HYPEROS-MEM: OS-CENTRIC PERSONAL MEMORY VIA FAST–SLOW HIERARCHICAL EVENT GRAPH

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing personal memory systems and benchmarks predominantly focus on isolated interaction domains (e.g., dialogue or video), yielding memories that capture only partial slices of user activity trajectories. By contrast, users leave massive cross-app, cross-modal, and cross-temporal digital trajectories on smartphone OS, such as the scattered conversations, orders, locations, and screens generated throughout a business trip. While building an OS-centric personal memory is highly promising, it faces three critical challenges: (1) fragment integration, as heterogeneous observations lack a unified structure; (2) event decoupling, given that parallel event logs intertwine chronologically; and (3) immediate attribution, because event membership and hierarchical context often require later evidence to resolve. To systematically study this problem, we construct PMM-Bench, comprising 3.83 million system-level records from 20 users, encompassing eight daily scenarios and featuring five representative evaluation settings. We further propose HyperOS-Mem, which organizes atomic observations into an evidence-grounded, temporally evolving, event-centric hierarchical memory graph through a coordinated fast–slow process. Experiments on PMM-Bench demonstrate that existing memory systems remain limited in this OS-level setting, whereas HyperOS-Mem achieves SOTA performance, validating the event-centric memory paradigm.

1 INTRODUCTION

As AI agents evolve from single-turn tools into long-term personal assistants, accumulating user experience and maintaining a durable model of the user become foundational to personalized intelligence (Zhong et al., 2024; Kang et al., 2025). Recent memory systems organize cross-session memories, user profiles, and supporting experiential evidence (Xu et al., 2026b; Xuan et al., 2026; Choi et al., 2026) for long-horizon dialogue, multimodal conversation, long video, and agent trajectories (Bei et al., 2026; Yeo et al., 2026b; Zhao et al., 2026; Yang et al., 2026b). However, each of these systems and benchmarks observes a single interaction domain, so the resulting memory covers only a slice of the user's activity, leaving an **experience coverage gap** (Deng et al., 2026a; Mei et al., 2026; Chai et al., 2026).

Users continuously generate system-level digital traces across applications, modalities, and time within smartphone OS, offering a unique foundation for comprehensive personal memory (Chen et al., 2026a). Unlike the semantically dense, well-bounded text of conventional memory settings, these traces form high-frequency, low-semantic, asynchronous, and interleaved streams of heterogeneous evidence, and a complete experience such as a business trip emerges only by integrating complementary signals across sources and modalities (Yeo et al., 2026a). In one user's OS stream (Fig. 1), sensor readings and screen operations make up 51.6% and 22.0% of all records, while directly descriptive records such as messages, screenshots, and gallery items account for about 0.21%.

Long-term memory requires explicit modeling of event structures, temporal relationships, and continuous evolution (Yu et al., 2026; Luo et al., 2026; Hu et al., 2026a; Chen et al., 2026b; Hu et al., 2026b; Yang et al., 2026a). In continuous OS streams, these issues couple into three challenges. (1) Fragment Integration: screen interactions, images, audio, and other system-level traces are scattered across applications and modalities without a unified semantic structure. (2) Event Decoupling: multiple real-world events evolve concurrently while their observations interleave along the timeline. (3) Immediate Attribution: the semantic value, event membership, and hierarchical position of an incoming observation are often unclear on arrival and become clear only as evidence accumulates.

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.

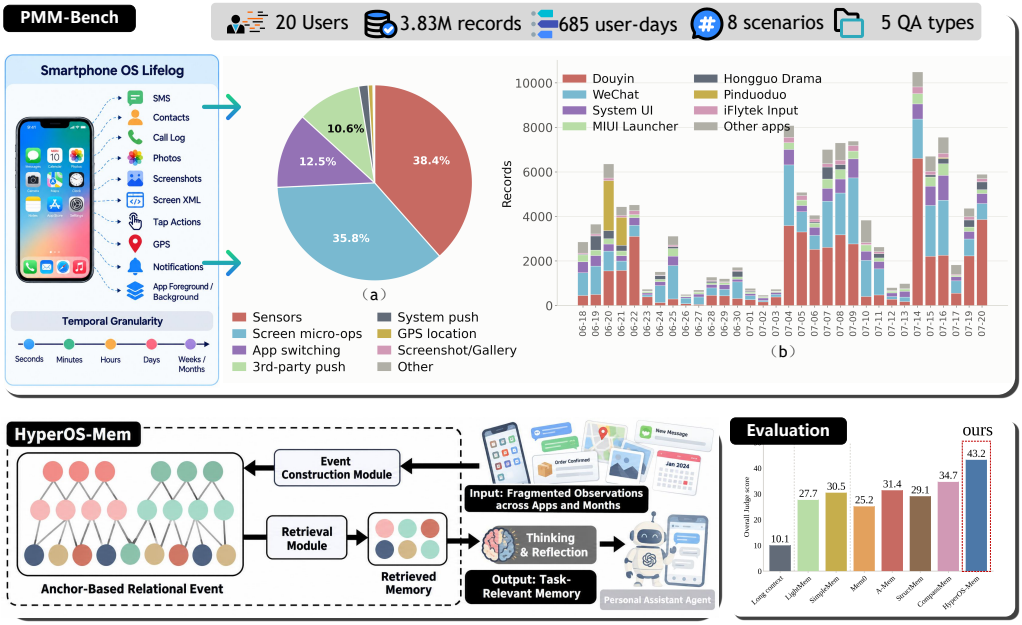


Figure 1: Top: Overview of PMM-Bench. (a) Data type composition for a representative user; (b) its distribution across time and applications. Bottom: The overall architecture of HyperOS-Mem and its performance evaluation on PMM-Bench.

In this paper, we introduce PMM-Bench and HyperOS-Mem for long-term personal memory from smartphone OS-level observation streams. PMM-Bench contains 3.83 million real system-level records from 20 Android users over eight daily scenarios, with about 2,000 questions spanning five tasks: single-hop recall, cross-application aggregation, temporal reasoning, event tracking, and habit induction. HyperOS-Mem builds on the temporal asymmetry of system observations: each atomic observation must be processed on arrival, while its event attribution, boundaries, and higher-level context often emerge only after later evidence accumulates. HyperOS-Mem therefore organizes observations into an evidence-traceable, evolving, event-centric hierarchical memory graph (Fig. 1, bottom). The fast path, Dual-Rate Evidence-Deferred attribution (DRED), attributes observations online in one pass without a generative LLM while maintaining parallel event threads; the slow path, SAGE, uses accumulated evidence to refine event structures, extract observation-grounded facts, and consolidate long-term memories across events. Our contributions are threefold:

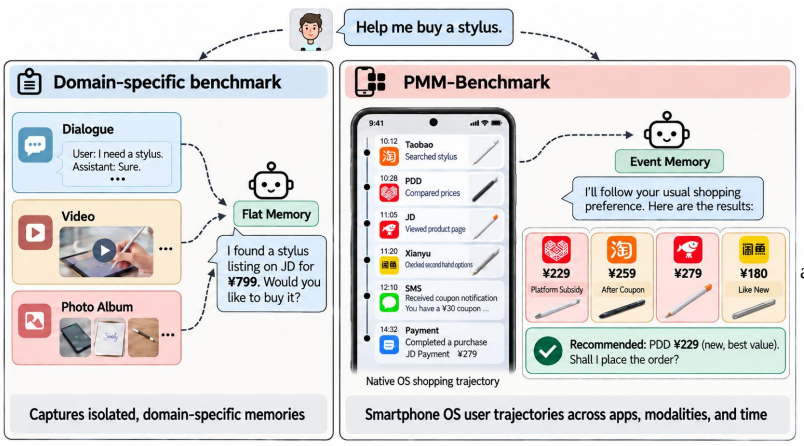
- **PMM-Bench.** We study OS-centric personal memory and build a benchmark for it from real long-term device records that span applications, modalities, and time.
- **HyperOS-Mem.** We propose a fast–slow event-centric hierarchical memory framework with LLM-free online attribution and a slow path that grounds facts and memories in source observations.
- **Empirical evaluation.** HyperOS-Mem outperforms baselines on PMM-Bench, validating event-centric memory, generalizes to two public benchmarks, and features DRED as a plug-in.

2 PMM-BENCH

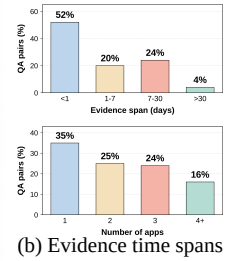
PMM-Bench is built in four stages: on-device data collection, activity context reconstruction, QA annotation, and quality review, with each answer linked to its supporting observations.

The source streams were recorded on the devices of 20 consenting Android users over 685 user-days. Each participant reviewed the logged traces and confirmed their export. The streams contain 3,829,719 records across eight everyday scenarios, covering screen operations, application switches, screenshots and gallery items, messages and notifications, locations, and sensor readings. As activities span applications and days and interleave, annotators remove empty records, consolidate redundant captures, and group related operations into events of twelve user-need categories.

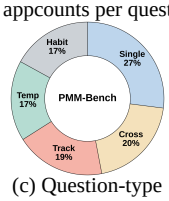
From these activity contexts, annotators write 90–120 questions per user from the user's first-person perspective. Each question carries a reference answer, an evidence set linking to supporting oper-



(a) Domain-specific vs. OS-centric benchmarks



(b) Evidence time spans and appcounts per question.



(c) Question-type distribution

Figure 2: Overview of PMM-Bench. (a) A stylus-shopping example compares domain-specific benchmarks with PMM-Bench, highlighting user activities that span applications, modalities, and time. (b) Distributions of evidence time spans and the number of applications supporting each question. (c) Proportions of the five question types in PMM-Bench.

Table 1: PMM-Bench and existing memory benchmarks. Modalities: T text, I image, G interface actions, L location, S sensor. **Natural** indicates whether the benchmark is primarily built from naturally occurring user traces rather than synthesized or task-scripted interactions.

Benchmark	Source	Natural	Scale	Modality
Dialogue				
LoCoMo (Maharana et al., 2024)	role-play dialogue	✗	50 conv / 7.5k Q	T
LongMemEval (Wu et al., 2025)	assistant dialogue	✗	500 Q / 1.5M tok	T
Mem-Gallery (Bei et al., 2026)	image–text dialogue	✗	multi-session	T, I
Digital footprint and file				
AMA-Bench (Zhao et al., 2026)	agent traces	partly	six domains	T
MobileMem (Deng et al., 2026a)	mobile app traces	partly	long-horizon	T
MemGUI-Bench (Liu et al., 2026a)	mobile GUI episodes	✗	128 tasks / 26 apps	I, G
FileGramBench (Liu et al., 2026d)	file-system behavior	partly	640 traces / 4.6k Q	T, I
Lifelog				
PMM-Bench	**Smartphone OS lifelog**	✓	**20 users / 3.83M rec**	T, I, G, L, S

ations, and a type label. Evidence often spans several applications and long periods (Figure 2b), and the type label (Figure 2c) identifies the primary ability needed to answer: single-hop recall locates one record; aggregation across applications joins records from different applications; temporal reasoning orders or dates events; event tracking follows an activity through its stages to the latest supported state; and habit induction identifies a regularity across several independent events.

To verify these question–answer pairs, reviewers check whether the linked observations support each answer and whether relevant evidence is missing. A RAG-based tool retrieves relevant events as review context (details in Appendix C). Table 1 compares PMM-Bench with existing memory benchmarks, many of which rely on generated dialogues or simulated activity traces. With real device records (Figure 2a), cross-application coverage, and question-level evidence, PMM-Bench tests whether memory systems connect observations across applications and time to answer questions about users' real activities, their progression, and recurring habits.

3 HyperOS-Mem

An OS-level observation stream is a time-ordered sequence of atomic observations $\mathcal{O} = \{o_1, o_2, \dots\}$ collected from screens, messages, notifications, the gallery, locations, and sensors. HyperOS-Mem builds the stream into a hierarchical event graph $\mathcal{G}$, from which retrieval returns the

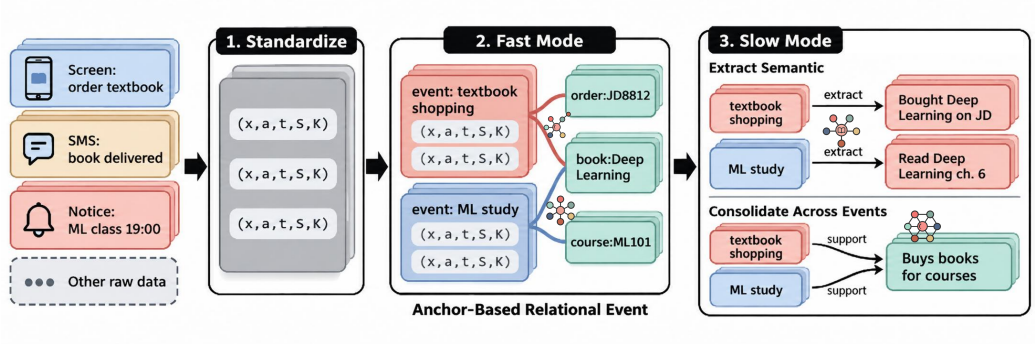


Figure 3: Overview of HyperOS-Mem. (1) Standardization converts device records such as screen operations, messages, and notifications into uniform atomic observations $(\mathbf{x}, a, t, \mathcal{S}, \mathcal{K})$. (2) The fast path DRED attributes observations online to parallel events and registers strong identifiers as entity anchors, and events sharing an anchor stand in relation. (3) The slow path SAGE applies SplitEvent, SummarizeEvent, and ExtractFacts to correct event structure and extract facts from events (top), and consolidates a long-term memory from the events that jointly support it (bottom).

Table 2: Layers of the hierarchical event graph.

Layer	Written by	Written from	Support
Event	DRED	observations attributed online	its observations M_E
Fact	SAGE, ExtractFacts	a single event	$\Pi_f \subseteq M_E$
Long-term memory	SAGE, consolidation	events of an anchor neighborhood	a set of events Ω_m

context that an answering agent uses to answer questions about personal history. The unit of the graph is an *anchor-based relational event* (ABRE): an event holds the observations attributed to it together with the anchors extracted from those observations. Table 2 gives the organization of the event, fact, and long-term memory layers: the structure of each layer matches the granularity at which its content is written and read, and every abstraction records the evidence that supports it.

Construction rests on an attribution map

$$\sigma : \mathcal{O} \to \mathcal{E} \cup \{\perp\}, \qquad M_E = \{o \in \mathcal{O} : \sigma(o) = E\}, \tag{1}$$

where $\mathcal{E}$ is the set of events and σ assigns an observation to an event or leaves it unassigned. Determining σ online involves a temporal asymmetry: an observation requires processing when it arrives, while its attribution, the boundary of the event holding it, and its position in the hierarchy may settle only after later evidence arrives. HyperOS-Mem therefore organizes construction into the three stages of Figure 3. Standardization converts heterogeneous device records into uniform atomic observations (Section 3.1). The fast path DRED maintains parallel events in a single online pass, writes σ, and registers entity anchors, without invoking a generative model (Section 3.2). The slow path SAGE revisits formed events in batches once evidence has accumulated, corrects their structure, extracts facts from events, and consolidates long-term memories across events (Section 3.3). Section 3.4 describes retrieval.

3.1 From Device Records to Atomic Observations

Each device record is converted into a uniform atomic observation $o_t = (\mathbf{x}_t, a_t, t, \mathcal{S}_t, \mathcal{K}_t)$, where $\mathbf{x}_t$ is a textual rendering with a modality tag, a_t is the host application, t is the occurrence time, and $\mathcal{S}_t$ and $\mathcal{K}_t$ are two sets of keys. A screen operation is transcribed by a vision-language model into a description of the page, its salient content, and the action performed; messages and notifications keep their text; gallery images are described by the same model; duplicate notifications and identical consecutive screens are merged into one observation (Appendix C.3). Keys are extracted by regular expressions and mapped by $\kappa(\cdot)$ to canonical typed keys, with merchant names normalized through a dictionary so that mentions of one referent in different sources map to the same key. Keys whose referent is unique within the stream of one user, such as tracking numbers and phone numbers, are the strong identifiers $\mathcal{S}_t$, and an inverted index maps each $s \in \mathcal{S}_t$ to the set $\mathrm{index}(s)$ of events that

contain it; the remaining keys, such as merchants, brands, and amounts, are the contextual keys $\mathcal{K}_t$, which are not indexed and serve as anchor candidates in SAGE.

The output of a text encoder is distributed anisotropically, so the similarity of any two texts carries a component independent of their content (Ethayarajh, 2019; Gao et al., 2019; Li et al., 2020). We represent an observation by centering within each processing period,

$$z_t = \mathrm{Norm}(f(\mathbf{x}_t) - \mu)\,, \qquad (2)$$

with $\mathrm{sim}(u, v) = u^\top v$, where f is a text encoder, $\mathrm{Norm}(v) = v/\|v\|_2$, and μ is the mean encoder output over the observations of the period; the similarities and thresholds below are defined in this space. Centering removes this component and keeps the surface differences between sources: observations of one activity across applications share little wording.

Algorithm 1: DRED

foreach $o_t \in \mathcal{O}$ *in temporal order* **do**
  $z_t \leftarrow \mathrm{Norm}(f(\mathbf{x}_t) - \mu)$; update the active set $\mathcal{A}_t$; determine E_t^* (Eq. 3)
  if E_t^* *is defined* **then**
    attach o_t to E_t^*; link the identifiers $\kappa(\mathcal{S}_t)$; update the prototype $c_{E_t^*}$ (Eq. 4)
  else if $\mathcal{P} \neq \emptyset$ ***and*** $\max_{p\in\mathcal{P}} \mathrm{sim}(z_t, z_p) \geq \tau_p$ **then**
    promote $\{p^*, o_t\}$ to a new event in $\mathcal{A}_t$
  else
    insert $(o_t, \hat{E}_t, t)$ into the pending set $\mathcal{P}$
  end
  foreach $(o_p, \hat{E}_p, t_p) \in \mathcal{P}$ *with* $t - t_p > \Delta_p$ **do**
    if $\hat{E}_p$ *is defined* **then** absorb o_p into $\hat{E}_p$ without updating its prototype
    else create an isolated event $\{o_p\}$
  end
end
merge events sharing a merchant key

3.2 Event Memory and Online Attribution

Besides the observation set M_E of Eq. 1, DRED maintains for each event a unit-vector prototype c_E and the interval $I_E = [t_E^{\mathrm{s}}, t_E^{\mathrm{e}}]$ spanned by its observations, and records for each observation the rule $\rho \in \{\textsc{attach}, \textsc{promote}, \textsc{absorb}\}$ that admitted it. An entity anchor refers to a definite referent and is registered by DRED at attribution; a topic anchor is written by SAGE.

DRED processes observations in temporal order and maintains the event set $\mathcal{E}$, the active set $\mathcal{A}_t = \{E \in \mathcal{E} : t - t_E^{\mathrm{e}} \leq \Delta_a\}$, and a pending set $\mathcal{P}$, using only index lookups and inner products. For o_t it either writes $\sigma(o_t)$ or, when the evidence is insufficient, defers that decision and records the main event $\hat{E}_t = \arg\max_{E\in\mathcal{A}_t} \mathrm{sim}(z_t, c_E)$, which is undefined when $\mathcal{A}_t = \emptyset$.

An order screen, a delivery notification, and a location trace share little wording, so their embedding similarity is unreliable, whereas a tracking number matches literally. Attribution is therefore decided first by identifiers. With $\mathcal{I}_t = \bigcup_{s\in\mathcal{S}_t} \mathrm{index}(s)$ the events that share a strong identifier with o_t, identifier-first attribution takes

$$E_t^* = \begin{cases} \arg\max_{E\in\mathcal{I}_t} \mathrm{sim}(z_t, c_E), & \mathcal{I}_t \neq \emptyset, \\ \arg\max_{E\in\mathcal{A}_t} \mathrm{sim}(z_t, c_E), & \mathcal{I}_t = \emptyset \text{ and } \max_{E\in\mathcal{A}_t} \mathrm{sim}(z_t, c_E) \geq \tau_a, \\ \perp, & \text{otherwise, deferring attribution to } \mathcal{P}. \end{cases} \qquad (3)$$

If E_t^* is defined, o_t is appended with $\rho_t = \textsc{attach}$, E_t^* re-enters $\mathcal{A}_t$ if needed, and $\kappa(\mathcal{S}_t)$ is registered as its entity anchors. Attribution ignores a_t, so an event may span applications, and $\mathcal{I}_t$ is not restricted to the active window, so a delivery message arriving days later still reaches the same event. After an observation is admitted, the prototype follows the dual-rate update

$$c_E \leftarrow \mathrm{Norm}((1 - \alpha_{a_t})c_E + \alpha_{a_t} z_t)\,, \qquad \alpha_a = \begin{cases} \alpha_{\mathrm{chat}}, & a \in \mathcal{H}, \\ \alpha_{\mathrm{def}}, & \text{otherwise}, \end{cases} \qquad (4)$$

where $\mathcal{H}$ is the set of messaging applications and $\alpha_{\mathrm{chat}} < \alpha_{\mathrm{def}}$. Consecutive chat messages vary in wording more than in topic, so the smaller rate keeps the prototype on the established topic and limits the drift caused by frequent low-density messages; content in other applications changes continuously, which the larger rate follows.

When E_t^* is undefined, deferred attribution compares z_t with the pending observations. If $\max_{p\in\mathcal{P}} \mathrm{sim}(z_t, z_p) \geq \tau_p$, the maximizing p^* and o_t form a new event in which p^* initializes the prototype and o_t is admitted by Eq. 4, both with $\rho = \textsc{promote}$; otherwise $(o_t, \hat{E}_t, t)$ enters $\mathcal{P}$. Setting $\tau_p > \tau_a$ requires stronger evidence to form an event than to extend one. A pending observation older than Δ_p is appended to its main event with $\rho = \textsc{absorb}$ and without a prototype update, so an unrelated interruption stays in that event's record without shifting its prototype; a pending observation with no main event forms an isolated event, which does not enter $\mathcal{A}_t$ and is reached through $\mathrm{index}(\cdot)$. Algorithm 1 gives the procedure, including the end-of-period merge of events that share a merchant key.

3.3 Refinement and Grounded Extraction

SAGE (Split, Anchor, Ground, Extract) revises σ in batches after later evidence has accumulated. The batch queue $\mathcal{Q}$ holds events created or extended since the previous batch, isolated events, and events whose share of ABSORB observations exceeds a threshold. For each $E \in \mathcal{Q}$, an LLM receives M_E, the canonical keys of its observations, the topic anchors whose intervals intersect I_E, and a split indicator computed from the cohesion $\mathrm{coh}(E) = |M_E|^{-1} \sum_{o_t \in M_E} \mathrm{sim}(z_t, c_E)$ (Appendix H), and acts on E through three tools; an output that violates its tool's constraint is returned to the LLM for revision. SplitEvent, triggered by the split indicator, separates the mixture left by absorption: a temporal clustering pass groups M_E, the LLM assigns the clusters to activities, and each activity becomes a child event that inherits its observations and entity anchors. SummarizeEvent writes the summary and type of E and links it to entity anchors from its canonical keys and to topic anchors, merging a proposed topic into a similar existing one so that one activity shares a topic anchor across events. ExtractFacts returns facts from M_E; a fact similar to an existing fact of the same event revises it and merges their supports.

Consolidation then covers the changed events $\mathcal{D}$ and their anchor neighborhood $\mathcal{N}(\mathcal{D}) = \{E' : \exists E \in \mathcal{D},\ A_{E'} \cap A_E \neq \emptyset\}$: the LLM receives their facts and the most similar long-term memories, and returns new or revised ones whose support is a nonempty subset of $\mathcal{D} \cup \mathcal{N}(\mathcal{D})$.

3.4 Retrieval over the Event Graph

Given a query q, we combine identifier-matched events retrieved through $\mathrm{index}(\cdot)$ with dense seeds from facts, event summaries, and long-term memories, giving identifier matches precedence. Their associated events provide entry points for one-hop expansion through shared anchors, with degree-discounted propagation. Candidates are merged, deduplicated, ranked, and retained within a fixed budget. A controller checks whether the retained evidence supports an answer under the query's contextual and temporal constraints. If not, it follows an anchor or opens an event or long-term memory to collect further evidence. Key coverage guides navigation but does not establish evidence sufficiency. Navigation is bounded by a step limit and a per-step candidate cap. Returned evidence preserves its links to supporting observations, while navigation paths record how it was found. Appendix G gives the procedure and controller prompt.

4 Experiments

4.1 Experimental Setup

Baselines. We compare three baseline groups on PMM-Bench: **reference baselines** that directly access observations (long context and flat dense retrieval), **compact-entry memory** that consolidates history into compact entries (Mem0(Chhikara et al., 2025), SimpleMem(Liu et al., 2026b), and LightMem(Fang et al., 2026)), and **relational memory** that organizes connections among memory units (A-Mem(Xu et al., 2026b), StructMem(Xu et al., 2026a), and CompassMem(Hu et al., 2026a)). Descriptions and adaptations appear in Appendices D and D.5, and related work in Appendix B. We also evaluate an identifier control: CompassMem with DRED's identifier index and no other changes. An LLM judge scores answers by question type and overall, with text-overlap F1 between generated and reference answers as a complementary metric.

4.2 Main Results

Table 3 reports the comparison; Table 15 repeats it with GPT-5-mini as the response model, and Appendix F.2 compares build and query costs. (1) Both memory families outperform the reference baselines, mainly on the four task types beyond single-hop recall: compact-entry memory highlights useful information through extraction and consolidation, and relational memory organizes distributed context through connections among memory units.

(2) HyperOS-Mem achieves the best results on all five task types. Over SimpleMem and CompassMem, the strongest unmodified baselines of the two families, it improves Judge/F1 by 12.7/11.3 and 8.5/7.5 points, with the largest gains on cross-application aggregation (18.1 and 13.3 points). DRED maintains cross-source event attribution online and defers uncertain decisions, and SAGE uses later evidence to revise events, extract facts, and induce patterns across events, which supplies evidence at the granularities these tasks require.

Table 3: Accuracy on PMM-Bench. Single: single-hop recall; Cross: aggregation across applications; Temp: temporal reasoning; Track: event tracking; Habit: habit induction. Long context is bounded by a 128K window, which holds under 2% of one user's textualized stream. The identifier control grants CompassMem the identifier index of DRED. The response model is GPT-4.1-mini and the judge model is GPT-4o. Judge is the unweighted mean of the five per-type scores.

Method	**Question type**					**Overall**	
	Single	*Cross*	*Temp*	*Track*	*Habit*	**Judge**	**F1**
References							
Long context (128K)	22.8	4.2	7.6	5.9	10.0	10.1	8.7
Flat dense retrieval	43.6	14.8	19.2	16.5	19.7	22.8	19.9
Compact-entry memory							
SimpleMem	42.3	25.4	28.2	26.8	29.6	30.5	27.1
LightMem	38.7	22.6	25.9	24.3	27.2	27.7	24.4
Mem0	40.2	18.9	22.4	19.6	24.8	25.2	22.3
Relational memory							
A-Mem	41.5	26.9	29.4	28.6	30.8	31.4	27.9
StructMem	40.9	23.8	27.6	25.1	27.9	29.1	25.8
CompassMem	43.1	30.2	34.8	33.5	32.1	34.7	30.9
↬ + *identifier index*	43.4	35.6	35.4	35.9	32.8	36.6	32.6
HyperOS-Mem	**48.2**↑5.1	**43.5**↑13.3	**41.1**↑6.3	**42.8**↑9.3	**40.6**↑8.5	**43.2**↑8.5	**38.4**↑7.5

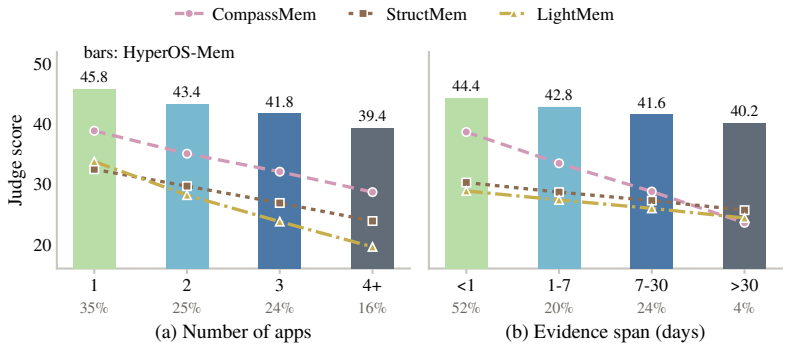
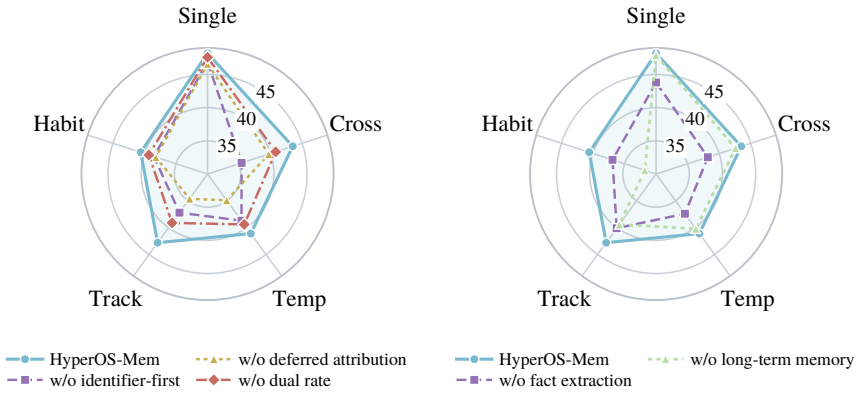


Figure 4: Accuracy on controlled slices of PMM-Bench, grouping questions by (a) the number of applications and (b) the time span of their annotated evidence. Bars show HyperOS-Mem and lines show baselines; the percentage under each slice is its share of questions.

(3) DRED's identifier index alone raises CompassMem from 34.7 to 36.6 Judge, mainly on cross-application aggregation (+5.4) and event tracking (+2.4), with +0.3 on single-hop recall. Identifier matching thus helps connect distributed evidence, yet the augmented CompassMem still trails HyperOS-Mem by 6.6 Judge and 5.8 F1 points.

4.3 Where the Gain Comes From

Figure 4 groups questions by the properties of their annotated evidence. (a) As evidence spreads from one application to four or more, the gap between HyperOS-Mem and LightMem widens from 12.0 to 19.8 points: the compact entries of LightMem do not record which observations belong to the same event, while StructMem and CompassMem keep links among memory units and degrade less. (b) As the evidence span grows from under one day to over one month, the gap to CompassMem widens from 5.7 to 16.6 points; StructMem anchors relations in time and LightMem consolidates long-term memory, so both drop by only about 4.5 points. HyperOS-Mem stays strong along both axes, owing to identifier-first and deferred attribution.



(a) Rules of DRED (b) Components of SAGE

Figure 5: Ablation of (a) DRED and (b) SAGE, removing one component at a time (Table 12).

4.4 Further Analysis

4.4.1 Ablation Study

In Figure 5(a), removing identifier-first attribution lowers cross-application aggregation by 8.1 points but single-hop recall by only 1.3, so identifiers mainly link evidence across applications. Closing the pending set lowers event tracking and temporal reasoning by 8.2 and 6.2 points, as deferral keeps activity stages intact until boundaries settle; a single update rate costs 3.7 points on tracking. In (b), removing fact extraction lowers every task type by 2.7–5.3 points, and removing long-term memory lowers habit induction by 8.8 points, which rests on cross-event regularities.

Table 4: DRED as a plug-in.

Method	Question type			Overall
	Single	*Cross*	*Track*	**Judge**
LightMem	38.7	22.6	24.3	27.7
+ DRED	**39.1**↑0.4	**28.2**↑5.6	**30.0**↑5.7	**30.9**↑3.2
StructMem	40.9	23.8	25.1	29.1
+ DRED	**41.2**↑0.3	**29.6**↑5.8	**31.8**↑6.7	**32.4**↑3.3
CompassMem	43.1	30.2	33.5	34.7
+ DRED	**43.6**↑0.5	**38.4**↑8.2	**38.6**↑5.1	**38.2**↑3.5
HyperOS-Mem	**48.2**	**43.5**	**42.8**	**43.2**

4.4.2 DRED as a plug-in

Table 4 attaches DRED to three memory systems in place of their own segmentation, leaving the rest of each system unchanged. All three gain, by 3.2 to 3.5 points overall. CompassMem with DRED reaches 38.2, above the 36.6 it reaches with the identifier index alone (Table 3), so deferral and the dual rate contribute beyond identifier matching. The three systems with DRED stay 5.0 to 12.3 points below HyperOS-Mem, which keeps the events DRED forms in a graph and lets SAGE extract facts and long-term memory from them.

4.4.3 Generalization

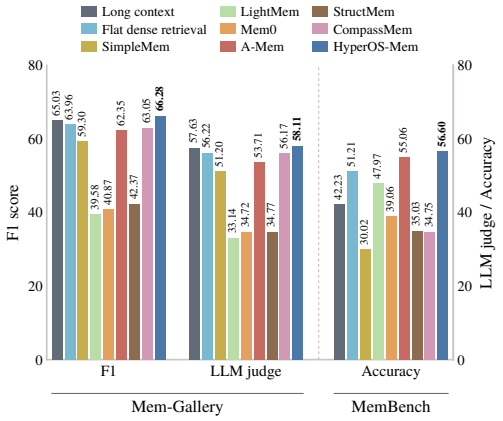
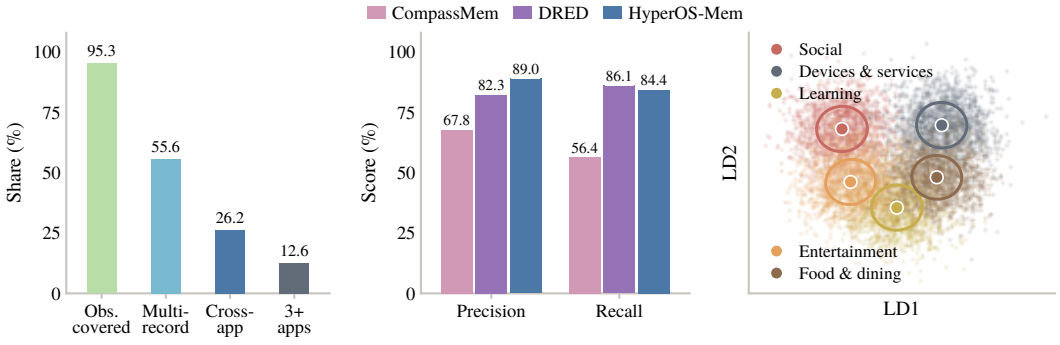


Figure 6: **Results on Mem-Gallery and MemBench.** F1 is read on the left axis, LLM judge and accuracy on the right. Per-type results are in Tables 13 and 14.

We evaluate HyperOS-Mem on two public memory benchmarks under their official protocols (Figure 6). On Mem-Gallery, a multimodal dialogue benchmark, it scores above every memory system and level with long context, which fits the whole dialogue in its window. On MemBench, it leads the strongest baseline by 1.5 points overall and leads on observational factual questions: events locate the relevant activity, and each event keeps its source observations.

4.4.4 Quantitative Analysis for Event Construction

In Figure 7(a), DRED groups 157,684 observations from one participant's stream into 25,853 events, covering 95.3% of the source observations; 55.6% of the events contain multiple observations, 26.2% integrate evidence from different applications, and 12.6% span three or more applications. Against human-annotated segmentations (b), DRED raises recall over CompassMem from 56.4 to 86.1, keeping one activity continuous within interleaved OS streams, and the slow-path refinement raises precision from 82.3 to 89.0. DRED keeps a slightly higher recall than the full system as it avoids splitting evolving events online, whereas SAGE later separates wrongly merged observations.



(a) Events formed by DRED (b) Construction quality (c) Semantic domains after SAGE

Figure 7: Event organization on one participant's full stream. (a) Coverage of observations by DRED events, and shares of events with multiple records, multiple applications, and three or more applications. (b) Precision and recall of event segmentation against human annotation. (c) Long-term memories consolidated by SAGE, projected to two dimensions and colored by semantic domain.

4.4.5 Qualitative analysis of the hierarchical event graph.

Figure 8 shows 190 observations from about ten minutes of campus-network troubleshooting. Observations of `E816` separated by 5 minutes and 56 seconds of gaming, messaging, and Dianping browsing remain in one event, showing continuity across interruptions. Evidence of the task spreads over three events: `E816` holds a WeChat help request (**A**) and browser views of the network-plan page (**B**), `E818` holds the engineer communication and contact searches, and `E814` holds the user-typed recovery report (**C**). A shared contact identifier links `E816` and `E818`, and a shared campus-network topic relates `E818` to `E814`, so the graph recovers the task progression. Events thus preserve evidence continuity across time and applications, while anchors organize complementary evidence across events into task-level memory grounded in the original observations. Appendix I gives further cases.

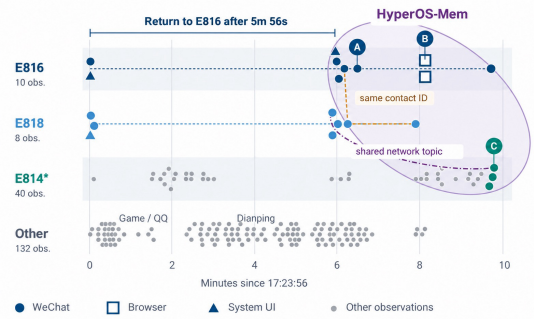


Figure 8: A qualitative case of the hierarchical event graph. The x-axis is elapsed time; rows are final events (gray: others); A–C mark evidence.

5 Conclusion

We extend personal memory beyond isolated interaction domains by introducing the first OS-level memory framework from smartphone observation streams. We build PMM-Bench, a benchmark with 3.83M records from 20 users, enabling systematic evaluation of cross-application and long-horizon memory reasoning. We further propose HyperOS-Mem, a fast–slow hierarchical architecture that efficiently organizes massive OS observations while supporting provenance-aware extraction of cross-application evidence. Experiments show that HyperOS-Mem outperforms existing memory systems on PMM-Bench, remains competitive on public memory benchmarks, and that DRED improves other memory systems as a plug-in.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing experiment code. We have not used generative AI tools to generate data, formulate claims, propose hypotheses, design the methodology or experiments, or interpret results. Additionally, we used generative AI tools to edit the paper for readability. We have reviewed all AI-assisted work: AI-assisted code was checked and tested by the authors, and AI-edited text was checked against its intended meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI. Separately, generative models are components of the system under study: a VLM converts screen operations and gallery images into text, an LLM calls the SAGE tools and consolidates long-term memory, and an LLM serves as judge in evaluation.

ETHICS STATEMENT

PMM-Bench is built from behavioral data collected from 20 consenting participants across 685 participant-days. Direct identifiers were removed, and the de-identified benchmark is distributed only through controlled access under a data-use agreement. Further details on consent, de-identification, validation, and release restrictions are provided in Appendix C.5.

REPRODUCIBILITY STATEMENT

The method is specified in Section 3 with DRED in Algorithm 1; models and evaluation details are provided in Appendices E, and baseline adaptation is described in D.5.

REFERENCES

Lei Bai, Zongsheng Cao, Yang Chen, Zhiyao Cui, Shangheng Du, Yue Fan, Shiyang Feng, Zijie Guo, Haonan He, Liang He, et al. Scaling the horizon, not the parameters: Reaching trillion-parameter performance with a 35b agent. *arXiv preprint arXiv:2606.30616*, 2026.

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-gallery: Benchmarking multimodal long-term conversational memory for mllm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40750–40784, 2026.

Huacan Chai, Yukai Wang, Yingxuan Yang, Dan Peng, Yuanyi Song, Zhihui Fu, Weiwen Liu, Jianghao Lin, Jun Wang, and Weinan Zhang. Smmbench: A benchmark for source-distributed multimodal agent memory. *arXiv preprint arXiv:2605.15710*, 2026.

Yijun Chen, Boyi Xiao, Yixian Zhao, Haoting Xia, Buqiang Xu, Jizhan Fang, Yanya Li, Yaqi Zheng, Xuehai Wang, Zirui Xue, et al. Lightmem-ego: Your ai memory for everyday life. *arXiv preprint arXiv:2607.11487*, 2026a.

Yijun Chen, Yaqi Zheng, Yanya Li, Boyi Xiao, Buqiang Xu, Shuofei Qiao, Jizhan Fang, Xinle Deng, Yunzhi Yao, Xuehai Wang, et al. Emˆ 2mem: Event-centric multimodal memory for large language models. *arXiv preprint arXiv:2609.00551*, 2026b.

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

Wonjun Choi, Yerim Kim, Yukyung Lee, and Susik Yoon. Pgmem: Tightly coupled persona-memory graph for lifelong personalized agents. *arXiv preprint arXiv:2608.01708*, 2026.

Xinle Deng, Yida Xue, Xiangyuan Ru, Haoming Xu, Shuofei Qiao, Mengru Wang, Yijun Chen, Buqiang Xu, Chen Jiang, Yuchen Eleanor Jiang, et al. MobileMem: Learning from a year of mobile experiences. *arXiv preprint arXiv:2608.13606*, 2026a.

Xinle Deng, Ruobin Zhong, Hujin Peng, Xiaoben Lu, Yanzhe Wu, Guang Li, Buqiang Xu, Yunzhi Yao, Jizhan Fang, Haoliang Cao, et al. Memtrace: Tracing and attributing errors in large language model memory systems. *arXiv preprint arXiv:2605.28732*, 2026b.

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.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 55–65, 2019.

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

Muxin Fu, Xiangyuan Xue, Yafu Li, Zefeng He, Siyuan Huang, Xiaoye Qu, Yu Cheng, and Yang Yang. Latentmem: Customizing latent memory for multi-agent systems. *arXiv preprint arXiv:2602.03036*, 2026.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Weidong Guo, Dakai Wang, Zixuan Wang, Hui Liu, and Yu Xu. T-mem: Memory that anticipates, not archives. *arXiv preprint arXiv:2606.15405*, 2026.

Zirui Guo, Xubin Ren, Lingrui Xu, Jiahao Zhang, and Chao Huang. Rag-anything: All-in-one rag framework. *arXiv preprint arXiv:2510.12323*, 2025a.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *EMNLP (Findings)*, pp. 10746–10761, 2025b.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *International Conference on Machine Learning*, 2025.

Yuyang Hu, Jiongnan Liu, Jiejun Tan, Yutao Zhu, and Zhicheng Dou. Memory matters more: Event-centric memory as a logic map for agent searching and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22389–22407, 2026a.

Yuyang Hu, Hongjin Qian, Shuting Wang, Jiongnan Liu, Ziliang Zhao, Jiejun Tan, Zheng Liu, and Zhicheng Dou. Sam: State-adaptive memory for long-horizon reasoning agent. *arXiv preprint arXiv:2605.24468*, 2026b.

Shuo Ji, Yibo Li, and Bryan Hooi. Memory is reconstructed, not retrieved: Graph memory for llm agents. *arXiv preprint arXiv:2606.06036*, 2026.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yu, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9119–9130, 2020.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, et al. Llmrouterbench: A massive benchmark and unified framework for llm routing. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37733–37754, 2026a.

Mo Li, LH Xu, Qitai Tan, Long Ma, Hongyong Song, Ting Cao, and Yunxin Liu. Sculptor: Empowering llms with cognitive agency via active context management. In *International Conference on Learning Representations*, volume 2026, pp. 153411–153440, 2026b.

Songze Li, Zhiqiang Liu, Zhengke Gui, Huajun Chen, and Wen Zhang. Enrich-on-graph: Query-graph alignment for complex reasoning with llm enriching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7683–7703, 2025.

Guangyi Liu, Pengxiang Zhao, Yaozhen Liang, Qinyi Luo, Shunye Tang, Yuxiang Chai, Weifeng Lin, Han Xiao, WenHao Wang, Siheng Chen, et al. Memgui-bench: Benchmarking memory of mobile gui agents in dynamic environments. *arXiv preprint arXiv:2602.06075*, 2026a.

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

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

Peng Liu, Zhen Tao, Jihao Zhao, Ding Chen, Yansong Zhang, Cuiping Li, Zhiyu Li, and Hong Chen. Mememo: Evaluating emotion in memory systems of agents. *arXiv preprint arXiv:2602.23944*, 2026c.

Shuai Liu, Shulin Tian, Kairui Hu, Yuhao Dong, Zhe Yang, Bo Li, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Filegram: Grounding agent personalization in file-system behavioral traces. *arXiv preprint arXiv:2604.04901*, 2026d.

YuFei Luo, Xiucheng Xu, and Zhen Yang. Memsif: From structured interactions to dual-track fact memory for llm agents. *arXiv preprint arXiv:2608.01742*, 2026.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Jingbiao Mei, Jinghong Chen, Guangyu Yang, Xinyu Hou, Margaret Li, and Bill Byrne. According to me: Long-term personalized referential memory qa. *arXiv preprint arXiv:2603.01990*, 2026.

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

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.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon llm agent via context-folding. *arXiv preprint arXiv:2510.11967*, 2025.

Juntong Wang, Haoyue Zhao, Xiyuan Wang, Yanbo Wang, Qiyan Deng, Muhan Zhang, et al. Sage: A self-evolving agentic graph-memory engine for structure-aware associative memory. *arXiv preprint arXiv:2605.12061*, 2026.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. In *International Conference on Learning Representations*, volume 2026, pp. 11202–11216, 2026.

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

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. Structmem: Structured memory for long-horizon behavior in llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146, 2026a.

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*, 38:17577–17604, 2026b.

Yunyi Xuan, Hao Yi, Fengling Mao, Daye Cai, Leikun Liang, Xingsheng He, Jiangnan Xie, Guoshuai Wang, Yushan Han, Wenwen Guo, et al. Usergpt technical report. *arXiv preprint arXiv:2605.08766*, 2026.

Chengyuan Yang, Zequn Sun, Wei Wei, and Wei Hu. Beyond static summarization: Proactive memory extraction for llm agents. *arXiv preprint arXiv:2601.04463*, 2026a.

Ke Yang, Zixi Chen, Xuan He, Jize Jiang, Michel Galley, Chenglong Wang, Jianfeng Gao, Jiawei Han, and ChengXiang Zhai. Plugmem: A task-agnostic plugin memory module for llm agents. *arXiv preprint arXiv:2603.03296*, 2026b.

Woongyeong Yeo, Kangsan Kim, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang. Universalrag: Retrieval-augmented generation over corpora of diverse modalities and granularities. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3843–3871, 2026a.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25599–25609, 2026b.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. Hypermem: Hypergraph memory for long-term conversations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35237–35254, 2026.

Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, et al. Ama-bench: Evaluating long-horizon memory for agentic applications. *arXiv preprint arXiv:2602.22769*, 2026.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

APPENDIX

A Limitations **14**

B Related Work **14**

C PMM-Bench Details **16**
C.1 Data Collection 16
C.2 Data Composition 16
C.3 Atomic Observation Construction 17
C.4 QA Construction and Evidence Annotation 22
C.5 Ethics and Release 23

D Baselines and Adaptation **24**
D.1 Reference Baselines 24
D.2 Compact-Entry Memory 24
D.3 Relational Memory 24
D.4 Input Adaptation 25
D.5 Adaptation Protocol 25

E Models and Evaluation Details **26**

F Additional Results **27**
F.1 Component analysis by question type. 27
F.2 Build and Query Overhead. 27
F.3 Additional benchmark comparisons. 27

G Retrieval Details **28**

H Hyperparameters **30**

I Case Study **31**

A LIMITATIONS

Our conclusions are based on device data from 20 Android users, and broader validation across users, devices, languages, and OS ecosystems would further strengthen their generality. The current benchmark uses fixed DRED thresholds and SummarizeEvent anchors to ensure consistent event construction, while SAGE is evaluated with a fixed model configuration. These choices support reproducible comparisons; future work can examine additional data sources and configurations under the same evaluation protocol.

B RELATED WORK

MEMORY SYSTEM

Memory is the cognitive foundation for agents to maintain persona consistency and reason over extended time spans. Early methods addressed context window limitations by externalizing history into flat vector databases (Park et al., 2023; Packer et al., 2023). Although effective for semantic matching, such approaches fundamentally treat interaction history as a collection of unordered

propositions, severing temporal progression, causal dependencies, and relational structure, which are the key elements for binding isolated events into a coherent narrative (Gao et al., 2023; Liu et al., 2024). This flat representation leads to fragmented retrieval, returning isolated facts in the absence of contextual support required for complex reasoning (Weller et al., 2026; Li et al., 2026b).

To bridge this gap, recent research has shifted towards structured architectures oriented towards reasoning (Yue et al., 2026; Ji et al., 2026; Wang et al., 2026). GraphRAG (Edge et al., 2024) and HippoRAG (Gutiérrez et al., 2025) organize document knowledge with entity relationship graphs and community summaries, LightRAG (Guo et al., 2025b) combines graph indexing, dual-layer retrieval, and incremental updates to improve the efficiency and adaptability of knowledge access, and GAM connects interaction encoding with memory consolidation through local event graphs and a global topic network, enabling memory to evolve from discrete fact storage toward multi-level contextual organization. However, such architectures face two major limitations when applied to continuous OS observation streams: First, compressing fluent narratives into rigid entity-relation triples leads to semantic loss (Fu et al., 2026; Li et al., 2025; Bai et al., 2026), and the instability of extraction can cause illusory relations to propagate as persistent structural noise (Li et al., 2026a; Guo et al., 2025a; Deng et al., 2026b); Second, their index construction relies on offline, expensive full-scale processing, making it difficult to adapt to continuously arriving and evolving OS observation streams.

Long-Term Personal Memory

Current research on long-term personal memory has mainly focused on organizing and updating historical information (Guo et al., 2026), as well as modeling its relationship with user representations. MemoryBank supports personalized user understanding through long-term interactive memory and continuous updates, MemoryOS adopts hierarchical storage to manage short-term, medium-term, and long-term personal memory, and A-MEM constructs an evolvable memory network through dynamic indexing and association (Zhong et al., 2024; Kang et al., 2025; Xu et al., 2026b). At the memory organization level, CompassMem constructs event graphs through incremental event segmentation and explicit relationships, MemSIF uses topical segments and event trajectories to model the continuity of events across time, and StructMem further explores event binding and hierarchical organization (Hu et al., 2026a; Luo et al., 2026; Xu et al., 2026a). Research on user modeling has further extended from historical information retrieval to long-term behavioral understanding (Liu et al., 2026c; Sun et al., 2025).UserGPT forms user profiles and narrative summaries from long and noisy behavioral trajectories, and PGMem connects profile signals to historical events that support or revise them through provenance and evidence edges (Xuan et al., 2026; Choi et al., 2026). However, these event-structuring approaches generally rely on explicit boundary signals—such as dialogue turns, video shot transitions, or document paragraph structures—to trigger event segmentation and consolidation. OS-level observation streams lack such natural boundaries: events are implicitly defined by semantic consistency, and multiple event threads are temporally interleaved. This paper focuses on a finer-grained stage before these memory representations are formed: how to progressively form evolvable events from low-level OS atomic observations that do not possess event boundaries, and maintain their connection to the original evidence during subsequent abstraction processes.

Multimodal Memory Benchmark

Multimodal and mobile memory research has further expanded the observation sources and evaluation scenarios underlying personal memory. Mem-Gallery studies long-term vision-language conversational memory, WorldMM constructs complementary episodic, semantic, and visual memories for long videos, and LightMem-Ego continuously accumulates visual and audio experiences from smartphones and wearable devices (Bei et al., 2026; Yeo et al., 2026b; Chen et al., 2026a). MobileMem, which is closer to the scenario presented in this paper, synthesizes long-term trajectories from mobile application experiences, ATM-Bench covers years of personal materials such as images, videos, and emails, DynamicMem simulates long-term multi-application behavioral changes, and SMMBench specifically examines the combination of multimodal evidence scattered from independent sources (Deng et al., 2026a; Zhao et al., 2026; Chai et al., 2026). PMM-Bench takes continuously generated underlying Android system records from real users as input and evaluates the memory system's ability to perform long-term memory and event-level reasoning on fine-grained, temporally interleaved, and cross-application observation streams.

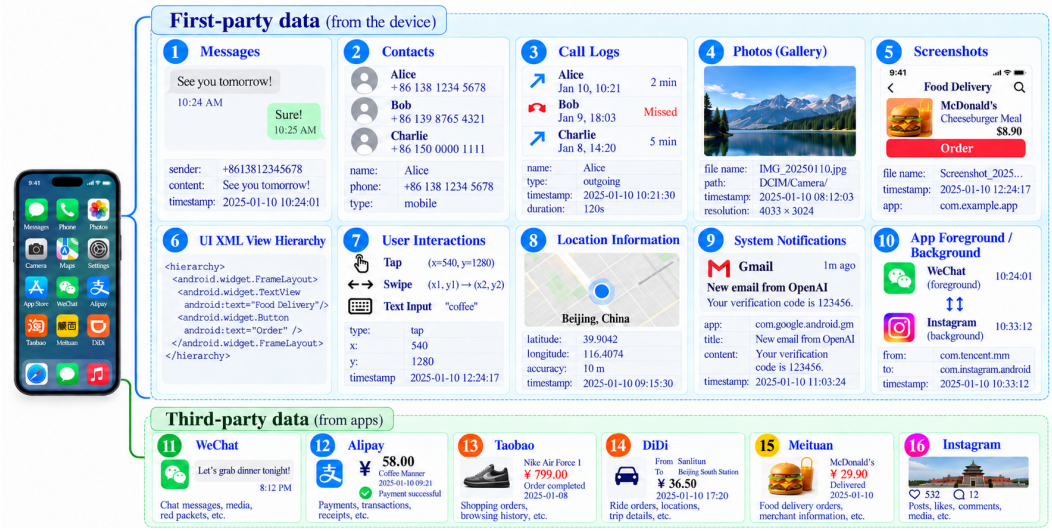


Figure 9: Trace families recorded by the on-device logger. Records of all families share a single timeline through the system timestamp and the host application identifier.

C PMM-BENCH DETAILS

C.1 DATA COLLECTION

Each participant installed a logger on their device to collect ten types of traces generated during everyday smartphone use, including messages, contact lists, call logs, gallery photos, screenshots, XML view hierarchies of the current interface, user interaction events such as taps, swipes, and text input, location information, system notifications, and application foreground/background transitions (Figure 9). For tap events, the logger records the corresponding screen coordinates, and the target UI element can be further inferred by matching these coordinates with the XML view hierarchy captured at the same time. Each record contains a system timestamp and an identifier of the associated application, allowing heterogeneous observations from different sources and data types to be aligned on a unified timeline. Screenshots and XML view hierarchies are not captured at a fixed sampling rate; instead, they are triggered by user interaction events so that data collection remains synchronized with actual device usage. All records are stored locally on the participant's device during the collection period and are exported only after being reviewed and confirmed by the participant at the end of the study. The consent procedure, the exclusion of the applications that participants marked as private, and the de-identification applied before release are described in the Ethics statement.

C.2 DATA COMPOSITION

In the largest stream (Figure 1), sensor records account for 51.6% of 699,798 records, screen operations for 22.0%, application switches for 13.7%, and GPS locations for 7.0%, whereas SMS messages (1,148) and screenshots or gallery items (312) together contribute approximately 0.21%. Daily volume ranges from about 2,000 to over 15,000 records, and the mix of applications changes from day to day. The observations cover eight everyday scenarios, defined in Table 5. Scenarios describe activity contexts: one application may support several scenarios, and one activity may span several applications.

To further characterize PMM-Bench from the user perspective, we analyze the usage data of 20 users, including behavior-type composition, perceptual modalities, activity levels, and application usage. These statistics collectively reflect the diversity of the benchmark's data streams and provide context for the subsequent analysis of cross-modal, cross-application, and long-term temporal memory capabilities. Figure 10 shows that users are distributed across multiple behavior types. Figure 11 reveals substantial differences among users in active days and major behavior-record counts, which exhibit only a weak negative correlation ($r = -0.235$). Figure 12 further illustrates individualized application usage: some users' records are highly concentrated in a single application, whereas others exhibit more distributed cross-application activity. Overall, these three views demonstrate the diversity of user behavior scenarios and the substantial variation in user-level data scale and compo-

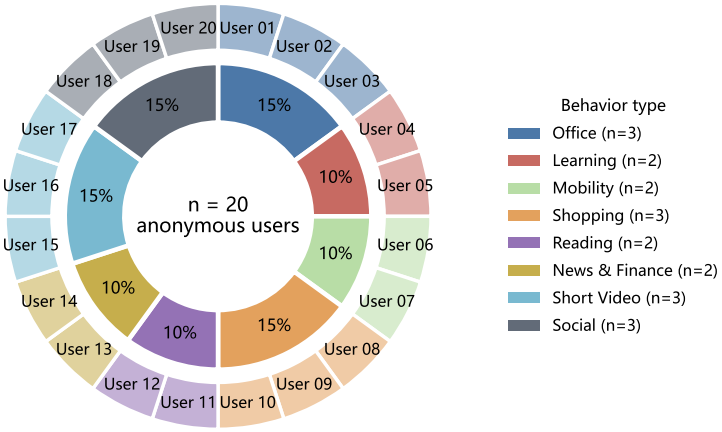


Figure 10: Anonymous user composition by behavior type. The inner ring shows the sampling quota and the outer ring identifies the 20 users as User 01–User 20.

sition. For a more detailed view of individual streams, Figure 13 shows the record composition and daily volume of four representative users. The figure complements the aggregate user-level analysis by illustrating how record-family proportions and daily activity vary within individual streams.

Table 5: Everyday scenarios covered by PMM-Bench.

Scenario	Definition and example
News	Accessing news and public information, such as reading a report and searching for related background.
Short video	Viewing or interacting with short-form videos, such as browsing a feed, searching for a creator, and saving content.
Travel	Arranging transportation, accommodation, or navigation, such as booking a trip and coordinating pickup through chat.
Shopping	Comparing, purchasing, and following up on products, such as checking prices across platforms and asking about warranty terms.
Office work	Handling work-related communication, documents, and tasks, such as discussing edits and sending a revised document.
Reading	Reading books or long-form content, such as adding a bookmark and later resuming an e-book.
Study	Learning through courses, reference materials, or exercises, such as watching a lesson and taking notes.
Social interaction	Maintaining interpersonal relationships, such as exchanging updates, sharing content, and arranging meetings.

C.3 Atomic Observation Construction

This paper designs specific parsers for different types of raw mobile data. Table 6 summarizes the parsers and processing rules corresponding to each data type. Each parser first applies its specific filtering rules (Table 7) to remove invalid data and low-value noise, then performs information extraction, de-duplication, and semantic description based on the data type. Finally, all types of raw data are converted into a unified atomic observation representation. For data requiring further semantic information extraction or filtering, we employ corresponding prompts, including a visual content description prompt, a notification filtering prompt, and an SMS filtering prompt.

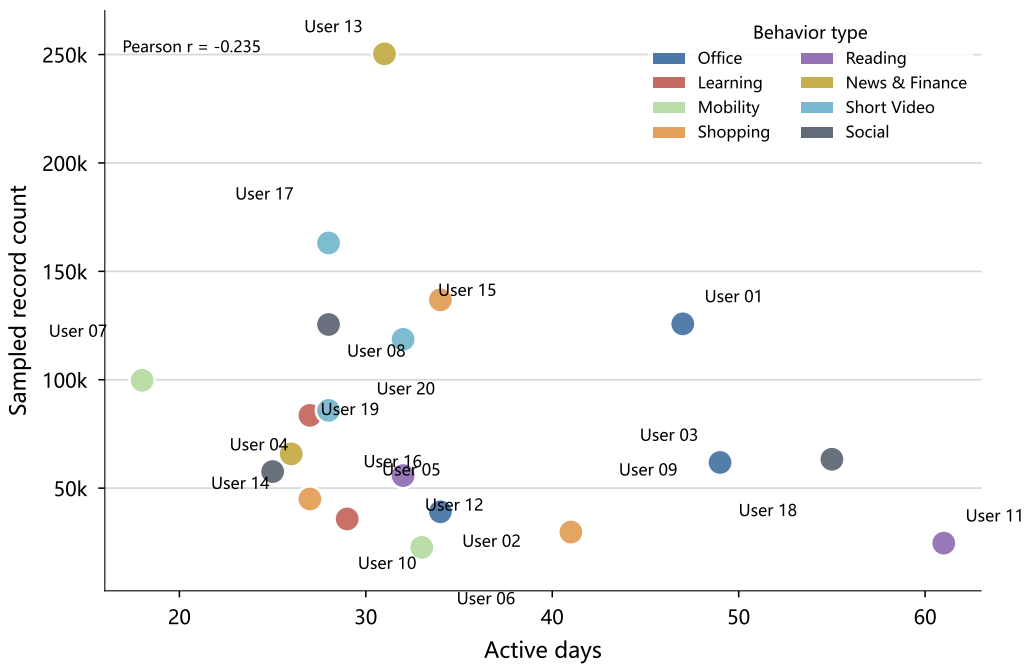


Figure 11: Active days versus sampled-record count for anonymous users. The record totals are descriptive benchmark counts, and the color indicates the behavior type.

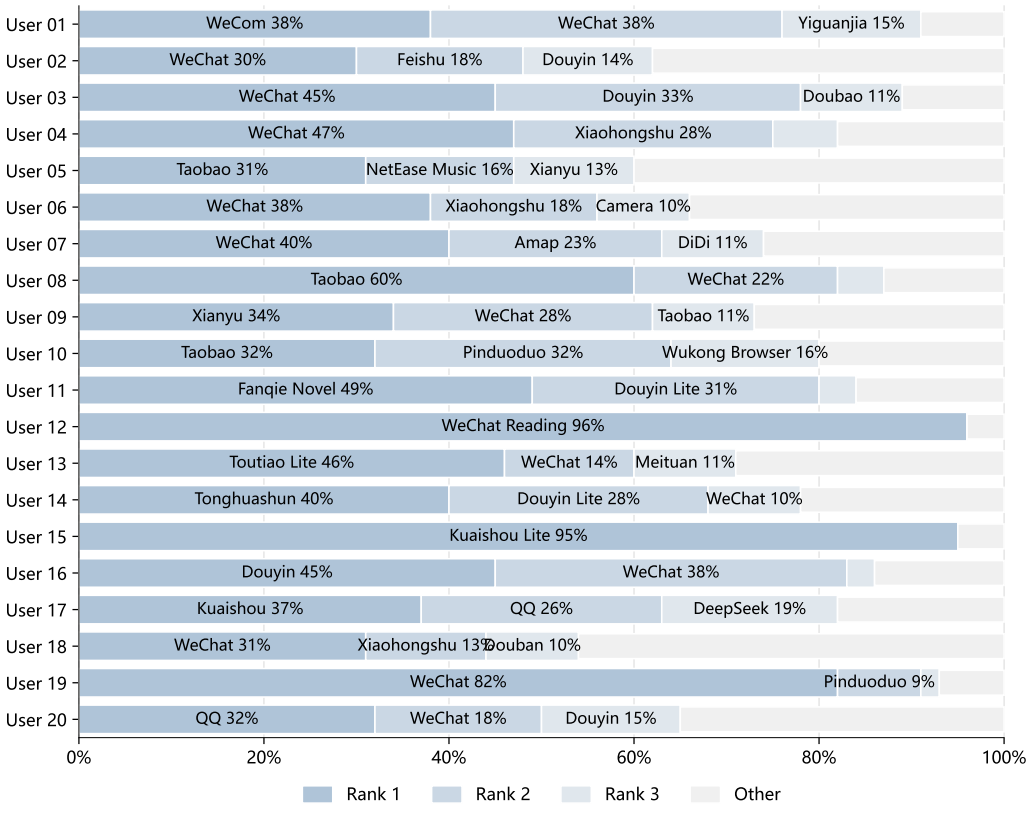


Figure 12: Within-user application concentration. Bars show the five highest-ranked application shares and the residual share without exposing application identities.

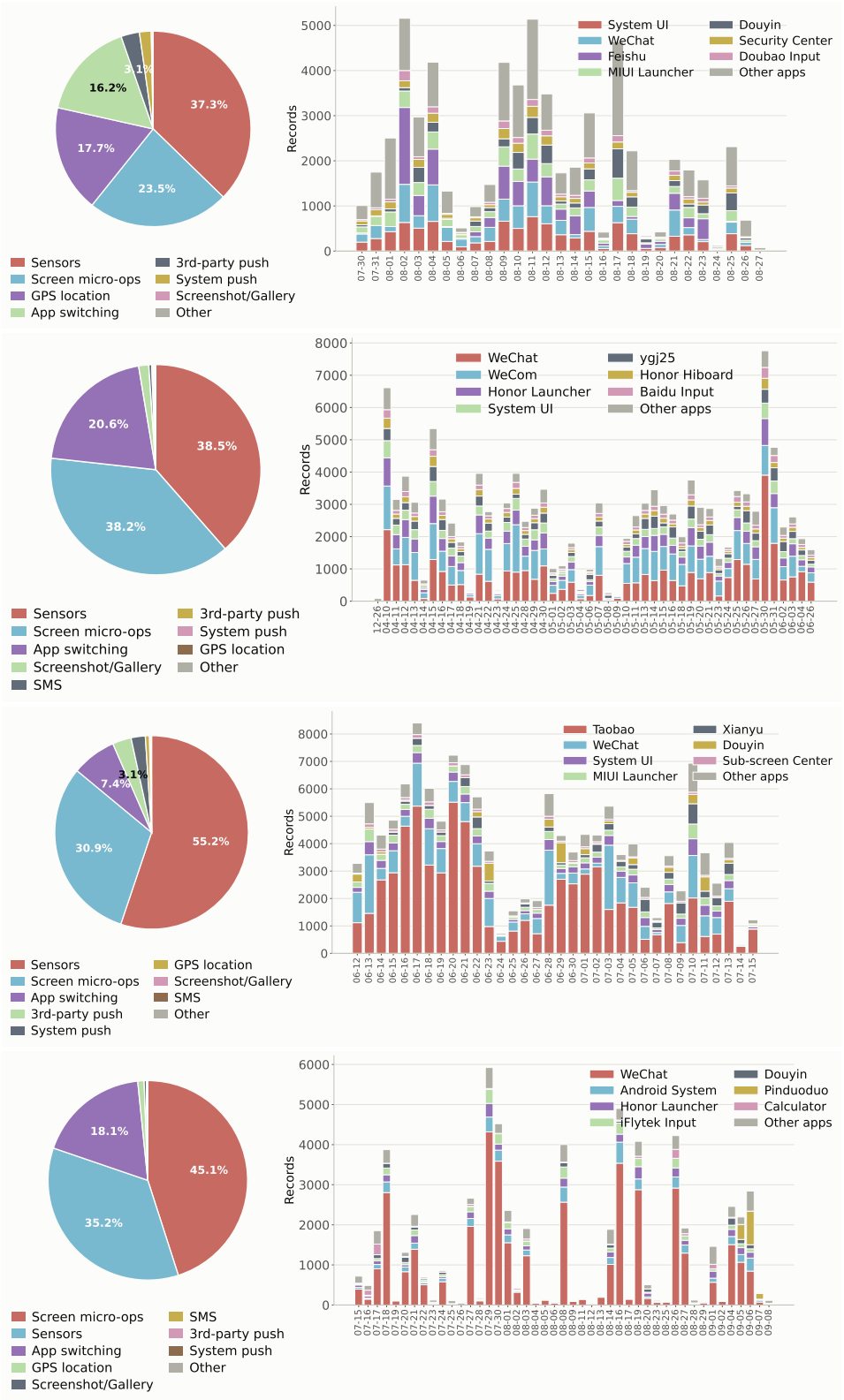


Figure 13: Composition and daily volume of four PMM-Bench streams, one user per row. Left: share of each record family. Right: records per active day, stacked by the seven most frequent applications and an aggregate of the remainder.

Table 6: Key processing operations for different types of mobile data.

Data type	Filter rules	Key processing
Screen/UI interaction	R0, R1a, R3, R4	OCR; visual–prompt fusion; semantic description
Gallery/file media	R0, R1a, R4	OCR; visible-content description
Notification	R0, R2, R4, R6, R7	Field extraction; lifecycle; template de-duplication
SMS	R0, R4, R5, R7	Sender/recipient extraction; direction determination
User-authored text	R0, R4	Text extraction; lifecycle processing
Calendar/task	R0, R4	Event extraction; state tracking
Device status/location	R0, R1a, R1b, R4	Invalid-state filtering; temporal aggregation
Audio recording	R0, R4	Hash de-duplication; transcription; summarization

Table 7: Filtering rules.

Rule ID	Scope	Filtering rule (condition; action)
R0	Untyped or action-free; all types	Missing type/action, invalid JSON, incomplete object, or no parseable event; remove and log the reason.
R1a	Device sensors	Accelerometer, Light, Gyroscope, or Battery perception; remove consecutive sensor noise.
R1b	Wi-Fi	Empty, "not connected", "¡unknown ssid¿", or `None`; remove empty connection states.
R2	Push	`active_name == PUSH_DELETE`; remove notification-clear events.
R3	Screen / UI	Pure gesture with no activity or key content; remove pure UI gestures.
R4	All types	Collector package or client-side cyclic-data marker; remove collector self-loop records.
R5	SMS	Advertising-number fragment and advertising term without a protected term; remove advertising SMS.
R6	Push	System-heartbeat title, such as charging or screen state; remove heartbeat notifications.
R7	Push / SMS	Empty `title` and `content`; remove empty communication records.
R8	All types	Fails a task-specific time, app, type, category, or keyword condition; remove or exclude from the task.

Prompt 1: Visual Content Description

```
You need to generate a factual Chinese semantic description from one
 raw mobile-data record.

The input includes the image associated with the record, together
with any available application name, time, user operation, and OCR
text.

Based on the input:
1. Describe the main content in the image that is relevant to the
current record.
2. Extract clearly visible or readable text, objects, products,
amounts, places, and dates.
3. Combine the image and OCR information into a concise and complete
 Chinese description.
4. Describe only information explicitly supported by the input; do
not fill in uncertain content.
5. If the image and OCR conflict, follow information that is clearly
 visible in the image.

For screen images and UI interactions:
- You may combine the application name, user operation, and image
content to describe the currently observable interaction.
```

```
- Do not infer that the user has purchased, paid, sent, arrived, or
completed an action merely from the page state.

For gallery and file media:
- Describe only content visibly present in the image or media.
- The existence of a media file does not prove that the user viewed,
 captured, downloaded, saved, liked, or shared it.
- Do not output user behavior or intent that is not directly
supported by the current record.

Output one concise, factual Chinese description.
```

Prompt 2: Notification Filtering

```
You need to determine whether one mobile notification has long-term
personal-memory value.

The input is one notification record, including its title, body,
source application, and relevant context.

Based only on the current record, determine whether it should be
retained.

Prefer retaining:
1. Security information, such as verification codes, logins, account
 security, and identity authentication.
2. Transaction information, such as charges, payments, transfers,
bills, and bank transactions.
3. Logistics information, such as express delivery, pickup, parcels,
 and delivery.
4. Work and scheduling information, such as meetings, projects,
contracts, invoices, and interviews.
5. Travel and medical information, such as flights, tickets,
hospitals, appointments, medication, and consultations.

Filter out:
1. Advertising and marketing information.
2. Recommendations, discounts, promotions, red packets, and other
short-term marketing content.
3. System operating status and low-value short-term noise.

Use only the current notification record. Do not infer whether the
user read, clicked, replied to, or took any subsequent action.

Output:
{
  "decision": "retain or delete",
  "category": "category",
  "reason": "reason"
}
```

Prompt 3: SMS Filtering

```
You need to determine whether one mobile SMS message has long-term
personal-memory value.

The input is one SMS record, including the sender or recipient,
message body, explicit operation, and relevant context.
```

```
Based on the current record, determine whether it should be retained
.

Prefer retaining:
1. Account security.
2. Transactions.
3. Logistics.
4. Work.
5. Customer communication.
6. Travel.
7. Medical information.
8. Messages explicitly sent by the user.

Filter out:
1. Advertising.
2. Marketing.
3. Recommendations.
4. Discounts.
5. Short-term campaigns.
6. Other low-value noise messages.

The sending or receiving direction may be determined only from an
explicit operation in the input; do not infer it from the message
body.

Do not infer whether the user read, understood, agreed with,
received, or took any subsequent action based on the message.

Output:
{
  "decision": "retain or delete",
  "category": "category",
  "reason": "reason"
}
```

C.4 QA Construction and Evidence Annotation

Figure 14 presents the construction and review workflow. Annotators inspect each user's atomic operations and remove or consolidate records that add no information, including empty transitions, collector notifications, and duplicate captures. They then group operations into coherent events according to the activity's objects, user intent, and progression. An event may span applications and temporal gaps, while interleaved unrelated activities remain separate. Each event retains its operations and timestamps and receives an event summary.

Events are organized into twelve categories according to their underlying purposes. These categories organize material for annotation, whereas the eight scenarios describe data coverage. For example, a car-purchase discussion in a messaging application belongs to transportation and travel. Ambiguous assignments are checked against the source operations. Annotators read the grouped events and write 90–120 questions per user from the user's first-person perspective. Each question is paired with a type label, a reference answer, and supporting evidence. Question selection follows the information available in the recorded history. Questions follow the five evaluation categories in Table 8. These categories describe the primary reasoning task required to answer each question.

Each QA pair retains an evidence summary and the relevant atomic operations. The summary identifies the supporting events and their temporal scope. Each operation includes its timestamp, source application, and recorded content, with links to source screenshots when available. These records allow the answer to be checked against specific observations. For example, an order record provides the temporal anchor for a question about what followed a purchase, while subsequent warranty-consultation records support the answer. Composite answers retain evidence for their constituent facts or steps. After draft QA pairs have been prepared, retrieval surfaces related events and source operations for human review. During the review stage, reviewers compare reference answers with

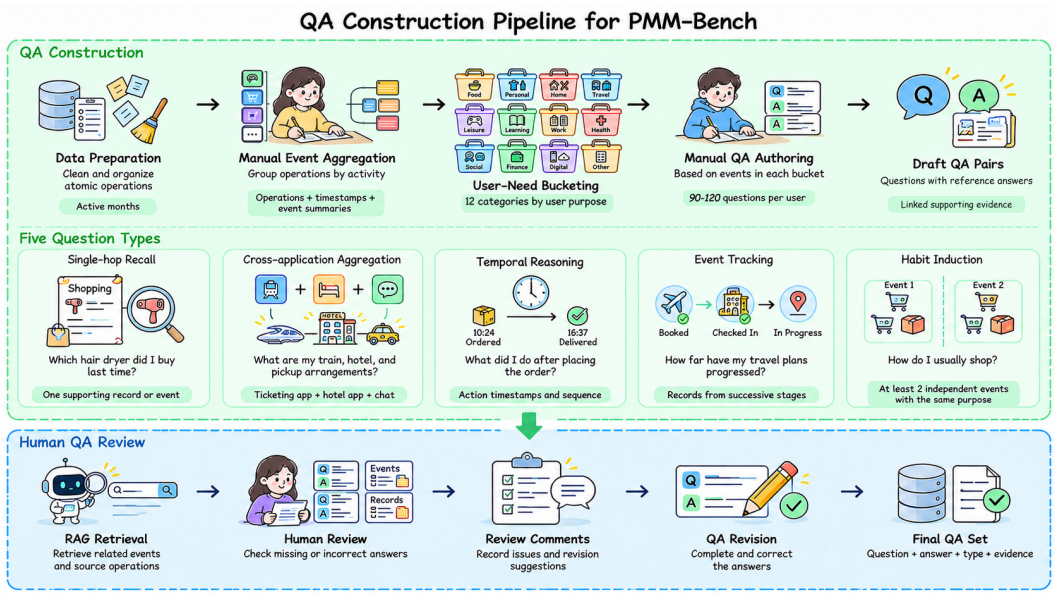


Figure 14: QA construction pipeline for PMM-Bench. Top: data preparation, manual event aggregation, user-need bucketing, and question authoring. Bottom: retrieval-assisted human review and revision of questions, reference answers, and supporting evidence.

Table 8: Question types, evidence requirements, and illustrative questions in PMM-Bench.

Type	Definition and evidence requirement	Example
Single-hop recall	Retrieve a specific fact directly supported by a record or event.	Which model of hair dryer did I buy last time?
Cross-application aggregation	Combine complementary evidence from multiple applications to answer the question.	What are my train, hotel, and pickup arrangements for this trip?
Temporal reasoning	Establish the order or timing of actions from timestamped observations.	What did I do after placing the hair-dryer order?
Event tracking	Connect stages of one activity to recover its progress and latest observed state.	How far has my thesis progressed?
Habit induction	Infer recurring behavior from at least two independent events with a shared purpose.	What do I usually do before ordering food delivery?

supporting observations and available screenshots through a dedicated interface, checking completeness, factual accuracy, evidence sufficiency, and consistency with the question's scope and time range. They record identified issues and revision suggestions, then return the QA pair for revision or approve it after verification.

Event-tracking answers describe the latest supported state; an unrecorded action is not treated as proof that it never occurred. Habit-induction evidence distinguishes independent events from repeated captures of the same activity.

C.5 Ethics and Release

Informed Consent. All participants provided informed consent prior to data collection and were informed of the research purpose, collection period, categories of collected data, intended use for research and benchmark construction, potential privacy risks, and the planned controlled release of the benchmark.

De-identification and Validation. Direct identifiers, including personal names, phone numbers, account identifiers, and detailed addresses, were removed or anonymized in text data, while corresponding sensitive information in visual data, such as screenshots, was visually masked. Screenshots, messages, notifications, location-related records, and free-form text were further screened

through rule-based filtering, automated content checks, and manual review. Records that could not be reliably de-identified were excluded from release.

Licensing and Controlled Release. Raw smartphone data are not released publicly. The de-identified PMM-Bench is provided only through controlled access and is governed by a Data Use Agreement (DUA). Applicants are required to provide their research purpose and institutional affiliation and agree not to attempt participant re-identification, contact participants, redistribute the data, combine the benchmark with external information for identity inference, or use the data beyond the approved research scope.

D Baselines and Adaptation

Table 9 compares the documented designs of the evaluated memory systems.

D.1 Reference Baselines

Long context. The textualized stream is placed in the context window of the backbone, and records beyond the window are truncated from the oldest end. A 128K window holds under 2% of one user's stream, so this reference measures what remains reachable without an intermediate memory unit rather than what a model could do with the whole history.

Flat dense retrieval. Normalized observations are embedded and retrieved by similarity, with no organization above the observation. It is the natural lower bound for structured memory and separates gains attributable to retrieval alone.

D.2 Compact-Entry Memory

Mem0. Mem0 centers on fact extraction and consistency-preserving updates. For each new message pair, it uses a conversation summary and recent messages as context to extract candidate memories. It then retrieves semantically similar stored memories and lets an LLM select `ADD`, `UPDATE`, `DELETE`, or `NOOP` to incorporate new information, reduce redundancy, and reconcile changes (Chhikara et al., 2025).

SimpleMem. SimpleMem combines semantic compression with online consolidation. It processes overlapping input windows, uses semantic-density gating to filter low-value content, and jointly performs fact extraction, coreference resolution, and temporal normalization to produce self-contained memory units. During writing, related facts within the current session are synthesized into unified entries. Retrieval combines dense semantic embeddings, lexical inverted indexes, and symbolic metadata such as timestamps, with its scope adapted to the inferred query intent (Liu et al., 2026b).

LightMem. LightMem combines topic-aware organization with decoupled online and offline maintenance. Lightweight compression first filters redundant input, and attention and semantic similarity between adjacent turns identify topic segments. Short-term memory organizes and summarizes these segments into entries. Long-term memory first inserts new entries with timestamps and subsequently updates them offline (Fang et al., 2026).

D.3 Relational Memory

A-Mem. A-Mem constructs interlinked, evolving atomic notes containing original content, timestamps, keywords, tags, contextual descriptions, embeddings, and links. For each new note, it retrieves semantically similar historical notes and uses an LLM to determine meaningful connections. The retrieved notes may then have their context, keywords, and tags revised in light of the new information (Xu et al., 2026b).

StructMem. StructMem combines event-level binding with cross-event synthesis. For each utterance, it extracts factual entries describing event content and relational entries capturing interpersonal dynamics, causal influences, and temporal dependencies. Both perspectives are bound by their originating timestamp. Periodically, the system uses buffered, unconsolidated entries to retrieve semantically related historical seeds. It reconstructs each seed event by gathering entries with the same

Table 9: Comparison of documented memory-system designs. **OS**: OS-specific cross-application ingestion. **MM**: non-text input processing. **EvU**: event-based memory units. **LF**: generative-LLM-free initial organization. **Link**: persistent relational organization among memory units. **Rev**: post-construction structural revision. **MRS**: multi-record synthesis during memory construction. **LTM**: cross-event induction of recurring patterns, preferences, or routines. **Prov**: access to source content or source references. Orange marks limited support; crosses denote mechanisms not explicitly specified in the cited design. OS and MM refer to the original input pipelines, not capabilities provided by shared PMM-Bench adapters.

	Scope		Organization			Refinement			Trace
Method	OS	MM	EvU	LF	Link	Rev	MRS	LTM	Prov
Reference baselines									
Long context	✗	✗	✗	✓	✗	✗	✗	✗	✓
Flat dense retrieval	✗	✗	✗	✓	✗	✗	✗	✗	✓
Compact-entry memory									
Mem0 (Chhikara et al., 2025)	✗	✗	✗	✗	✗	✓	✓	✓	✗
SimpleMem (Liu et al., 2026b)	✗	✗	✗	✓	✗	✗	✓	✓	✗
LightMem (Fang et al., 2026)	✗	✗	✓	✓	✗	✓	✓	✓	✓
Relational memory									
A-Mem (Xu et al., 2026b)	✗	✗	✗	✗	✓	✓	✓	✓	✓
StructMem (Xu et al., 2026a)	✗	✗	✓	✗	✓	✗	✓	✓	✓
CompassMem (Hu et al., 2026a)	✗	✗	✓	✓	✓	✓	✓	✗	✓
HyperOS-Mem	✓	✓	✓	✓	✓	✓	✓	✓	✓

timestamp, then synthesizes cross-event relations from the reconstructed and buffered context (Xu et al., 2026a).

CompassMem. CompassMem incrementally constructs a hierarchical event graph through event segmentation, relation extraction, event fusion, and topic organization. Each event retains its source observations, temporal information, summary, and participants. An open-ended predicate set represents logical relations, including causal, temporal, motivational, and part-of relations. At query time, a Planner decomposes the question into subgoals, while multiple Explorers navigate the graph to collect evidence. Queries are refined when the collected evidence is insufficient (Hu et al., 2026a).

D.4 Input Adaptation

We represent each user's observation history within the specified period as a chronologically ordered sequence of records. Each record contains the observation text, timestamp, application name, user identifier, and observation identifier, with source-file line references retained as metadata.

System-specific adapters convert these records into the native input formats. SimpleMem receives timestamped dialogue entries through its batch interface; Mem0 receives individual user messages with observation identifiers and timestamps retained in metadata; and A-Mem receives timestamped text notes. LightMem and StructMem receive each observation as a user message paired with an empty assistant message, satisfying their paired-turn buffers while ensuring that all actual observations are included in user-only extraction. For CompassMem, which requires session-based input, we partition the stream when the gap between consecutive observations exceeds 30 minutes or adding the next observation would exceed an approximately 4,000-token session limit. Session boundaries fall between complete observations, without overlap or omission. Flat dense retrieval indexes one record per observation, while Long context sequentially inserts complete observations into a 128,000-token FIFO window.

D.5 Adaptation Protocol

The memory systems are integrated through their native memory construction and retrieval interfaces. Flat dense retrieval embeds each observation and retrieves relevant records by computing cosine similarity between the question and all observations in the corpus; Long context directly supplies the complete observations retained in its FIFO window to the answer model. SimpleMem constructs memories from dialogue entries and combines hybrid retrieval, query planning, and reflection to obtain relevant information; LightMem processes observations through pre-compression, topic segmentation, memory extraction, and offline updating; and Mem0 extracts facts from messages

and updates its memories. CompassMem first extracts events and relations from observation sessions, then performs cross-session merging and hierarchy construction, followed by question-driven multi-path search. The A-Mem integration includes note analysis, memory linking and evolution, and agentic retrieval; StructMem constructs event memories and cross-event synthesis memories. Table 10 lists the input format, configuration parameters, and mode of each baseline.

Table 10: Input formats, configuration parameters, and modes of the baselines.

Baseline	Implementation	Input format	Configuration parameters and modes
Long context (128K)	Official reference implementation	Chronologically ordered observation text	FIFO mode; context window: 128,000 tokens
Flat dense retrieval	In-house implementation	One record per observation	Exact search; cosine similarity; Top-30
SimpleMem	Official implementation	Timestamped dialogue entries in batches	Batch ingestion; hybrid retrieval, query planning, and reflection enabled; at most two reflection rounds
LightMem	Official implementation	User message paired with an empty assistant message	Compression retention rate: 0.8; pre-compression, topic segmentation, and offline updating enabled; update similarity threshold: 0.9
Mem0	Official implementation	User message with source metadata	Fact extraction and memory updating enabled
A-Mem	Official implementation	Timestamped observation notes	Memory evolution and agentic retrieval enabled; memory consolidation triggered every 100 evolution events
StructMem	Official implementation	User message paired with an empty assistant message	Event extraction, cross-event synthesis, and offline updating enabled; compression retention rate: 0.6; synthesis window: one hour; global retrieval of 15 seed memories
CompassMem	Official implementation	Chronologically grouped observation sessions	Three search paths; at most three rounds; navigation similarity threshold: 0.7; relation reasoning, subgoal planning, and query refinement enabled; early stopping disabled

E Models and Evaluation Details

Models. Table 11 summarizes the models used in the main-table evaluation. The evaluated baselines share the same model for each applicable stage. For LLM-based memory construction and retrieval in the adapted baselines, we disable thinking and retain each system's native stage-specific sampling parameters.

Table 11: Models and settings used in the main-table evaluation.

Stage	Model	Role and settings
Memory construction	Qwen3.8-27B	LLM-based memory extraction, updating, and retrieval; thinking disabled
Embedding	GME-Qwen2-VL-7B	Text encoding; 3,584 dimensions; L2 normalization
Answer generation	GPT-4.1-mini	Generates answers from the supplied context; temperature 0; maximum 2,048 output tokens
Answer scoring	GPT-4o	Scores answers using question-type-specific rubrics; temperature 0; maximum 8,192 output tokens

Answer generation. For retrieval-based systems, we retain at most 30 returned memory records and construct the answer context from a prefix of complete records in their returned order, with a limit of 48,000 characters in the serialized context. Long context uses a 128,000-token FIFO window of complete observations, removing the oldest observations as needed to accommodate the question prompt and output budget. We use a distinct answer-generation prompt for each question type. For a given question type, the prompt is identical across all evaluated systems.

Table 12: Component analysis of HyperOS-Mem on PMM-Bench. Single: single-hop recall; Cross: aggregation across applications; Temp: temporal reasoning; Track: event tracking; Habit: habit induction. Judge is the unweighted mean of the five per-type scores.

Variant	Question type					Overall
	Single	*Cross*	*Temp*	*Track*	*Habit*	**Judge**
DRED						
w/o identifier-first attribution	46.9	35.4	38.7	37.2	38.5	39.3
w/o deferred attribution	46.6	39.7	34.9	34.6	38.2	38.8
w/o dual-rate prototype update	47.6	40.8	39.4	39.1	39.3	41.2
SAGE						
w/o fact extraction	43.8	38.2	37.4	40.1	36.9	39.3
w/o long-term memory	47.9	42.6	40.2	39.4	31.8	40.4
HyperOS-Mem	**48.2**	**43.5**	**41.1**	**42.8**	**40.6**	**43.2**

Answer scoring. GPT-4o evaluates each generated answer against the question and reference answers using a shared rubric for its question type. The rubric checks the requested facts for Single, the required items and attributes for Cross, temporal order and continuation for Temp, event progression and the latest supported state for Track, and recurring personal procedures and their boundaries for Habit. Semantically equivalent expressions are accepted. The judge assigns an integer score from 0 to 5. Scores of 4 or 5 count as correct and require all core points to be accurate and complete, allowing only nonessential omissions or minor ambiguity. Scores of 0–3 count as incorrect; major factual or temporal errors receive at most 2. For questions marked unanswerable by the reference, explicit abstention without invented facts counts as correct; abstention on an answerable question counts as incorrect.

F Additional Results

F.1 Component analysis by question type.

Table 12 lists the numbers plotted in Figure 5. The upper block varies the rules of DRED and the lower block the parts SAGE adds.

F.2 Build and Query Overhead.

As illustrated in Figure 15, the Long Context approach incurs no additional memory construction overhead, yielding a build cost of zero; however, it necessitates feeding the extensive context into the model during the query phase, which results in a substantially high query delay. Flat Dense Retrieval performs only flat retrieval without constructing higher-order memory representations; consequently, while it is computationally inexpensive to build and fast to query, its accuracy remains inherently limited. Mem0 performs explicit memory extraction and continuous updating. In contrast, A-Mem, StructMem, and CompassMem involve more complex memory organization, linking mechanisms, or event structures, thereby incurring a relatively high overall build cost. Although CompassMem delivers strong performance, it relies on a complex event graph and computationally intensive query-time reasoning. Ultimately, HyperOS-Mem offers a more favorable accuracy-efficiency trade-off, achieving significantly higher accuracy compared to lightweight baseline systems while maintaining computational efficiency.

F.3 Additional benchmark comparisons.

We supplement the main experimental comparisons with results on Mem-Gallery (Table 13) and MemBench (Table 14), together with an additional PMM-Bench comparison using GPT-5-mini as the response model (Table 15). All numerical scores are reported as percentages.

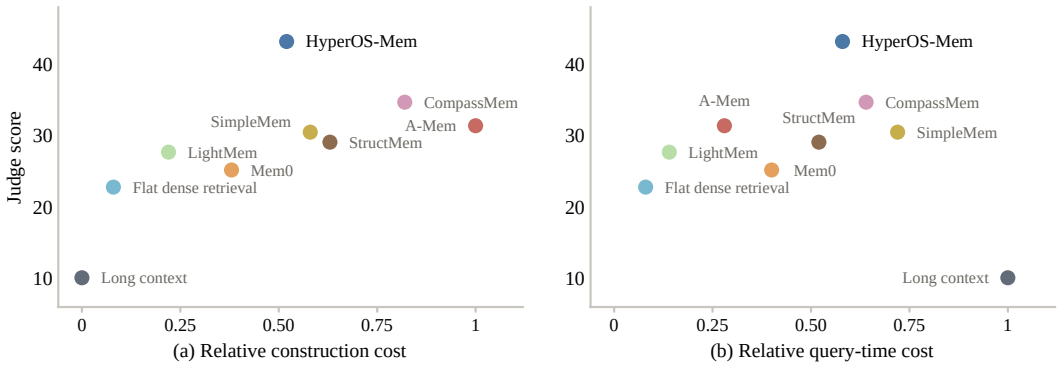


Figure 15: Cost of memory construction and of question answering. LLM tokens spent at each stage, against accuracy.

Table 13: Results on Mem-Gallery. Judge is the primary answer-quality metric: the mean of per-question ratings $(0, 0.25, 0.5, 0.75, 1)$, multiplied by 100. F1 measures token overlap after text normalization and Porter stemming, taking the best reference match per question and then averaging across questions. EM is the percentage of questions for which the model answer and at least one reference become identical after applying the same text-processing rules to both. The response model is GPT-4o-mini and the judge model is Qwen2.5-72B-Instruct-GPTQ-Int4.

Method	**Judge ↑**	**F1 ↑**	**EM ↑**
References			
Long context (128K)	57.63	65.03	**44.01**
Flat dense retrieval	56.22	63.96	40.68
Compact-entry memory			
SimpleMem	51.20	59.30	35.94
LightMem	33.14	39.58	25.89
Mem0	34.72	40.87	25.60
Relational memory			
A-Mem	53.71	62.35	39.10
StructMem	34.77	42.37	26.53
CompassMem	56.17	63.05	41.32
HyperOS-Mem	**58.11**	**66.28**	41.85

G Retrieval Details

Retrieval procedure. The initial pass retains at most B evidence nodes. At each navigation step, the controller receives the retained evidence and at most C available action–target pairs. `FOLLOW_ANCHOR` retrieves adjacent events; `OPEN_EVENT` exposes relevant facts and source observations; `OPEN_MEMORY` exposes the memory's supporting events. Each expansion is followed by candidate merging, deduplication, and pruning to B retained nodes. Each action–target pair is executed at most once. The controller stops when the retained evidence is sufficient; the outer procedure also stops when no unvisited action remains or T actions have been executed. These resource limits do not imply evidence sufficiency. The budgets are listed in Table 16; T excludes initial retrieval and answer generation.

Provenance and output. Facts retain their observation supports Π_f, events retain M_E, and long-term memories retain their supporting events Ω_m. These provenance links trace returned evidence to observations; navigation paths are stored separately. Routing anchors and provenance links do not count toward B. The final answer context follows the shared evaluation limit of 30 complete records and 48,000 serialized characters.

Prompt 4 gives the controller template.

Table 14: Accuracy on MemBench. Participatory and observational memory are each evaluated with factual and reflective questions. Accuracy is the fraction of questions answered with the correct option. Overall is the average of the four category accuracies weighted by their question counts, covering 23,166 questions in total. The response model is GPT-4o-mini; no LLM judge is used.

Method	Overall ↑	Participatory		Observational	
		Factual ↑	*Reflective* ↑	*Factual* ↑	*Reflective* ↑
References					
Long context (128K)	42.23	49.70	76.30	22.91	**86.20**
Flat dense retrieval	51.21	**58.15**	**95.60**	33.84	76.05
Compact-entry memory					
SimpleMem	30.02	29.48	87.95	11.44	78.05
LightMem	47.97	30.65	87.15	48.01	77.85
Mem0	39.06	15.34	88.20	39.17	84.20
Relational memory					
A-Mem	55.06	49.60	94.35	47.78	78.30
StructMem	35.03	39.62	87.40	14.65	78.10
CompassMem	34.75	40.54	78.35	14.96	78.55
HyperOS-Mem	**56.60**	47.90	90.30	**51.60**	85.60

Table 15: Results on PMM-Bench with an alternative response model. Single: single-hop recall; Cross: aggregation across applications; Temp: temporal reasoning; Track: event tracking; Habit: habit induction. The five category columns and Judge report category-level and overall accuracy, respectively, with ratings of 4 or 5 on a 0–5 scale counted as correct. F1 measures answer overlap using Chinese characters and English word/number tokens, taking the best reference match per question and then averaging across questions. The response model is GPT-5-mini and the judge model is GPT-4o.

Method	Question type					Overall	
	Single	*Cross*	*Temp*	*Track*	*Habit*	**Judge** ↑	**F1** ↑
References							
Long context (128K)	23.08	4.00	7.82	5.56	11.30	10.35	9.12
Flat dense retrieval	**61.54**	10.00	18.75	33.33	32.50	31.22	22.10
Compact-entry memory							
SimpleMem	30.77	21.62	29.80	20.67	32.31	27.03	28.50
LightMem	31.23	23.12	24.30	22.10	33.63	26.88	23.80
Mem0	23.08	15.00	25.10	20.00	37.50	24.14	21.80
Relational memory							
A-Mem	42.10	27.80	28.20	25.70	29.70	30.70	28.60
StructMem	38.20	25.10	29.30	26.30	28.10	29.40	26.20
CompassMem	23.08	29.80	**35.10**	16.67	34.27	27.78	28.50
HyperOS-Mem	53.85	**35.00**	31.25	**38.89**	**37.50**	**39.30**	**36.90**

Prompt 4: Controller for navigation over the hierarchical event graph.

```
Collect evidence for a question from a personal memory graph.

Facts link to source observations within events. Events group
observations of an activity across applications. Long-term memories
link to supporting events. Shared entity or topic anchors connect
events.

Question: {question}
Uncovered query keys (possibly empty): {uncovered_keys}
```

```
Retained evidence: {evidence_digest}
# Each node: id | type | time | content | support IDs

Available actions: {candidates}
FOLLOW_ANCHOR | anchor_id | name | degree | events reached
OPEN_EVENT    | event_id  | interval | apps | facts matched
OPEN_MEMORY   | memory_id | summary | supporting events

Stop when the retained evidence supports an answer, including the
required temporal relations and contextual constraints. Key
coverage alone is insufficient. If evidence is missing, choose
the listed action most likely to supply it. Among equally relevant
anchors, prefer lower degree. Use only listed action--ID pairs.
Do not infer missing facts.

Return STRICT JSON with:
- "stop": boolean
- "action": "FOLLOW_ANCHOR", "OPEN_EVENT", "OPEN_MEMORY", or null
- "id": a listed integer ID, or null
- "reason": one short clause

When stopping, set both "action" and "id" to null.
```

H HYPERPARAMETERS

Table 16 lists the construction and retrieval constants used in the reported experiments. Attribution similarities are cosines in the centered space of Section 3.1, which is why the attribution thresholds are small in absolute terms. The required orderings hold in the implementation: $\tau_p > \tau_a$ and $\alpha_{\text{chat}} < \alpha_{\text{def}}$. The split indicator of Section 3.3 fires when $\text{coh}(E) < \theta_c$ and $|M_E| \geq n_{\min}$. Construction settings are shared across users. Retrieval uses fixed budgets of $B = 30$ retained evidence nodes, $C = 5$ candidate action–target pairs per navigation step, and $T = 3$ navigation actions after initial retrieval. The same settings are used for every user.

Table 16: Construction and retrieval constants of HyperOS-Mem, with one setting for all users.

Symbol	Name	Value	Role
DRED			
α_{def}	ALPHA	0.30	prototype update rate outside messaging apps
α_{chat}	ALPHA_CHAT	0.12	prototype update rate inside messaging apps
τ_a	SPLIT_COS	0.22	threshold for attachment to an active event
τ_p	PROMOTE_COS	0.25	threshold for promoting two pending observations
Δ_a	ACTIVE_WINDOW	10 min	active duration after an event's last observation
Δ_p	PEND_WIN	3 min	waiting time for confirming evidence
Δ_s	SESSION_CAP	2 h	maximum separation for merchant-key merging
Slow path (SAGE)			
θ_c	COH_SPLIT	0.55	cohesion threshold for the split indicator
$n_{\min}$	BIG	40	minimum event size considered for splitting
τ_0	MINBUCKET	0.05	similarity below which a new cluster is opened
τ_s	SUBCOS	0.45	threshold for matching an existing topic anchor
Retrieval			
B	EVIDENCE_CAP	30	maximum retained evidence nodes
C	ACTION_CAP	5	maximum candidate actions shown per step
T	NAV_STEPS	3	maximum actions after initial retrieval

I CASE STUDY

This appendix presents a representative case study of user U01, an undergraduate student with approximately three months of daily Android usage. In HyperOS-Mem, DRED assigns atomic observations to evolving events, while SAGE further refines event structure, extracts supported facts, and consolidates long-term memory. The following cases illustrate how these memory representations support different forms of personalized reasoning. Figure 16 demonstrates single-source recall through a hotel-booking event. Figure 17 shows cross-application and long-term aggregation for personalized food recommendation. Figure 18 illustrates multi-stage event tracking across thesis-related activities, supporting identification of the current thesis stage and a grounded next-step recommendation.

CASE 1: SINGLE-SOURCE RECALL

Query

Which hotel did I stay at in Shenzhen last June?

Evidence Scope

Based on atomic observations related to accommodation collected during June, including screenshots and received notifications.

Atomic Observations

Hotel-booking trajectory



Trace Details

Evidence	Date	Time	App / Source	Caption / observation content
Atomic (a)	06-16	09:58	Meituan	Browsed and considered booking Shenzhen Longgang Central District Sando Space Hotel.
Atomic (b)	06-17	01:03	Meituan	Selected a room on the hotel-booking platform and prepared to place the order.
Atomic (c)	06-17	01:18	Meituan	Successfully booked a deluxe twin room near Shenzhen Science Park Metro Station.

Booking Signals

Location
Shenzhen Science Park Metro Station
Mixc World area

Hotel
7 Days Inn

Stay
Jun. 20–21
1 night

Booking Platform
Meituan

Recommendation Path

Filter by June → Locate accommodation-related events → Trace hotel booking → Retrieve hotel name

The retrieval process narrows the search to June, identifies accommodation-related events, and follows the hotel-booking evidence to recover the hotel name.

Answer

7 Days Inn — Shenzhen Science Park Metro Station, Mixc World branch.

Figure 16: Case 1: Single-source recall supported by a continuous hotel-booking trajectory.

CASE 2: CROSS-APP WORKFLOW

Query

Please help me order a food delivery.

Evidence Scope

Based on atomic observation data related to food delivery collected over the past month, including screenshots and received push notifications (e.g., SMS and WeChat). For ease of presentation and understanding, only a representative subset of the atomic observations is shown.

Atomic Observations

Continuous event (a)–(f): one coherent ordering trajectory



Representative screenshots from long-term traces (g)–(m): selected non-contiguous observations



Trace Details

Evidence	Date	Time	App / Source	Caption / observation content
Atomic (a)	2026-04-30	18:07:14	Qianwen	Selected a delivery order for braised-noodle soup in Qianwen.
Atomic (b)	2026-04-30	18:07:25	App switch	Switched from the Qianwen assistant to Taobao.
Atomic (c)	2026-04-30	18:08:14	Taobao Flash Sale	Viewed the delivery listing for handmade northeastern dumplings (Fengyuwan Store).
Atomic (d)	2026-04-30	18:08:41	Taobao Flash Sale	Prepared to check out.
Atomic (e)	2026-04-30	19:12:47	Qianwen	Placed an order through Qianwen for mushroom-and-pork braised-noodle soup.
Atomic (f)	2026-04-30	19:17:57	Taobao Flash Sale	Checked the Taobao Flash Sale order; estimated delivery was 19:35–19:50.
Atomic (g)	2026-05-01	13:59:18	Dianping	Searched Dianping for Qiba Lengmian Korean Rice Wine House (Dayun Store).
Atomic (h)	2026-05-05	10:44:46	Meituan	Browsed the menu of a Cantonese–Hong Kong-style restaurant on Meituan Waimai and considered ordering.
Atomic (i)	2026-05-09	19:10:19	App switch	Switched from Meituan to Meituan FanFan to view coupons.
Atomic (j)	2026-05-11	09:50:37	WeChat	Ordered a beverage through a food-delivery platform.
Atomic (k)	2026-05-18	18:26:28	WeChat	Claimed a Meituan Waimai coupon through WeChat Pay.
Atomic (l)	2026-05-26	10:38:33	Alipay	Viewed coupons in Alipay while preparing to order.
Atomic (m)	2026-06-02	19:20:34	Douyin	Viewed the fresh-green-pepper mala grilled-fish set from Tanyu Restaurant on Douyin Group Buying.
Other data	–	–	Push notifications	Text notifications received from WeChat, Alipay, SMS, Meituan, Taobao Flash Sale, and other apps about food discussions, successful orders, delivery in progress, and pickup.

Preference Signals

- **Platform frequency**
  - Meituan (12)
  - WeChat (5)
  - Taobao Flash Sale (4)
- **Meal categories**
  - Coffee/drinks (7)
  - Noodles (3)
  - Rice meals (2)
- **Browsing interests**
  - Hong Kong-style BBQ (3)
  - Da-lu noodles (2)
- **Chat topics**
  - Da-lu noodles
  - Instant noodles
  - Tea eggs

Recommendation Path

Monthly order → Platform & meal preference → Dietary preference memory → Personalized recommendation

Unlike a simple replay of the latest order, the recommendation combines a coherent recent event with recurring signals found across non-contiguous history.

Answer

Based on your dietary record for the past month, I suggest that you use the group discount coupons / special vouchers / flash purchase red envelopes on Meituan and order a bowl of danhu noodles.

Figure 17: Case 2: Cross-app workflow for food-delivery ordering and personalized recommendation from inter-app evidence.

CASE 3: MULTI-STAGE EVENT TRACKING

Query

Please help me review my thesis progress and determine what I should do next.

Evidence Scope

Based on thesis-related atomic observations collected from late May to mid-June, including screenshots, chat messages, document browsing records, and system logs. For ease of presentation and understanding, only a small subset of representative atomic observations is shown.

Atomic Observations

Recent continuous progress sequence (a)–(f): one coherent thesis-finalization trajectory



Representative non-contiguous thesis records (g)–(l): selected observations from the longer history



Trace Details

Evidence	Date	Time	App / Source	Caption / observation content
Atomic (a)	2026-06-04	15:17:54	WeChat	Viewed the Yinmeng printing-order details, including the assignment sheet, general grading form, and undergraduate thesis materials.
Atomic (b)	2026-06-04	15:17:57	WeChat	Viewed the Yinmeng printing-order details, including the undergraduate thesis and student thesis materials.
Atomic (c)	2026-06-04	15:18:18	WeChat	Opened the printing details for the Guangzhou Business College assignment sheet in the Yinmeng order.
Atomic (d)	2026-06-04	15:24:15	WeChat	Used a floating WeChat window to view the Yinmeng printing order, including the general grading form and undergraduate thesis.
Atomic (e)	2026-06-04	15:24:18	Memo	Viewed the printing details in Yinmeng My Orders and opened Memo.
Atomic (f)	2026-06-04	15:24:57	Memo	Viewed the printing details in Yinmeng My Orders and recorded the price in Memo.
Atomic (g)	2026-05-28	08:41:02	WeChat	Viewed document attachments in a chat with xxyy, including the originality statement, authorization form, and thesis materials.
Atomic (h)	2026-05-28	08:41:22	WeChat	Previewed the 2026 Guangzhou Business College undergraduate thesis (design).
Atomic (i)	2026-05-31	21:09:15	Notes	Edited a research document on cross-border mergers and acquisitions.
Atomic (j)	2026-06-05	18:09:00	WeChat	Searched group-chat records about thesis printing and submission of thesis attachments.
Atomic (k)	2026-06-10	10:15:32	WeChat	Viewed a plagiarism-check report.
Atomic (l)	2026-06-13	17:29:44	WeChat	Sent a document about investor irrationality under the Guizhou Moutai alcohol-ban policy, noted a 6.3% similarity rate, and asked the teacher about printing requirements.

Progress Signals

- **Thesis status**
  - Late-stage finalization
  - Final draft
- **Verification**
  - Plagiarism checks
  - Printing requirement check
- **Thesis materials**
  - Thesis proposal & progress record
  - Defense record
- **Coordination**
  - Classmate communication
  - Teacher communication

Next-step Path

Historical thesis record → Stage identification → Pending tasks → Next-step prediction

The response combines the continuous printing-related records with non-contiguous thesis records, linking thesis writing, plagiarism checking, document review, and printing preparation to reconstruct the current progress and determine the next steps.

Answer

Your thesis has entered the late-stage finalization phase. Based on the records, you handled the progress schedule, thesis materials, final-format checks, and two plagiarism checks from late May to mid-June, while also discussing final-version uploads, file versions, and finalization with classmates and teachers. Next, comprehensively check the thesis body, cover, and attachments, confirm the final file version, and complete the formal submission. The available records do not include an explicit successful-submission receipt, so it is not yet possible to confirm that the thesis has been successfully submitted and approved.

Figure 18: Multi-stage event tracking across a continuous thesis-finalization sequence and representative non-contiguous records.

Case 1 | Insurance Claim Logs

Query	*What progress has Zheng Yujia's claim in Haofuli made?*
Mem0	**Factual Entries** May 19: Application submitted. May 26: Zheng Yongxiang, CNY 508.60, closed; repeated notification observations. [Distractor] Data-collection service inactive; heartbeat 09:25:40.
A-MEM	**Memory Notes + Links** "Claim details" ↔ "Status recheck": …248983; patient Zheng Yujia, recipient Zheng Yongxiang; Submitted on May 19 and May 20 morning. [Distractor] Claims portal and FAQs.
LightMem	**Topic Summaries** May claims summary: Application submitted; later notifications show a CNY 508.60 closure. [Distractor] Pension and shopping promotions.
SimpleMem	**Structured Facts** May 20 | Zheng Yujia | application …647405 | Submitted. May 26 | Zheng Yongxiang, Wu Heping | multiple closed claims.
CompassMem	**Event Graph + Active Search** E1: Zheng Yujia, claim …248983, Submitted (May 19) → follow_up → E2: Initial Document Review (May 20, 16:24). E3: May 26 closure list; no claim-ID match to E1–E2.
HyperOS-Mem	**Event Memory** May 19 | Haofuli: Claim …248983, Submitted; patient Zheng Yujia, recipient Zheng Yongxiang. May 20 | Status checks: Submitted at 09:26 → Initial Document Review at 16:24; same claim ID. May 26 | Haofuli: Multiple closures and repeated CNY 508.60 notice; no target claim-ID match.
Reference	Claim …248983: Submitted on May 19 → Initial Document Review on May 20. The May 26 closure list does not confirm closure of this claim.
Illustrative Answers	
Mem0	Submission and a closure notice found; target claim's latest status unclear.
A-MEM	Submitted in the retrieved May 20 morning record; no later update retrieved.
LightMem	The application progressed to closure, with CNY 508.60 compensation.
SimpleMem	Submitted in the retrieved record; later closures cannot be matched to it.
CompassMem	Submitted → Initial Document Review; closure of this claim unconfirmed.
HyperOS-Mem	Submitted → Initial Document Review by May 20, 16:24; same recipient does not prove closure.

Figure 19: Tracking claim-specific state changes through event context.

Comparison Cases

As shown in Figure 19, HyperOS-Mem organizes the claim history into event memories containing timestamps, claim identifiers, and participant roles, preserving the transition from Submitted to Initial Document Review. Combining the claim identifier with patient and payment-recipient information supports identification of the latest documented state while keeping unmatched closure notices separate. This illustrates the value of event memory for tracking state changes, preserving identity context, and constraining conclusions, allowing the answer to distinguish confirmed progress from matters requiring further verification.

As shown in Figure 20, HyperOS-Mem organizes records from conversations, e-signing, a pharmacy application, and notes into events, providing a stage-based view of examination coordination, document submission, signing, and subsequent activity. Each event retains its time and source, enabling a comprehensive progress review while distinguishing direct onboarding evidence from potentially related later observations. This illustrates its value for cross-application evidence organization, stage coverage, and preservation of attribution boundaries, preventing document submission, completed signing, or pharmacy activity from being overstated as completion of all onboarding requirements.

Case 2 | Onboarding Logs

Query	*How far has my Haoyaoshi onboarding progressed in the records?*
Mem0	**Factual Entries** Sent qualification PDF and bank-card photo to Haoyaoshi HR; completed e-signing. [Distractor] Other jobs: CNY 3,000 probation salary; part-time cook interview.
A-MEM	**Memory Notes + Links** "Examination wait" ↔ "Package inquiry": June 24, waiting to be called; results at 4 p.m. "Documents sent" → same HR contact, June 29. [Distractor] Railway training and inventory files in WeChat Favorites.
LightMem	**Topic Summaries** Onboarding summary: Discussed the examination wait, received a checklist, and sent qualification and bank-card documents. [Distractor] Track inspection: work authorization 02:11; clearance 03:50.
SimpleMem	**Structured Facts** June 24 | examination coordination; June 29 | documents sent. July 2 | Shangshangqian e-signing | "Done, I've signed."
CompassMem	**Event Graph + Active Search** E1: Examination → follow_up → E2: Documents → follow_up → E3: E-signing; June 24–July 2. E4: July 3 pharmacy operations. E5: Separate "Haoyaoshi Orders" note. Their connection is unconfirmed.
HyperOS-Mem	**Event Memory** June 24 | WeChat: HR discussed the examination wait. June 29 | WeChat: Sent qualification PDF and bank-card photo. July 2 | E-signing: "Done, I've signed"; HR acknowledged. July 3, 06:15 | Pharmacy app: Batch entry and quantity sold: 1; employer unspecified. July 3, 07:13 | Notes: "Haoyaoshi Orders", "July 2 47"; link to pharmacy activity unconfirmed.
Reference	Examination discussed → documents sent → e-signing completed. Pharmacy operations appear on July 3; employer attribution and remaining requirements need confirmation.
Illustrative Answers	
Mem0	Documents sent and e-signing completed; no store operations retrieved.
A-MEM	Examination discussed and documents sent; later stages not retrieved.
LightMem	Examination coordination and document submission; later progress unclear.
SimpleMem	Examination → documents → e-signing; no later work operations retrieved.
CompassMem	Three HR-related stages documented; pharmacy–note link and remaining requirements unconfirmed.
HyperOS-Mem	Three HR-related stages documented, followed by separate pharmacy and note events; verify attribution and remaining requirements.

Figure 20: Cross-application evidence integration for onboarding progress tracking.