser	Street photography; morning running; outdoor camping; ...	200,000	Unmarried
leimingxuan	Lei Mingxuan	Male	31	College diploma	Clothing-store owner	American pool; recreational roller skating; ...	150,000	Unmarried; lives with his parents
lumingqiang	Lu Mingqiang	Male	35	Junior middle school	Electrical machinery installer and repairer	Squash; running; physical exercise; ...	40,000	Married; two children
maxiulan	Ma Xiulan	Female	41	Junior middle school	Homemaker and part-time worker	Embroidery and knitting; housework; ...	5,000	Married; three children
songyajing	Song Yajing	Female	32	College diploma	Boutique owner and clerk	English-language music; shopping; meeting friends; ...	80,000	Married; one child
sunyuwei	Sun Yuwei	Female	22	University student (Chinese)	Student and freelance writer	Reading; contemporary pop music; journaling; ...	80,000	Unmarried
yemingxuan	Ye Mingxuan	Male	26	Bachelor's degree	Mechanical engineer	Woodworking; collecting modern coins;...	200,000	Unmarried
yinhao	Yin Hao	Male	24	Secondary vocational school	CNC machine operator	Morning running; freshwater fishing; camping; ...	120,000	Unmarried
yuxiaowei	Yu Xiaowei	Female	35	General high school	Insurance agent	Yoga; marathons; meeting friends; ...	300,000	Married; two children
yuxiaowen	Yu Xiaowen	Female	24	Bachelor's degree	Resident physician	City walks; reading; European classical music; ...	120,000	Unmarried

Table 14: Category counts in LifeBench. QA labels are non-exclusive; phone-record categories are mutually exclusive.

Code	QA category	Count	Phone-record category	Count
0	Single-hop Retrieval (SH)	1,611	`agent_chat`	8,177
1	Temporal Reasoning (TR)	530	`note`	7,836
2	Multi-hop Reasoning (MH)	1,106	`push`	6,026
3	Non-declarative Memory (ND)	526	`calendar`	6,011
4	Unanswerable (UA)	585	`sms`	5,923
5	Causal Reasoning (CR)	373	`photo`	5,604
6	Hidden-information Mining (HI)	194	`call`	3,947
7	Conflict Detection (CD)	424	`fitness_health`	3,641
8	Knowledge Update (KU)	389	`contact`	244
Unique QA pairs		3,380	Total phone records	47,409

D.1 Population-Level Behavioral Evaluation

Evaluation sample construction. For each synthetic dataset, we randomly select one user and extract a continuous evaluation window of up to 90 days. For LifeBench, the selected sample is Feng Haoran, covering September 19–December 18, 2025. For Mem-PAL, the selected sample is Li Xiaofeng, covering August 11–November 8, 2024. We retain the timestamped daily events and associated phone-interaction records within each interval and apply the same interest-extraction pipeline to both samples.

Interest extraction and temporal visualization. We follow the protocol introduced in Section 3.3 of OmniBehavior (Chen et al., 2026). OmniBehavior interleaves timestamped interactions from five Kuaishou scenarios into a chronological history for each user, covering 90 days (September 1–November 30, 2025). Following its Section 3.1, DeepSeek-V4-Flash assigns one interest category and three descriptive keywords to every interacted item. The drift analysis uses the category label: fine-grained labels are mapped into a shared set of macro-level categories so that semantically equivalent interests expressed in different scenarios or datasets are counted together.

For visualization, interactions are first grouped by user, calendar day, and macro-level category. Let $n_{u,t,c}$ be the number of interactions of user u assigned to category c on day t. We represent the daily composition of interest c by

$$p_{u,t,c} = \frac{n_{u,t,c}}{\sum_j n_{u,t,j}}.$$

The horizontal axis in Figure 4 is the day index in the approximately three-month window, and each curve traces one of the most prominent categories for the user. The vertical axis is the category's share of the user's interests on that day. The paper (Chen et al., 2026) describes the curves as temporal interest distributions. For the LifeBench and Mem-PAL panels, we apply the same extractor and category taxonomy, use comparable 90-day windows, retain the five categories with the largest total mass, and plot a seven-day rolling mean of $p_{u,t,c}$ to suppress daily noise. Thus, the figure supports comparison of temporal patterns—persistence, emergence, decay, recurrence, and abrupt bursts—rather than comparison of category names across different users.

Interest-drift metric. The scalar drift score is computed from category presence rather than from the smoothed curves. Let $C_{u,t} = \{c : n_{u,t,c} > 0\}$ denote the set of macro-level interest categories observed for user u on day t. Following OmniBehavior, drift between two consecutive days is the Jaccard distance between their category sets:

$$d_{u,t} = 1 - J(C_{u,t-1}, C_{u,t}) = 1 - \frac{|C_{u,t-1} \cap C_{u,t}|}{|C_{u,t-1} \cup C_{u,t}|}.$$

A value of 0 means that the two days contain exactly the same category set, whereas a value of 1 means that they share no category. We average $d_{u,t}$ over valid consecutive-day pairs and then over users to obtain the dataset-level drift rate. When both days contain no categorized event, the pair is undefined and excluded; when only one day is empty, its overlap is zero and the drift is one. This convention is important for sparse synthetic histories, because treating missing days as stable interests would artificially reduce their drift.

Using OmniBehavior's real-user drift rate of 0.63 as a reference, LifeBench shows the smallest gap (0.09), preserving day-to-day variation while maintaining overlapping interests over time. LoCoMo's lower drift rate (0.17) indicates limited changes in its active categories across adjacent days, with occasional task-driven spikes failing to produce sustained evolution. Conversely, Mem-PAL exceeds the real-user reference by 0.18, indicating excessive category turnover, consistent with its sparse, abrupt curves.

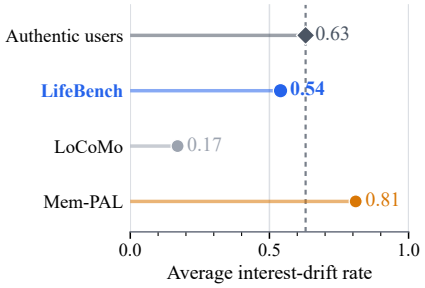

Figure 9: Average interest-drift rates. The dashed line marks the authentic-user reference.

D.2 Mobility Evaluation

Real-world data typically covers only specific content or population groups. We therefore compare LifeBench-Beijing with three real Beijing references and two synthetic baselines to evaluate the plausibility of its movement trajectories. Appendix C.1 describes the subset construction, personas, and sample trajectories; this section specifies the reference sources, unified processing, metrics, and comparison protocol. The comparisons examine individual movement geometry, dwell duration, and location use.

D.2.1 Data Sources and Unified Processing

Synthetic Data Benchmarks. We compare LifeBench-Beijing with two synthetic baselines for Beijing mobility trajectories: WorldMove and CoPB. WorldMove uses a trained diffusion-based mobility generator and public city features; we use its released Beijing dataset (Yuan et al., 2026). CoPB generates activity intentions with an LLM-based workflow and maps them to locations with a mobility model; we use its released Beijing trajectories. Its official pipeline includes Tencent and ChinaMobile data-specific training and generation routes, although the released trajectories do not identify their precise generation branch or checkpoint (Shao et al., 2024). Results against those two references can therefore reflect domain overlap or tuning.

Real reference datasets. We select three open-source, high-quality real-human mobility trajectory datasets as references to evaluate the quality of synthetic data. ChinaMobile (Shao et al., 2024) comprises Beijing cellular-location observations with stable user identifiers: 1,246 users and 4,163,651

raw records from July–August 2017. It is useful for daily displacement and activity range, but cell-derived positions are less precise than GPS and observation is incomplete. Tencent POI (Shao et al., 2024) comprises 4,333,091 published place-stay intervals from 25,579 users in October–December 2019; gaps between stays are unobserved, so it is a sparse POI-visit reference rather than a complete movement trace. GeoLife supplies GPS trajectories with stable user identities (Zheng et al., 2010); its raw release contains 182 users, 18,670 files, and 24,876,978 points. We retain Beijing observations. GPS recording is user-initiated and concentrated around trips. After common-grid filtering and a minimum of four valid half-hour slots per day, the three references contain 56,261, 1,580,003, and 7,047 user-days, respectively. Their mean daily observed-slot counts are 15.06, 23.02, and 9.05.

Common representation. Every source is converted to Beijing local time and the same 82×155 approximately 1-km Beijing grid, with 48 half-hour slots per day and -1 for missing locations. LIFEBENCH-BEIJING uses its final reconciled WGS-84 coordinates, converting from GCJ-02 when needed; WorldMove's 165,328 released Beijing days already use the target slots and grid. CoPB activity intervals are assigned only to slot centers that they cover, leaving inter-activity gaps missing. For ChinaMobile and GeoLife, the point nearest each slot center is retained without interpolation; for Tencent, a slot is populated only when its center lies inside a published stay interval. Coordinates from the three references and CoPB are mapped to the nearest grid center only if it is within 3 km. No movement is inferred across missing slots or across midnight.

Comparison protocol. Of 295 exported LIFEBENCH-BEIJING days, 265 have all 48 slots within the Beijing grid and enter the main comparison. For each of 50 seeds (100–149), we sample 265 user-days from each reference and baseline. The primary comparison uses only slots jointly observed in a real day and its matched synthetic day; real days and CoPB require at least four valid slots. Thus, scores are compared across methods *within* each reference, not across references with different observation mechanisms. Descriptive statistics instead use all valid days without equal-size sampling or coverage matching.

D.2.2 MOBILITY METRICS AND DISTRIBUTIONAL COMPARISONS

Distances use Haversine $d(\cdot, \cdot)$ between representative grid-cell coordinates, with an Earth radius of 6371.0088 km. A valid slot has an observed grid cell; missing slots do not create inferred movement.

Jump length. For consecutive valid slots x_t and x_{t+1} in different cells, jump length is $\Delta r_t = d(x_t, x_{t+1})$. Pairs located within the same cell, pairs separated by a missing slot, and pairs where the two slots span midnight are excluded. This measures the distance of an observed move, conditional on movement occurring.

Daily travel distance. For each user-day, daily travel distance sums jumps, $r_w = \sum_t \mathbf{1}[x_t \neq x_{t+1}] d(x_t, x_{t+1})$, over consecutive valid slot pairs. Under coverage matching, the real day's missing-slot mask is applied to each synthetic day before this sum is computed.

Radius of gyration. Radius of gyration summarizes the spatial extent of a user-day:

$$r_g = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|r_i - r_c\|^2}, \qquad r_c = \frac{1}{N} \sum_{i=1}^{N} r_i,$$

where r_i is the representative coordinate of valid slot i and N is the number of valid slots. Repeated occupancy weights the center by time spent in a cell.

Dwell time. Dwell time Δt is the duration of a maximal run of consecutive valid slots in one cell. A run of m half-hour slots contributes $30m$ minutes. Missing slots break a run, so the measure does not assume that a user remained at an unobserved location.

Daily visited locations. The daily visited-location count S_d is the number of distinct valid grid cells observed in a user-day. It counts cells rather than semantic venues: distinct venues inside one cell are merged, and a venue spanning cells can contribute more than one cell.

Jensen–Shannon divergence. For each scalar metric, real and synthetic samples are binned with the same edges and normalized into probability vectors P and Q. We use 1-km bins for jump length and radius of gyration, 5-km bins for daily travel distance, 30-minute bins for dwell time, and integer bins for daily visited locations; The long tails of distance distributions are retained within overflow bins. With $M = (P + Q)/2$, we compute

$$\mathrm{JSD}(P, Q) = \frac{1}{2} \sum_{k:P_k>0} P_k \log_2 \frac{P_k}{M_k} + \frac{1}{2} \sum_{k:Q_k>0} Q_k \log_2 \frac{Q_k}{M_k}.$$

Base-2 JSD lies in $[0, 1]$, with 0 indicating identical distributions. The comparison applies each real day's visibility mask to the matched synthetic day before computing JSD; lower values indicate closer agreement. We report full-sample means and medians for the five spatial and temporal scalar metrics and observed-slot count; these are descriptive, not substitutes for coverage-matched JSD.

D.2.3 Results and Conclusions

Coverage-matched individual statistics. Table 15 shows that LifeBench-Beijing is competitive with mobility generators trained or calibrated on real trajectory data, although LifeBench-Beijing does not use real mobility trajectories to train its generation pipeline. It has the lowest jump-length and daily-distance JSD against ChinaMobile (0.0679 and 0.1582) and the lowest jump-length, daily-distance, and radius-of-gyration JSD against GeoLife (0.0618, 0.2194, and 0.0828). These measures capture the geometry of individual movement: the length of a move, the distance traveled in a day, and the size of the daily activity space. Their agreement across two independently collected references supports the plausibility of LifeBench-Beijing's synthesized daily events and the locations assigned to them. For Tencent POI, CoPB has a Tencent-specific training route, while the released trajectories do not identify the branch used. Setting CoPB aside for this comparison, LifeBench-Beijing achieves lower JSD than WorldMove for radius of gyration (0.1428 versus 0.1659), dwell time (0.1294 versus 0.2287), and daily visited locations (0.3195 versus 0.4196). Figure 10 visualizes this pattern alongside the complementary strengths of the baselines.

Full-sample descriptive scales. Table 16 provides a complementary view of movement scale. LifeBench-Beijing offers the most balanced match to ChinaMobile and GeoLife in activity-space size, typical jump length, and daily location count: its mean radius of gyration is 4.43 km versus 4.61 and 4.53 km, its median jump length is 3.14 km versus 1.89 and 3.23 km, and it visits 6.57 cells per day on average versus 5.78 and 5.23. Its mean daily distance is higher, however, at 29.45 km versus 24.50 and 21.28 km. WorldMove produces a much wider activity space (mean radius 8.97 km), more daily locations (11.28), and a heavy long-distance tail (mean daily distance 74.20 km versus median 16.74 km); these patterns resemble a more travel-intensive setting than the two references. CoPB has the opposite tendency: its median daily distance is zero and it visits only 2.40 cells per day on average. Its low-movement structure closely matches Tencent's sparse POI stays, yielding the lowest JSD on four Tencent metrics in Table 15. Because CoPB's pipeline includes Tencent and ChinaMobile-specific training or calibration routes, its Tencent result does not by itself demonstrate independent generalization. The full-sample scales remain descriptive because the references differ substantially in observation coverage.

Overall, LifeBench-Beijing's life-event-driven generation produces daily trajectories with human-like movement geometry across multiple real references, while maintaining plausible activity-space size and location use. Its main remaining discrepancy in these comparisons is elevated mean daily travel distance. The distributional and descriptive results together support the realism of the generated daily-life behavior and its geographic grounding.

D.3 Automated Quality Evaluation

We evaluate all users in LifeBench-General using the method described below.

Relation consistency. First, we construct a structured relation knowledge base for each persona from the initial profile, including its predefined social network. We then traverse the generated events day by day in chronological order. For each event, participant mentions are converted into relation cases and assessed against the relations defined in the day-start knowledge base using the

Table 15: Coverage-matched JSD to each real reference (mean over 50 samples of 265 days; lower is better). Bold marks the lowest JSD within a reference and metric.

Metric	ChinaMobile			Tencent POI			GeoLife		
	LifeBench-Beijing	WorldMove	CoPB	LifeBench-Beijing	WorldMove	CoPB	LifeBench-Beijing	WorldMove	CoPB
Jump length	**0.0679**	0.2408	0.1544	0.2781	**0.1467**	0.3250	**0.0618**	0.3696	0.2726
Daily distance	**0.1582**	0.2381	0.2031	0.3732	0.2991	**0.0212**	**0.2194**	0.3166	0.2342
Radius of gyration	0.1088	0.2030	**0.0820**	0.1428	0.1659	**0.0457**	**0.0828**	0.2663	0.1617
Dwell time	0.0480	**0.0214**	0.0912	0.1294	0.2287	**0.0054**	0.1128	**0.0663**	0.1937
Daily visited locations	0.1098	**0.0707**	0.2497	0.3195	0.4196	**0.0565**	0.2025	**0.1124**	0.2645

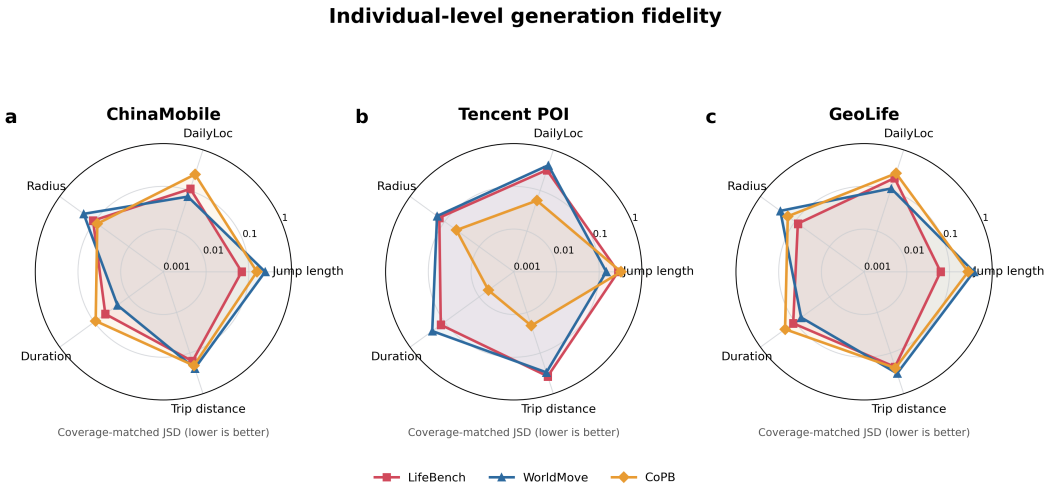

Figure 10: Individual-level fidelity against three real references. The radar labels Jump length, DailyLoc, Radius, Duration, and Trip distance correspond to Jump length, Daily visited locations, Radius of gyration, Dwell time, and Daily distance in Table 15, respectively. All axes show coverage-matched JSD on a logarithmic radius; closer to the center is better.

current event and source events that occurred earlier on the same day; future events are excluded from the evaluation context.

An LLM-based selector assigns each relation case to a structured decision state. Its output is subsequently checked by a programmatic validator that enforces complete case coverage, valid person identifiers, admissible evidence references, route-specific constraints, and same-day consistency.

Table 17 summarizes how different relation cases are interpreted and whether they modify the maintained relation state.

Participant-level decisions are aggregated into an event-level label using the priority

$$\textsc{Nonconform} > \textsc{Unknown} > \textsc{Conform} > \textsc{Not-Applicable}.$$

An event is nonconforming if it contains either a sudden-familiar conflict or an unexplained relation change. Unknown and non-applicable events are retained as separate outcomes rather than being forced into a binary judgment.

Let N_{conform} and $N_{\text{nonconform}}$ denote the numbers of conforming and nonconforming events. The relation-consistency rate is defined as

$$R_{\text{rel}}(\%) = 100 \times \frac{N_{\text{conform}}}{N_{\text{conform}} + N_{\text{nonconform}}},$$

with unknown and non-applicable events excluded from the denominator.

After each day, the relation knowledge base is dynamically updated. Only evidence-supported additions and updates from source events are admitted; they are encoded as restricted patches, programmatically validated, and committed atomically at day's end to form the knowledge base for the next day. This allows the evaluation to track evolving relationships without using future information.

Table 16: Full-sample mobility statistics. Each paired entry gives mean / median; the final column gives the mean number of valid half-hour slots per day. These descriptive values do not use equal-size sampling or coverage matching.

Dataset	Jump length (km)	Daily distance (km)	Radius of gyration (km)	Dwell time (min)	Daily visited locations	Mean valid slots/day
LIFEBENCH-BEIJING	3.93 / 3.14	29.45 / 24.29	4.43 / 4.63	164.21 / 30	6.57 / 7	46.52
WorldMove	5.36 / 1.15	74.20 / 16.74	8.97 / 1.77	96.99 / 30	11.28 / 10	48.00
CoPB	4.42 / 2.29	1.81 / 0.00	2.62 / 1.74	222.54 / 180	2.40 / 2	26.03
ChinaMobile	4.16 / 1.89	24.50 / 16.22	4.61 / 3.17	41.34 / 30	5.78 / 5	15.06
Tencent POI	1.83 / 1.23	0.19 / 0.00	2.13 / 0.49	250.45 / 180	1.75 / 2	23.02
GeoLife	5.04 / 3.23	21.28 / 14.39	4.53 / 3.72	38.73 / 30	5.23 / 5	9.05

Table 17: Decision rules for temporal relation-consistency evaluation.

Decision	Trigger	Operation
Existing relation	Known person; no unsupported relation change	Keep
Supported evolution	Explicit relation establishment or extension	Add / Update
Temporary contact	One-off interaction without a persistent relation	No change
Relation conflict	Sudden familiarity or unexplained identity/relation change	Flag nonconformity
Insufficient evidence	Identity or relation cannot be resolved without speculation	Mark unknown

Event-type diversity. Event-type diversity is evaluated from the aggregate distribution of generated event categories. Let n_i denote the number of events assigned to category i, let $N = \sum_i n_i$, and let $p_i = n_i/N$ for the K observed categories. We report normalized Shannon entropy and the Simpson diversity index:

$$H_{\text{norm}} = -\frac{\sum_{i=1}^{K} p_i \log p_i}{\log K}, \qquad S = 1 - \sum_{i=1}^{K} p_i^2.$$

Normalized Shannon entropy measures the evenness of the event-category distribution relative to the maximum entropy over K categories. The Simpson diversity index represents the probability that two sampled events belong to different categories. Higher values indicate broader coverage and lower concentration, but neither metric evaluates the semantic correctness of individual events.

D.4 HUMAN EVALUATION

We sample two continuous months of daily-activity traces and associated phone-device records from three users. Table 18 summarizes the scale and time range of each sampled subset. Two independent reviewers annotate each user's sampled data. The human evaluation consists of two parts: the **rationality audit** and the **diversity rubric**.

Rationality audit. Reviewers inspect each daily-activity record along four dimensions: temporal coherence, event content, location, and participating persons. An activity receives a binary "valid" label **only if all four dimensions are plausible and mutually consistent**. Each phone record is mapped to the daily-activity trace within its corresponding time window. A phone record counts as valid iff: (i) the phone-device operation is semantically consistent with the matched real-world activity; and (ii) the phone-record content itself is realistic and plausible. For a given record category x, validity is defined as:

$$\text{Validity}_x = \frac{N_{\text{valid},x}}{N_{\text{reviewed},x}}.$$

This strict conjunctive rule rejects records that are partially reasonable yet contain contradictions, e.g., reasonable content paired with conflicting timestamps or participant information. For each

reviewer-user entry in Table 19, $N_{\text{reviewed},x}$ is the total number of inspected records of category x, and $N_{\text{valid},x}$ is the subset judged valid.

Table 18: Statistics of the subsets sampled for human evaluation. Activity and Phone give the numbers of reviewed activity and phone records, respectively. Start and End denote the first and last dates of each sampled evaluation window; all dates are in 2025.

User	Activity count	Phone count	Start	End
Yu Xiaowei	986	697	Sep. 1	Oct. 31
Feng Haoran	926	780	Jan. 1	Feb. 28
Sun Yuwei	979	840	May 1	Jun. 30

Table 19: Human-evaluation scores for the three sampled users. Activity and Phone report validity rates, and Diversity is scored out of 10. For each user, the first and second rows correspond to Reviewers 1 and 2, respectively.

User	Activity validity	Phone validity	Diversity (/10)
Yu Xiaowei	99.2%	98.0%	7
	99.5%	97.8%	6.5
Feng Haoran	97.6%	89.9%	8
	97.7%	91.2%	8.5
Sun Yuwei	99.5%	97.1%	8
	99.3%	98.3%	8

Diversity rubric. Reviewers score each user trace against five criteria. Each criterion is scored 0 (unsatisfactory), 1 (partial fulfillment), or 2 (strong fulfillment):

1. Activities differ meaningfully across users and manifest user-specific characteristics;
2. Activities from the same user exhibit day-to-day variation and avoid homogeneous repetition;
3. User-specific behaviours are balanced against common and occasional events, without rigid persona stereotypes;
4. Fine-grained attributes including sleep duration, body weight, diet, and exercise show realistic fluctuations;
5. The overall trace captures real-world complexity and unpredictability while maintaining internal coherence and avoiding obvious inconsistencies.

Summing the five item scores yields a maximum total of 10 points per reviewer-user assessment. Individual scores from the two reviewers are presented separately in Table 19.

Interpretation of human-evaluation results. Daily-activity validity and phone-record validity are *record-level fractional metrics* measuring the share of internally consistent entries. By contrast, diversity is an *ordinal rubric-based rating* and should not be interpreted as a probability or proportion. Note that this evaluation constitutes a *sampled audit* rather than exhaustive full-dataset annotation. All conclusions drawn apply only to the two-month sampled windows listed in Table 18. The human evaluation serves as complementary sanity checks that supplement our suite of automated evaluation metrics.

D.4.1 Case Study

The following examples illustrate representative failures identified during human evaluation: although we observe no semantic contradictions or severe logical errors, occasional behavioral distortions still arise.

Case 1: Face-to-Face Communication Recorded as an SMS. The daily activity explicitly describes an in-person conversation between Feng Haoran and his mother after he arrives home. However, the phone record represents the same conversation as an incoming SMS from his mother at 11:05. Although the message is semantically consistent with the activity, the use of remote communication is implausible in this face-to-face setting without an additional explanation.

Raw data: daily activity and phone record

Daily Activity (January 28, 11:00–11:30). "Greeting his parents and checking on his mother." Upon arriving home, Feng Haoran was greeted by his mother, Li Xiuying, who came out of the kitchen. He noticed that she looked pale and more tired than the previous year. She mentioned that his younger sister would arrive soon and that someone needed to pick her up.
Phone Record (SMS, 11:05). Contact: Li Xiuying (Mother).
Direction: Received.
Message: "I'm a little tired. Your sister will arrive soon. Go pick her up."
Human Judgment: Semantic consistency: Pass; Content plausibility: Fail.

Case 2: Future Outcomes Included in an Earlier Calendar Entry. The calendar entry was created at 09:00, one hour before the scheduled activity. However, its description already includes the client's positive feedback and the supervisor's verbal praise, both of which occurred after 10:00. While scheduling the meeting in advance is reasonable, recording its specific outcomes before they occur introduces a temporal inconsistency.

Raw data: daily activity and phone record

Daily Activity (January 17, 10:00–10:30). "Client Lin Zhihao's feedback and reporting." At 10:00, Feng Haoran received a WeChat message from his client Lin Zhihao, who highly praised the annual asset allocation proposal. Feng then reported the feedback to his supervisor, Qian Yongqiang, who verbally praised his work.
Phone Record (Calendar).
Created at: January 17, 09:00.
Scheduled time: January 17, 10:00–10:30.
Title: "Client Lin Zhihao's feedback and reporting."
Description: "10:00: Positive WeChat feedback from client Lin Zhihao; immediately reported to supervisor Qian Yongqiang and received verbal praise. Source: WeChat conversation and in-person discussion."
Human Judgment: Semantic consistency: Pass; Content plausibility: Fail.

E Main-Text Experiment Details

This section presents the detailed results of the main experiments in this paper. We conduct a horizontal comparison of nine representative and actively maintained state-of-the-art memory systems, followed by in-depth analysis and comprehensive evaluation. Compared with existing benchmark studies, our work achieves the most extensive and comprehensive evaluation of memory systems, which can serve as a reference for practitioners.

E.1 Memory-System Implementations

Mem0 (Chhikara et al., 2025) extracts and updates salient facts in a vector store, with optional entity–relation graphs for structured recall.

Cognee (cog, 2026) transforms source data into typed DataPoints across relational, graph, and vector stores and retrieves evidence through vector-seeded graph projection.

Hindsight (Latimer et al., 2025) extracts structured facts, experiences, and entities, and integrates them into networks such as observations. It employs multi-channel hybrid retrieval: sparse retrieval based on keywords, dense retrieval, temporal retrieval and graph traversal.

MemU (MemU) organizes source data, extracted memory items, and topic summaries into a three-level hierarchy for coarse-to-fine retrieval.

MemOS (Li et al., 2025) designs OS-like memory unit caches and lifecycle management mechanisms.

EverMemOS (Hu et al., 2026) For memory consolidation, EverMemOS follows the pipeline: chunking, aggregation and summarization of data → memcell construction → clustering → memscene. The retrieval is based on such abstraction memory rather than raw data.

MindMemOS (Liang et al., 2026) combines dense and sparse memory retrieval with a provenance-aware relationship graph and explicit memory-update operations.

Zep (Rasmussen et al., 2025) maintains a bi-temporal knowledge graph that preserves source episodes, fact validity, and update history for structured retrieval.

GraphRAG (Edge et al., 2024) builds an entity–relation graph with hierarchical community summaries to support local evidence retrieval and global corpus synthesis. For retrieval, it first computes similarity scores between the QA pair and stored summary memories (dense retrieval), retrieving the top-k memory nodes as the initial response candidates in the first round. Its drift mechanism enables the LLM to decompose the original query into subproblems. Starting from the initial nodes in the memory graph, the model continuously searches for other relevant nodes. Finally, it outputs all answers ranked by their scores.

We summarize the evaluated systems. Tables 20 and 22 separate the stored memory representation from the mechanisms used to retrieve it. Table 21 shows the memory units indeed used for answering. A check mark denotes a component in the declared pipeline; $+^*$ and $-^*$ denote optional capabilities enabled and disabled, respectively, in our evaluation; a dash denotes an unsupported or unused mechanism. All configurable dense text-embedding components use Qwen3-Embedding-4B in our evaluation. Qwen3-Reranker is used only when the evaluated retrieval route exposes a model-based reranking stage; routes without such a stage retain their native ranking procedure.

The two taxonomies above describe the intrinsic capabilities of each memory system. Notably, support for a representation in the capability table only means the system can store that form internally; it does not guarantee this representation will be returned during retrieval. Some systems persist raw source text internally for provenance, yet their retrieval pipelines do not expose raw source chunks to the answer-generation prompt under our evaluation configuration.

E.2 Validation of the LLM Judge

Following the setup in Section 3, every answer produced by the nine memory systems in Table 4 is judged by DeepSeek-V4-Flash, with each query scored three times and the majority vote taken as the final label. Because these labeled judges underlie all reported accuracies, we audit the judge against human annotation.

In practice, we draw a random sample of 150 questions from the full 3,380 questions using a fixed seed, including all question types. For each sampled question, we take the corresponding **EverMemOS** answer generated under the DeepSeek-V4-Flash setting and ask human annotators to relabel it by comparing it with the golden answer. This yields 150 EverMemOS answers, 92 that the judge had marked correct and 58 it had marked incorrect. The audit is therefore a meta-evaluation of the deployed judge on real main-experiment outputs, not a separate QA run. Since the same DeepSeek-V4-Flash model also serves as the default answer model, the audit additionally characterizes a same-model evaluator; the net-bias analysis below quantifies its directional error.

Table 23 shows the judge–human agreement, and Table 24 breaks it down by question category.

The judge agrees with the human reference on 140 of 150 questions (93.3%). When the judge marks an answer correct, it is right in 92.4% of the cases (85/92), and it misses 3 of the 88 answers that humans considered correct (recall 96.6%). Agreement is category-dependent (Table 24): perfect on Unanswerable and Conflict questions, above 87% on Single-hop, Knowledge update, Non-declarative (ND), Causal, Hidden information, and Multi-hop questions, and lowest on Temporal questions (77.3%).

E.2.1 Systematic Mismatches

The ten disagreements are systematic in both directions: seven false positives, in which the judge accepts an answer the human reference rejects, and three false negatives, in which it rejects a reasonable answer. The false positives follow three patterns: temporal-ordering questions whose sequence

Table 20: Memory-representation taxonomy of the evaluated systems. *Raw/full text* denotes retained source messages, documents, or resources; *derived text* denotes extracted facts, summaries, episodes, or other textual memory units. Object/record, relational table, hierarchy, and graph identify distinct structured representations rather than serialization formats. $+^*$ and $-^*$ mark optional components enabled and disabled in our evaluation.

System	Raw/full text	Derived text	Dense vectors	Object/ record	Relational table	Hierarchy	Graph	Other representation
Mem0	–	✓	✓	✓	–	–	$-^*$	Metadata-scoped atomic facts
Hindsight	✓	✓	✓	✓	–	–	✓	Typed temporal and causal networks
Cognee	✓	✓	✓	✓	✓	–	✓	Typed DataPoints
Zep	✓	✓	✓	✓	–	✓	✓	Bi-temporal validity state
EverMemOS	✓	✓	✓	✓	–	✓	–	Profiles, Foresight, and event logs
MindMemOS	✓	✓	✓	✓	–	–	✓	Version and provenance lineage
MemU	✓	✓	✓	✓	–	✓	–	Multimodal provenance
MemOS	✓	✓	$+^*$	✓	–	–	$-^*$	Activation and parametric memory (optional)
GraphRAG	✓	✓	✓	✓	✓	✓	✓	Community reports

Table 21: Representations of the final retrieved memory units used for answering. Columns follow Table 20, excluding Dense vectors and Other representation. A check mark denotes returned evidence; a dash denotes a representation not supplied as answer context. Hierarchy marks high-level episode or community summaries; Graph marks returned node/edge content, including textual linearizations. These marks do not imply that a complete hierarchy or graph is passed to the answer model. Categories are non-exclusive.

System	Raw/full text	Derived text	Object/ record	Relational table	Hierarchy	Graph
Mem0	–	✓	✓	–	–	–
Hindsight	✓	✓	✓	–	–	✓
Cognee	–	✓	–	–	–	✓
Zep	✓	✓	✓	–	✓	✓
EverMemOS	–	✓	–	–	✓	–
MindMemOS	–	✓	✓	–	–	✓
MemU	✓	–	–	–	–	–
MemOS	–	✓	✓	–	–	–
GraphRAG	–	✓	–	–	✓	–

is correct but whose dates are not; missed direct contradictions, mostly numeric mismatches or opposite conclusions.

The three false negatives share a single pattern: open-ended Non-declarative (ND) or Causal questions whose answers are reasonable and cover the golden content without restating its wording.

The year-end question asks the system to arrange Feng Haoran's personal-growth and career moments of 2025 in chronological order. One example of the temporal-ordering pattern, in raw form:

Raw data: answers and verdicts

Generated answer (judge: correct; human reference: correct order but incorrect date). "In chronological order: 1. **April 24, 2025**: in the morning, [he] confirmed that the paper 'Practice and Reflections on Fintech Regulation' was officially online on the Hunan Finance website and shared it with family and friends and his supervisor; 2. **October 29, 2025**: at his workstation, [he] used the Wind terminal and company databases to collect financial data and research reports for the first column; ... 6. **December 20, 2025** (after the competition): [he] received the year's 'Active Newcomer' award at the gym's year-end review."

Golden answer. "In chronological order: 1. the paper confirmation (September 5, 2025, morning, at home); 2. the first-column material collection (October 29, 2025, morning, at the office); ... 6. the 'Active Newcomer' award (December 20, 2025, after the competition)."

Table 22: Search-method taxonomy of the evaluated systems. *Dense* denotes embedding-based nearest-neighbor retrieval, *sparse* denotes lexical or BM25 retrieval, *fusion* includes mechanisms such as RRF or MMR, and *agentic* denotes an LLM-controlled iterative retrieval process. The final column records system-specific search mechanisms outside these predefined categories. $+^*$ and $-^*$ mark optional methods enabled and disabled in our evaluation.

System	Dense	Sparse	Metadata filter	Graph traversal	Temporal search	Hierarchical search	Fusion	Reranking	Agentic	Other search mechanism
Mem0	✓	–	✓	$-^*$	–	–	–	$+^*$	–	–
Hindsight	✓	✓	✓	✓	✓	–	✓	✓	–	Proof-count and recency scoring
Cognee	✓	–	–	✓	–	–	–	–	–	Importance-weighted triplet scoring
Zep	✓	✓	✓	✓	✓	–	✓	✓	–	Community search
EverMemOS	✓	✓	–	–	–	✓	✓	$+^*$	$+^*$	Scene-guided retrieval
MindMemOS	✓	✓	✓	✓	–	–	✓	$+^*$	$+^*$	Complementary-query generation
MemU	✓	–	–	–	–	✓	–	–	$-^*$	Category–Item retrieval with resource provenance
MemOS	$+^*$	–	✓	$-^*$	–	–	–	$+^*$	–	Intent-aware route scheduling
GraphRAG	✓	–	–	✓	–	✓	–	–	✓	Global map–reduce and DRIFT

Table 23: Judge–human agreement on the 150-question audit sample.

	Judge: correct	Judge: incorrect
Human: correct	85	3
Human: incorrect	7	55

Table 24: Per-category judge–human agreement on the audit sample. Categories follow multi-label counting.

Question category	n	Agreement (%)
Single-hop Retrieval (SH)	66	95.5
Multi-hop Reasoning (MH)	55	87.3
Non-declarative Memory (ND)	28	89.3
Unanswerable (UA)	27	100.0
Temporal Reasoning (TR)	22	77.3
Knowledge Update (KU)	21	95.2
Conflict Detection (CD)	19	100.0
Causal Reasoning (CR)	16	87.5
Hidden-information Mining (HI)	9	88.9

E.2.2 Net Bias and Usage

The two bias directions partially cancel: the judge marks 61.3% of the sample correct versus 58.7% by the human reference, an overestimate of about 2.7 points on this audit sample. Our conclusions rely primarily on relative comparisons among systems scored under the same procedure, which reduces but does not eliminate the effect of systematic judging errors. The audit supports using DeepSeek-V4-Flash as the judge for LifeBench, with two caveats: absolute accuracies may be modestly overestimated, and temporal questions carry the largest judge uncertainty (77.3% agreement). The observed 2.7-point difference is descriptive of this sample and is not applied as a universal correction to all reported scores.

E.3 Hindsight and EverMemOS Results across Model Parameter Scales

Considering the privacy and resource constraints of local deployment, we evaluate Hindsight and EverMemOS across three Qwen3 parameter scales. Figure 11 reports overall Micro accuracy. Even at 8B, Hindsight and EverMemOS retain accuracies of 45.92% and 47.40%, respectively. Hindsight improves substantially at 14B (58.43%) and then plateaus at 32B (58.11%), indicating diminishing returns beyond 14B. EverMemOS is less sensitive to scale, ranging only from 47.16% to 49.88%.

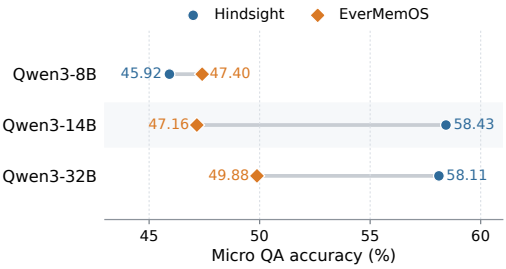

Figure 11: Micro QA (%) across LLM scales.

Table 25 further reports category-level performance. Simple tasks such as Single-hop, small LLMs remain effective, yet complex reasoning benefits more from larger models.

Table 25: Hindsight and EverMemOS accuracy (%) by LifeBench question category across Qwen3 parameter scales (thinking disabled). Category abbreviations follow Figure 3.

	Hindsight			EverMemOS		
Question category	Qwen3-8B	Qwen3-14B	Qwen3-32B	Qwen3-8B	Qwen3-14B	Qwen3-32B
SH	62.45	77.16	77.16	56.30	55.99	61.89
MH	31.01	35.80	35.80	27.12	24.23	30.56
UA	28.38	52.48	49.57	62.39	68.72	54.02
TR	23.77	27.74	26.79	17.36	17.36	21.89
ND	37.83	37.64	40.11	31.56	27.00	39.35
CD	62.97	78.07	80.19	62.97	64.39	69.34
KU	66.32	73.52	71.72	64.01	59.13	66.58
CR	37.00	41.82	43.43	35.39	31.64	39.14
HI	18.56	21.65	21.13	10.82	12.37	15.98
Overall	45.92	58.43	58.11	47.40	47.16	49.88

E.4 Detailed Cost Measurements

To quantify the lifecycle cost of memory systems, we measure LLM token consumption, ingestion, retrieval and answer latency. All measurements use year-long data of one sampled user (Sun Yuwei) under a strictly serialization protocol. This ensures a fair comparison but does not reflect the actual latency achievable with system-level optimizations during deployment.

The results in Figure 12 show that *intensive LLM use does not reliably translate into higher accuracy.* Hindsight provides the strongest overall accuracy–cost trade-off, whereas several more resource-intensive systems achieve no comparable improvement in accuracy. Hindsight's bottleneck is the reranking stage, which processes a large candidate set aggregated from multiple retrieval methods, as shown in Section 3.3. GraphRAG exhibits a distinct deferred-computation profile, incurring low ingestion costs but the highest retrieval latency and search-token consumption at query time.

To complement the aggregate cost–accuracy analysis in the main text, Table 26 reports the complete stage-wise measurements for the seven systems with full execution traces. The breakdown separates token use and latency across ingestion, retrieval, and answer generation, while also reporting storage growth and peak-memory overhead.

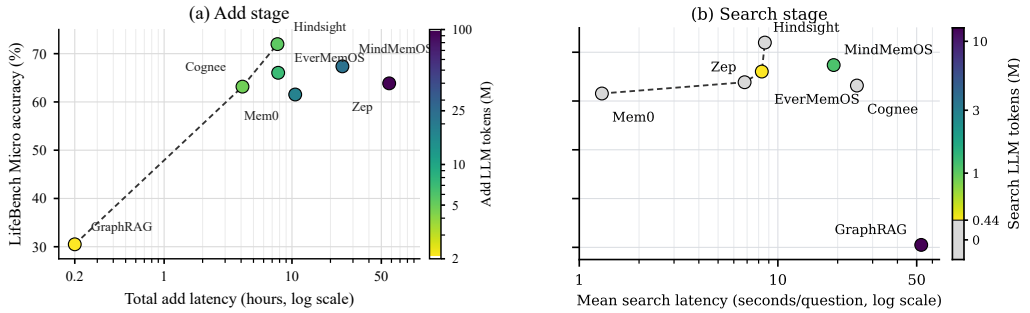

Figure 12: Stage-specific cost–accuracy trade-offs across memory systems. Panel (a) shows total add latency over the one-year history, and panel (b) shows mean search latency per question. Dashed lines mark the Pareto frontiers.

Table 26: Cost of processing one user's complete LifeBench workload across seven systems. Add and Search report total LLM tokens; Answer reports input and output tokens separately. Add latency is the total ingestion time over the year-long history; Search and Answer latencies are means per question. Storage increase and peak memory cover the complete run.

Memory system	LLM tokens (M)				Latency			Storage (GB)	Peak memory (GB)
	Add	Search	Ans. in	Ans. out	Add (h)	Search (s)	Answer (s)		
Mem0	17.69	0.00	0.37	0.35	10.6	1.3	32.7	0.10	0.11
Hindsight	5.41	0.00	2.13	0.14	7.7	8.6	14.9	0.15	0.82
Cognee	4.83	0.00	1.72	0.02	4.1	25.0	4.9	0.10	0.88
EverMemOS	7.04	0.44	0.88	0.35	7.8	8.3	38.6	0.84	0.72
MindMemOS	22.84	1.14	1.20	0.15	24.8	19.1	18.9	0.15	1.00
Zep	96.06	0.00	0.74	0.19	57.6	6.8	21.5	0.91	3.58
GraphRAG	2.10	12.65	0.09	0.04	0.2	52.7	8.3	0.52	1.12

E.5 Retrieval Capability Statistics

Table 27 reports the retrieval statistics discussed in Section 3.3, computed on the final post-reranking results. Recall is the percentage of gold evidence records covered by at least one returned unit; Results per query is the average number of units returned; Tokens per unit measures the average number of tokens within each memory unit. TP denotes the fraction of tokens belonging to question-useful sentences extracted from the retrieved units by the LLM.

Table 27: Retrieval recall and result statistics after reranking. The final column summarizes candidate construction before reranking or final ordering.

System	Recall (%)	Results per query	Tokens per unit	TP (%)	Pre-reranking construction
Mem0	53.51	40.00	76.85	4.58	320 initial → RRF → 40 → rerank
Cognee*	50.69	89.77	291.56	1.19	500 initial → triplet-importance score (no rerank)
Hindsight	66.98	37.67	103.68	3.00	900 initial → RRF → 300 → rerank
MemOS	54.17	39.52	105.80	2.51	480 initial → Combine → 80 → rerank
EverMemOS	57.39	19.64	815.50	0.89	R1: 60 → RRF → 20 → sufficiency check; if insufficient, R2: 60 → RRF → 20; merge (40) → rerank
MindMemOS	62.47	15.28	1425.40	1.65	R1: 60 → RRF → 10 → sufficiency check; if insufficient, R2: 60 → RRF → 10; merge (20) → rerank
Zep	63.04	40.00	287.38	2.47	560 initial → RRF → 160 → rerank

E.6 Failure Attribution beyond Retrieval Sufficiency

E.6.1 Experimental Design

Retrieval traces contain cases in which the context is judged sufficient, yet the answer remains incorrect. In contrast, providing raw evidence allows the same answer model to answer nearly all questions correctly. This contrast raises a central question: at which stage does otherwise sufficient evidence become unusable for answer generation? We study this question under three experimental conditions, defined in the main text in the section 3.4:

$$\text{Gold-Raw} \rightarrow \text{Gold-Unit} \rightarrow \text{Normal Retrieval}.$$

Hindsight receives source-linked *MemoryFacts*, metadata, and context, whereas EverMemOS receives generated Episodes. Comparing *Gold-Raw* with *Gold-Unit* reveals the extent to which answer-bearing information is lost when raw records are transformed into system-specific memories and passed through their native answer interfaces.

Comparing *Gold-Unit* with *Normal Retrieval* measures the additional degradation associated with normal retrieval and downstream processing, including failures to recover required memories and to preserve relevant information in the final answer context.

All conditions use DeepSeek-V4-Flash and the same three-vote evaluation protocol. The original workload contains 328 questions. After excluding two questions whose evidence cannot be mapped to EverMemOS memory units, 326 questions remain, including 267 answerable questions. We report the complete matched comparison on a shared 183-question subset. Because the *Gold-Raw–Gold-Unit* transition does not depend on retrieval, we additionally evaluate this transition on all 267 answerable questions.

E.6.2 Results and Attribution

Results are shown in the section 3.4. Both systems achieve 96.72 accuracy when provided with raw evidence, but their performance diverges after memory construction. They also experience substantial additional degradation under normal retrieval. The results indicate that retrieval is the main source of accuracy loss, while memory construction and source fidelity also matter.

Memory construction. On the 183-question subset, replacing *Gold-Raw* with *Gold-Unit* reduces accuracy. The same trend holds over all 267 answerable questions, with larger degradation for EverMemOS than for Hindsight.

This difference reflects how the two systems organize and preserve source information. Hindsight links extracted *MemoryFacts* to source chunks, retaining access to more original information. EverMemOS, in contrast, passes compressed Episodes to the answer model, making answer quality more dependent on the information retained during memory construction. The difference between the two systems further motivates the source-fidelity analysis in Section F.1.

Normal retrieval and downstream processing. Moving from *Gold-Unit* to *Normal Retrieval* reduces Hindsight accuracy by 26.78 points and EverMemOS accuracy by 25.14 points. For both systems, this degradation exceeds the loss observed during memory construction.

Under normal retrieval, the systems must identify question-relevant memories and preserve their information through downstream processing. Retrieval failures can prevent required evidence from reaching the answer model, while subsequent selection and context construction may further reduce its accessibility. This comparison measures the overall degradation associated with normal retrieval and downstream processing, without isolating their individual contributions.

E.6.3 Implications

The experiment shows that retrieval is the main source of accuracy loss for both systems, while raw-data linking and source fidelity also matter. Hindsight preserves more raw information by linking *MemoryFacts* to source chunks, whereas EverMemOS incurs greater information loss when raw

records are compressed into Episodes. Both systems experience further substantial degradation under normal retrieval.

More generally, retrieval sufficiency does not guarantee continuity from evidence to answer. Answer-bearing information must survive memory construction, search must expose the required memories, and those memories must remain available and influential throughout downstream processing and answer generation. We refer to this broader property as *retrieval-to-answer continuity*.

E.7 Analysis of Hindsight's Robustness under Batch and Dynamic Indexing

E.7.1 Experimental Scope

Hindsight retrieves evidence through four channels. We examine whether its batch and dynamic indexing robustness is associated with either the individual retrieval channels or the size of this pre-reranking candidate pool.

The analysis uses the same 326 questions from one fully traced LifeBench user, after excluding two questions whose evidence cannot be mapped consistently. These single-user results are used for controlled mechanism analysis and are separate from the aggregate results in Table 29.

E.7.2 Retrieval-Channel Ablation

We disable one retrieval channel at a time under batch and dynamic indexing evaluation while keeping RRF fusion, Cross-Encoder reranking, context construction, and answer generation unchanged.

Table 28: Accuracy change after removing each Hindsight retrieval channel.

Removed channel	Batch indexing change	Dynamic indexing change
Semantic	-20.25 pp	-21.47 pp
BM25	-1.84 pp	-2.15 pp
Graph	-2.45 pp	-1.23 pp
Temporal	-1.53 pp	$+1.23$ pp

Semantic retrieval is essential to absolute accuracy, but its removal causes similar losses in both settings and does not explain the batch and dynamic indexing stability. The other channels have substantially smaller effects. Temporal retrieval shows the largest directional difference: its removal reduces batch indexing accuracy but increases dynamic indexing accuracy. Nevertheless, none of the four protocol-dependent effects is statistically significant. The channel ablation therefore does not identify any single retrieval method as the source of Hindsight's robustness.

E.7.3 Candidate-Pool Ablation

We next examine whether Hindsight's broad pre-reranking candidate pool contributes to its robustness across batch and dynamic indexing memory states. Hindsight normally sends up to 300 RRF-fused candidates to a Cross-Encoder, although only 40 memories are eventually returned. For each of the same 326 questions, we reuse the RRF list saved by the corresponding baseline and construct a replay condition that retains only its first 40 candidates before Cross-Encoder reranking. The retrieval channels and RRF ordering are therefore held fixed, while reranking, filtering, context construction, and answer generation follow the original pipeline.

As shown in Figure 13, restricting the candidate pool substantially reduces accuracy in both settings. Batch indexing accuracy decreases by 12.27 percentage points, whereas dynamic indexing accuracy decreases by 19.64 points; both reductions are significant under paired McNemar tests ($p < 0.0001$). The dynamic indexing condition is therefore more sensitive to candidate-pool reduction. The dynamic-minus-batch accuracy difference changes from $+1.84$ points with the original pool to -5.53 points with replay-40, corresponding to a 7.37-point difference in sensitivity (95% CI $[0.61, 14.11]$).

The saved traces explain why the larger pool matters. In the original runs, 8,850 batch indexing and 8,233 dynamic indexing final-context fact occurrences originated below RRF rank 40 and were

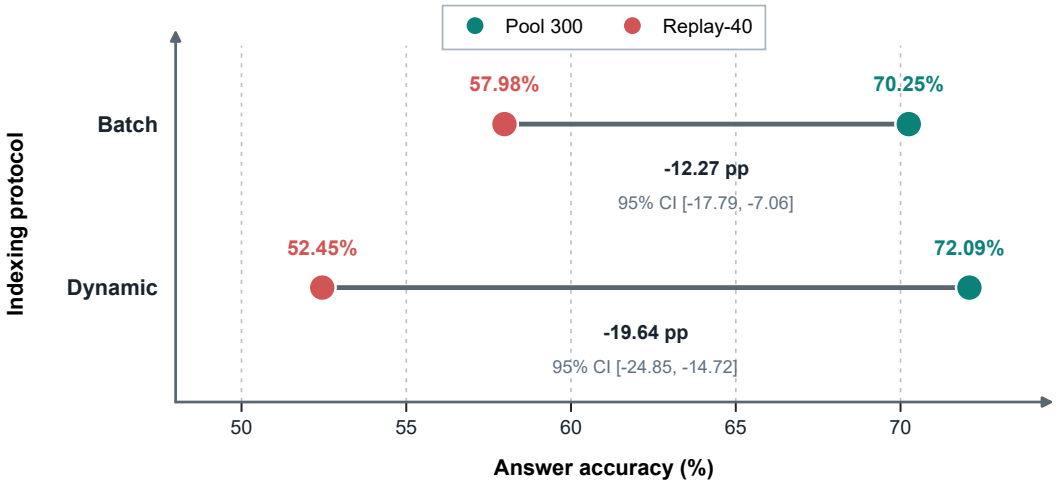

Figure 13: Effect of reducing Hindsight's pre-reranking candidate pool from 300 to 40 under batch and dynamic indexing evaluation. Points show answer accuracy, and each connecting segment shows the paired accuracy reduction caused by replaying only the first 40 RRF candidates. Confidence intervals are obtained by paired bootstrap over the same 326 questions.

subsequently promoted by the Cross-Encoder. Replay-40 removes these candidates before they can be reassessed and also removes their associated chunk and entity context. This effect is not explained by repeated Cross-Encoder scoring: the median Spearman correlation between the original and replayed rankings is 0.9985 in both settings.

Together with the retrieval-channel ablation, this result distinguishes channel identity from candidate coverage. No retrieval channel accounts for Hindsight's similar batch and dynamic indexing accuracy, whereas restricting the shared candidate pool harms both settings and disproportionately harms dynamic indexing. Broad candidate coverage therefore appears to be the strongest observed contributor to Hindsight's batch and dynamic indexing robustness: it allows the Cross-Encoder to recover useful evidence that individual retrieval channels place below the initial top 40.

E.7.4 Interpretation

The traces confirm that the larger candidate pool is actively used. Across the 326 questions, batch and dynamic indexing final-context fact occurrences were initially located below RRF rank 40 and were subsequently promoted by the Cross-Encoder. Restricting the pool removes these candidates before they can be reassessed and also changes their associated chunk and entity context.

The two ablations produce different conclusions. The channel ablation shows no individual retrieval channel explains Hindsight's batch and dynamic indexing robustness. In contrast, the candidate-pool ablation shows that broad candidate coverage is important in both settings and particularly important under dynamic indexing. The final limit of 40 controls how many memories reach answer generation, whereas the 300-candidate pool determines how deeply the Cross-Encoder can search the fused ranking. This ability to recover evidence from below the initial top 40 is the strongest observed contributor to Hindsight's robustness across batch and dynamic indexing memory states.

F Additional Experiments

F.1 Source Fidelity at the Answer Boundary

Replacing raw evidence with Gold-only memory reduces accuracy with Hindsight and EverMemOS. This persistent difference motivates an architectural examination of how each system carries source evidence to the answer model.

F.1.1 ARCHITECTURAL DIFFERENCE

Figure 14 shows the central distinction. Hindsight stores source documents as chunks and extracts *MemoryFacts* from them. Each Fact retains a *chunk_id*. After retrieval and reranking, the system constructs the answer context from selected Facts and their associated source chunks. The answer model can therefore recover details that are not fully reproduced in the extracted Fact, provided that the relevant Fact and chunk survive filtering.

EverMemOS aggregates messages into *MemCells* and generates compressed Episodes. Retrieval and answering operate primarily on these Episodes. Its default answer interface supplies numbered Episode text but does not restore the original messages. A detail omitted from the Episode therefore remains unavailable even when the correct Episode is retrieved.

Both systems use semantic abstraction, but Hindsight retains an active path back to the source evidence. EverMemOS depends more directly on the completeness of its compressed Episode.

F.1.2 ILLUSTRATIVE EVIDENCE

One source-oracle case illustrates this representation difference. The question asks for the full name of a client referred to as Mrs. Chen and whether a photograph identifies her. The original photograph record contains:

> *Caption:* Yu Xiaowei and client Mrs. Chen pose in the living room, holding an iPad and an insurance contract.
> *Face recognition:* Yu Xiaowei, Chen Meiling.

Hindsight preserves the identity binding between Mrs. Chen and Chen Meiling in its answer-visible *MemoryFacts*, together with a link to the source chunk. When these source-linked Facts are supplied directly, the answer model correctly identifies the client as Chen Meiling and cites the photograph.

EverMemOS generates Episodes that preserve the client visit, policy signing, and photograph, but omit the face-recognition value. Even when the relevant Episodes are supplied directly, the answer model can confirm that a photograph exists but cannot recover Mrs. Chen's full name.

Ordinary retrieval is bypassed in both conditions. The failure therefore occurs during memory construction: Hindsight retains an answer-visible path to the source metadata, whereas the compressed EverMemOS Episodes no longer contain the required identity binding.

F.1.3 IMPLICATION

The result distinguishes semantic abstraction from source fidelity. Compact memory units can support efficient retrieval, but omitted details become unrecoverable unless the system retains an active path back to the source.

Hindsight provides such a path through the connection between *MemoryFacts* and source chunks. EverMemOS's default answer path depends on the information retained in its Episodes. This architectural difference is consistent with Hindsight's smaller Raw-source–Gold-only loss.

Source addressability does not guarantee complete preservation. Hindsight may still miss a relevant Fact, filter its source chunk, or truncate recovered text under the context budget. It nevertheless provides a recovery path that is absent from EverMemOS's default Episode-centered answer interface.

F.2 DATA FLOW: DYNAMIC INTERACTION AND BATCH INGESTION

In this section, we investigate two data-flow paradigms for memory systems: dynamic data ingestion and batch indexing, where all phone records are preloaded before sequential query answering. We distinguish a question's semantic query time from its execution time. In the batch-indexing setting, we change only the query-execution schedule: all phone records are first ingested in their original chronological order, after which the questions are executed in their original order. Each question retains its original content and all stored fields, including its semantic query timestamp; the evidence cutoff used to construct its gold answer and the gold answer itself also remain unchanged. Temporal expressions such as "currently" or "at that time" therefore continue to refer to the original query

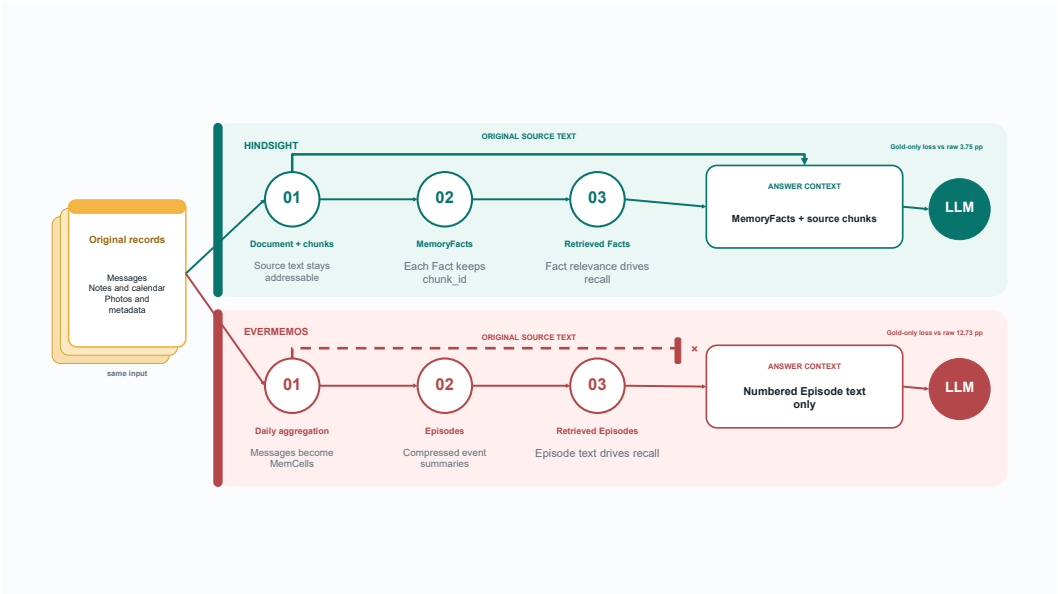

Figure 14: Source-evidence paths in Hindsight and EverMemOS. Hindsight connects extracted *MemoryFacts* to their source chunks and can restore those chunks during answer-context construction. EverMemOS supplies retrieved Episode text through its default answer path without restoring the original messages. The reported losses are measured on all 267 answerable questions.

timestamp, rather than the later time at which the question is executed. Consequently, the batch index may contain records dated later than a question's semantic query time. This is intentional and reflects the full-history indexing protocol; the system is still evaluated against the answer valid at the original query timestamp. Memory systems run with their original implementations and unchanged configurations, and both paradigms ingest identical phone records in the same order. Their only difference is how memory writes and query executions are interleaved. Thus, any accuracy difference can be attributed to the data-flow paradigm under this controlled setup. Table 29 reports the complete category-level comparison, including both Micro and Macro averages.

F.2.1 Case Study

Case: EverMemOS. The question asks: "I previously went to see that apartment in Lainyin Town—how much was the total price, roughly?" (reference answer: the total price is 1.18 million yuan.)

Raw data: retrieval and answers

dynamic indexing. Retrieved (rank 3 of 20, the March 16 record): "On March 16, 2025, Ye Mingxuan made progress on both the apartment purchase and the 'coin-chain' project. On the purchase side, he inspected Room 502 in Building 17 of Lainyin Town in the morning, decided to buy it, and discussed with the agent about negotiating the deposit in the afternoon. ... In the evening, he created a dedicated finance sheet for the purchase, entering the budget, the down-payment gap, and the loan plan ... Ye Mingxuan completed the scanning and restoration of the family's old photos, which he plans to give to his father as a birthday gift."
Answer (incorrect): "The memory does not directly record the final total price of the Lainyin Town Room 502, Building 17 apartment you visited, but back-calculating from the figures at the time ('total down payment 800,000 yuan, monthly payment 4,200 yuan, provident-fund loan 430,000 yuan, commercial loan 3.8%'), the total price is about 1.74 million yuan; more conservatively, about 1.70–1.75 million yuan."

batch indexing. Retrieved (rank 1 of 20, the March 16 record): "On the morning of March 16, 2025, Ye Mingxuan began sorting the family's old photos ... Meanwhile, Ye Mingxuan scheduled another item: at 10:10 he would go to Room 502, Unit 3, Building 17 of Lainyin Town to inspect in person a two-bedroom listing with a total price of 1.18 million yuan, led by an agent ... "
Answer (correct): "The apartment in Lainyin Town you previously visited (Room 502, Unit 3, Building 17 of Lainyin Town) has a total price of 1.18 million yuan. It is a two-bedroom listing that you inspected in person with an agent at 10:10 on the morning of March 16, 2025, and you decided to buy it at noon that day, planning to use a combined provident-fund loan with a monthly payment of about 4,200 yuan."

Differences in Memory Unit Granularity Caused by Update Frequency. In this case, the dynamic indexing memory unit combines house-purchase activities with unrelated project and family events, without explicitly retaining the apartment price. In contrast, the batch indexing memory unit preserves the price and presents a more focused account of the apartment viewing. Under dynamic indexing, daily ingestion restricts chunking to shorter time windows, potentially grouping unrelated activities into chronologically organized memory units. Batch indexing allows the system to process the complete data stream and aggregate topically related information across days. These differences in memory content and granularity can affect retrieval ranking and the availability of answer-bearing information.

F.2.2 Result Analysis

	GraphRAG		Hindsight		EverMemOS	
Category	Batch	Dynamic	Batch	Dynamic	Batch	Dynamic
SH	42.56	22.10	82.50	83.43	75.36	71.76
MH	22.86	12.93	55.06	53.98	51.45	48.55
CR	29.48	15.01	63.81	67.83	60.73	59.25
TR	19.15	12.83	47.74	44.34	41.74	38.30
KU	46.07	16.97	80.72	85.09	74.01	81.23
CD	47.00	11.08	81.13	80.66	73.91	67.45
HI	16.20	10.31	51.03	49.48	34.92	34.02
ND	27.86	19.77	57.22	58.75	54.06	54.75
UA	83.27	88.38	76.75	75.56	87.46	85.30
Micro	**43.21**	**30.50**	**72.10**	**71.98**	**69.08**	**66.04**
Macro	**37.16**	**23.26**	**66.22**	**66.57**	**61.52**	**60.07**

Table 29: Category-level batch indexing vs. dynamic indexing QA accuracy (%) for GraphRAG, Hindsight, and EverMemOS.

	EverMemOS			Hindsight		
Category	Dense	Sparse	Δ	Dense	Sparse	Δ
SH	74.8	79.4	+4.7	83.2	82.7	−0.5
MH	54.4	66.0	+11.6	60.5	62.6	+2.1
TR	37.7	54.7	+17.0	58.5	62.3	+3.8
ND	64.4	66.1	+1.7	55.9	61.0	+5.1
CD	57.1	77.1	+20.0	72.9	78.6	+5.7
CR	69.2	72.3	+3.1	72.3	60.0	−12.3
KU	85.2	79.6	−5.6	83.3	79.6	−3.7
HI	48.0	60.0	+12.0	56.0	44.0	−12.0
UA	87.7	87.7	0.0	77.8	82.7	+4.9
Micro	**70.2**	**76.4**	**+6.2**	**74.0**	**75.7**	**+1.8**

Table 30: Per-category and Micro accuracy (%) for the density ablation, comparing dense with sparse contexts on 453 questions. Category rows use multi-label counting and abbreviations from Figure 3; Micro is the only aggregate score. Δ is sparse minus dense. The experiment uses three users.

As shown in Table 29, the Macro results show a similar degradation for GraphRAG and EverMemOS, whereas Hindsight remains nearly unchanged across the two indexing modes. Memory unit architectures such as the graph structure in GraphRAG and the hierarchical structure in EverMemOS involve information aggregation and linking. The batch indexing setting allows memory aggregation to access richer and more complete information, which typically yields more coherent aggregated memory units. To further verify our hypothesis, we categorize and count the system accuracy based on the original semantic query timestamp for single-hop questions (to ensure uniform question difficulty). As illustrated in Figure 15, GraphRAG shows a clear degradation of system capability over time under the dynamic indexing setting, and EverMemOS displays a mild version of this trend. In contrast, the batch indexing mode substantially mitigates such failure.

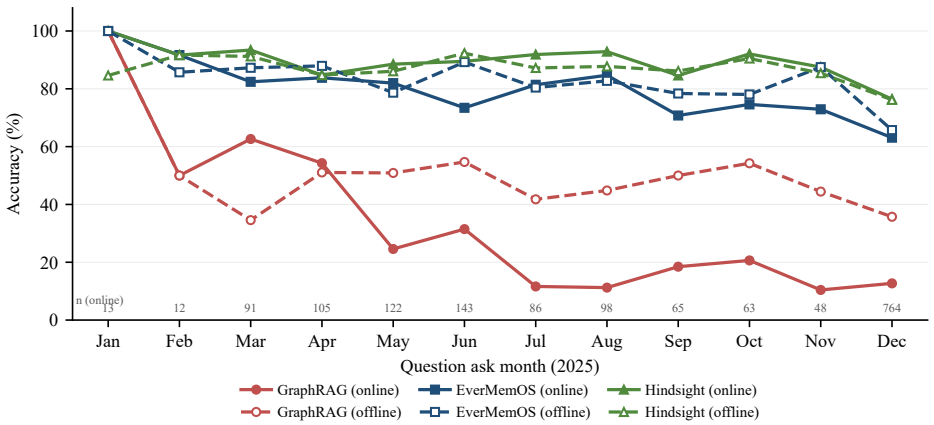

Figure 15: System accuracy (%) on Single-hop questions under the batch and dynamic indexing protocol, grouped by the month of each question's original semantic query timestamp.

F.3 Context Density: Dense vs Sparse Temporal Context

The construction details and full statistics for the dense and sparse datasets are provided in the section 3.6. This appendix presents the experimental results obtained from these controlled variants.

F.3.1 Case Study

Case: EverMemOS. The question asks "What weight did Sun Yuwei's scale show on the morning of April 15?" (reference answer: 46.0 kg);

Raw data: retrieval and answers

Dense. Retrieved (rank 1 of 20): "On the morning of April 16, Sun Yuwei weighed herself ... her weight reached 46.0 kg, achieving her gain goal ... In addition, on April 15 she recorded a morning weigh-in note and arranged her morning walk and yoga (6:45–7:45) ... "
Answer (incorrect): "Based on the current memory, the exact weight cannot be determined; the memory only mentions that she logged a morning weigh-in note on April 15 without recording the value."

Sparse. Retrieved (rank 1 of 20): "April 15: ... she woke at 6:15 and her weight reached 46.0 kg, achieving her gain goal ..." *(rank 2 of 20):* "... after waking at 6:15 she weighed herself barefoot; the scale read 46.0 kg, achieving her goal of gaining from 44 to 46 kg ... "
Answer (correct): "At 6:15 that morning she weighed herself barefoot; the scale read 46.0 kg, achieving her goal of gaining from 44 to 46 kg; at 6:20 she photographed the reading as proof."

Information Loss of Memory Units Caused by Dense Data. In the dense-context results, since data from two adjacent days (both with weigh-in records) are available, the memory unit merges and summarizes the episodic memory. It attributes the weigh-in entry to April 16 and only leaves a note for April 15, failing to mention the weigh-in record of April 15. This information loss originates from the LLM summarization. In contrast, under sparse context, the rank-1 and rank-2 retrieved units directly describe the weigh-in information on April 15 itself, yielding a correct final answer.

F.3.2 Result Analysis

As shown in Table 30, sparse contexts raise the Micro accuracy of both systems. EverMemOS achieves prominent improvements on Multi-hop, Temporal and Conflict categories, while Hindsight remains stable Micro accuracy, consistent with its density robustness reported in the main text. Hindsight degrades on Causal and Hidden Information queries under sparse traces, and both systems see slight accuracy drops for knowledge-update questions.