SCREENMEM: BENCHMARKING PROACTIVE MEMORY MAINTENANCE OVER LONG-HORIZON COMPUTER-USE STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory is a fundamental capability of computer-use agents, enabling them to retain and reuse information accumulated throughout everyday computer use. Yet existing evaluations do not fully characterize a key setting for persistent memory: continuously observing long-horizon device activity, deciding what to preserve before future information needs are known, and later answering from the resulting memory without revisiting the original stream. We introduce ScreenMem, a benchmark for proactive, query-agnostic memory maintenance over day-long computer-use streams. ScreenMem reconstructs application interfaces and replays interactions from open-source real-world traces, producing 10 user-day streams across PC and mobile devices totaling 100.1 hours, together with 484 human-verified questions covering facts and states, state changes, activities and events, and behavioral patterns. Under a unified causal-ingestion protocol, we evaluate 15 existing memory systems spanning recent-context, implicit visual, explicit multimodal, and generic text-memory paradigms on both PC and mobile streams. Across 15 systems, the strongest evaluated system reaches only 31.4% overall accuracy, while State Evolution accuracy peaks at 15.8%. Further analyses reveal substantial variation across temporal horizons, together with pronounced differences in ingestion and storage efficiency. ScreenMem can facilitate systematic analysis of current memory systems and further research on persistent memory maintenance for computer-use agents.

1 INTRODUCTION

Computer-use agents are increasingly capable of operating computers and mobile devices to execute complex tasks across applications and interfaces (Xie et al., 2024; Rawles et al., 2025). Long-term assistance, however, requires more than completing isolated tasks: an assistant must also remember information encountered during everyday computer use, evolving device states, transient events, and recurring behaviors. Much of this context arises while users interact directly with their devices rather than during explicit agent interactions, so memory accumulated only from conversations or task executions can omit information needed for future assistance.

Maintaining such memory is fundamentally different from retrieving history after a request is known. During ongoing computer use, the system does not know which observations will matter later, while retaining and repeatedly processing the full history are costly and often impractical. It must therefore decide online what to preserve, how to update information whose validity changes, and when repeated activities should be consolidated into longer-term patterns. We refer to this setting as *proactive, query-agnostic memory maintenance*: constructing and updating persistent memory before downstream information needs are known, where *proactive* refers to memory maintenance rather than intervention in user activities. The central question is thus *what should be remembered before the system knows what it will later be asked.*

Existing benchmarks study complementary aspects of memory for computer-use agents, including cross-session task execution, personalization from historical records, and memory over screen or mobile interaction histories (Liu et al., 2026a; Lyu et al., 2026; Yang et al., 2025a; Wang & Chen, 2025; Deng et al., 2026). However, they do not jointly evaluate causal memory formation over long-

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

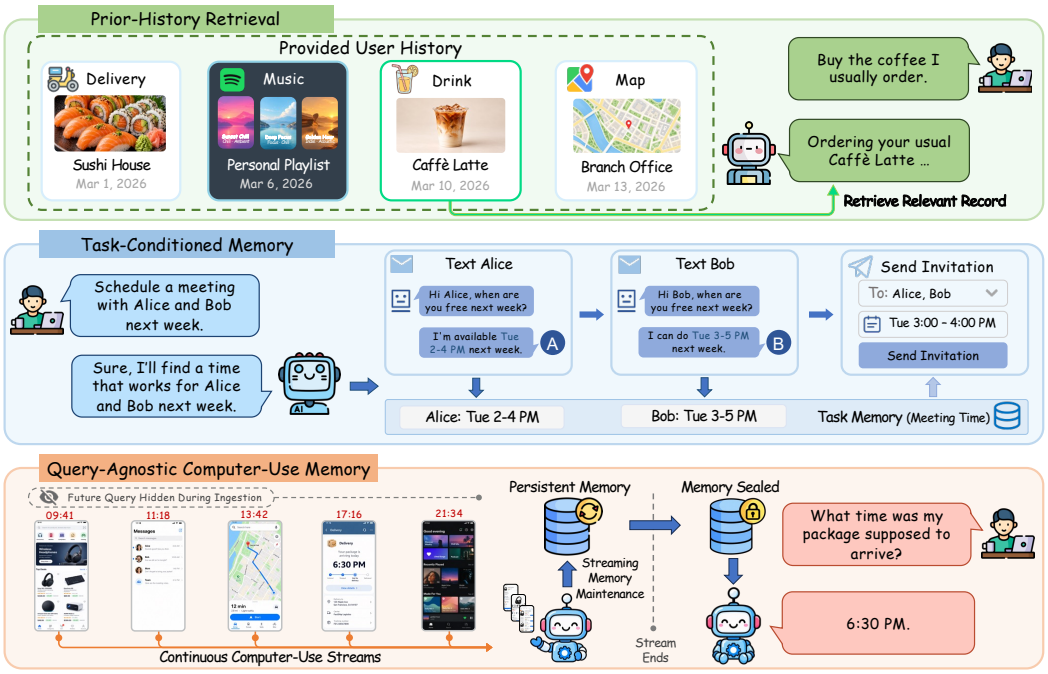


Figure 1: Comparison of memory formulations for computer-use agents. Prior-history retrieval selects from an available history after a request is known; task-conditioned memory supports an explicit task; ScreenMem evaluates query-agnostic memory maintenance over continuous computer-use streams.

horizon computer-use streams when future evaluation queries are unavailable during ingestion. This shifts the focus from retrieving a provided history or supporting a known task to evaluating what a system chooses to preserve before future information needs are known. Table 1 summarizes this distinction across representative benchmarks.

To study this setting, we introduce **ScreenMem**, comprising 10 reconstructed PC and mobile user-day streams derived from open-source real-world traces (Bakhshizadeh et al., 2024; Yang et al., 2025a), totaling 100.1 hours, together with 484 human-verified questions across four memory capabilities: facts and states, state evolution, activities and transient events, and behavioral patterns. Systems ingest each user-day stream causally while evaluation queries remain hidden, then answer from the resulting sealed memory without revisiting the original stream. Beyond answer accuracy, we measure ingestion time, token consumption, and persistent storage to characterize the cost of continuous memory maintenance.

We benchmark 15 existing memory systems spanning recent-context, implicit visual, explicit multimodal, and generic text-memory paradigms. Across all 15 systems, the strongest method reaches only 31.4% overall accuracy, with 29.2% on PC and 35.5% on mobile. State evolution remains particularly difficult, with the strongest system reaching only 15.8%, while systems also exhibit substantial variation in ingestion and storage efficiency. Together, these results show that current memory systems remain far from reliably maintaining the information needed across day-long computer-use streams and motivate analysis beyond a single aggregate leaderboard score.

This paper makes the following contributions:

- We formulate *proactive, query-agnostic memory maintenance* over long-horizon computer-use streams, separating causal memory formation from future query-conditioned retrieval and task execution.
- We introduce ScreenMem, with 10 PC and mobile user-day streams totaling 100.1 hours and 484 human-verified questions covering four complementary memory capabilities under a sealed-memory evaluation protocol.

Table 1: Comparison of representative memory benchmarks for computer-use agents. UI Stream denotes memory formed from temporally ordered UI observations, and Query-Agnostic denotes that evaluation queries are hidden during memory formation.

Benchmark	Device	Input Modality	UI Stream	Query Agnostic	Transient Events	State Evolution	Behavior Patterns
FingerTip 20K	Mobile	Text / Image	✗	✗	✗	✗	✓
AndroidIntent	Mobile	Text	✗	✗	✗	✗	✓
MemGUI-Bench	Mobile	Text / Image	✗	✗	✗	✗	✗
ScreenshotVQA	PC	Image	✓	✗	✗	✗	✗
MobileMem	Mobile	Text / Image	✗	✓	✗	✓	✓
ScreenMem	**PC + Mobile**	**Video**	✓	✓	✓	✓	✓

- We systematically evaluate 15 memory systems and analyze performance across platforms, capabilities, temporal distance, and efficiency, revealing substantial headroom for future computer-use memory systems.

2 Related Work

2.1 Agent Memory Systems

Long-term memory enables agents to preserve information beyond a single context window. Text-centric systems such as Mem0, A-MEM, MemOS, and EverMemOS extract, organize, consolidate, and retrieve salient information from conversational or interaction histories (Chhikara et al., 2025; Xu et al., 2026a; Li et al., 2025; Hu et al., 2026). Recent multimodal and streaming systems extend persistent memory to continuous visual observations using structured episodic or semantic memories, multi-scale representations, or compressed visual states (Wang & Chen, 2025; Long et al., 2026; Yeo et al., 2026; Liang et al., 2026; Wen et al., 2026; Zhang et al., 2026; Xie et al., 2026). These methods provide representative paradigms for long-horizon retention under bounded context or storage, ranging from explicit semantic memory to compact latent visual representations. Our focus is the evaluation regime in which such systems must form persistent memory causally from day-long device activity before future information needs are known.

2.2 Benchmarks for Long-Term and Streaming Memory

Long-term memory has been evaluated across conversational and interactive settings. LoCoMo and LongMemEval study retrieval and reasoning over extended interaction histories, while MemGym evaluates memory formed during long-horizon agent execution (Maharana et al., 2024; Wu et al., 2024; Xu et al., 2026b). These benchmarks establish general challenges in persistent memory but do not specifically target everyday computer-use activity.

Recent computer-use benchmarks investigate memory and personalization from different forms of interaction history. FingerTip 20K and AndroidIntent leverage long-term user records for personalized or proactive mobile assistance, while MemGUI-Bench evaluates memory across repeated GUI tasks (Yang et al., 2025a; Lyu et al., 2026; Liu et al., 2026a). ScreenshotVQA and MobileMem move closer to persistent computer-use memory through temporally ordered screen observations and long-term mobile experience histories, respectively (Wang & Chen, 2025; Deng et al., 2026). These benchmarks cover complementary aspects of personalization, cross-session experience, and temporal memory, but differ in how memory is formed and what information is explicitly evaluated.

Related streaming-video benchmarks study memory or temporally situated interaction over continuous visual observations (Liu et al., 2026b; Forte et al., 2026; Yang et al., 2025b; Shi et al., 2026). In contrast, ScreenMem specifically isolates proactive, query-agnostic memory maintenance over day-long PC and mobile UI streams: observations are ingested causally while future evaluation questions remain hidden, and subsequent answers rely only on the resulting persistent memory.

3 SCREENMEM

We introduce **ScreenMem**, a benchmark for evaluating proactive, query-agnostic memory maintenance over long-horizon computer-use streams. Unlike conventional video question answering, where a query may guide which parts of the input should be inspected, ScreenMem requires a system to construct and maintain persistent memory before future information needs are known. We build trace-grounded user-day streams across PC and mobile devices and pair them with evidence-grounded questions that probe complementary long-term memory capabilities.

3.1 PROBLEM FORMULATION

Let a user-day stream be a temporally ordered sequence

$$\mathcal{S} = \{(x_t, \tau_t)\}_{t=1}^{T}, \tag{1}$$

where x_t denotes the device observation at timestamp τ_t. The stream is processed causally: at time t, the system has access only to observations up to the current time. Persistent memory is updated incrementally as

$$\mathcal{M}_t = \mathcal{U}_\theta(\mathcal{M}_{t-1}, x_t, \tau_t), \qquad t = 1, \ldots, T, \tag{2}$$

where $\mathcal{U}_\theta$ may represent any memory-maintenance mechanism.

Each user day is associated with an evaluation set

$$\mathcal{Q} = \{(q_i, a_i, c_i, \mathcal{E}_i)\}_{i=1}^{N}, \tag{3}$$

where q_i and a_i denote a question and its reference answer, c_i denotes the memory capability being evaluated, and $\mathcal{E}_i$ denotes the supporting temporal evidence. Crucially, $\mathcal{Q}$ remains unavailable throughout stream ingestion.

After the final observation, memory is sealed as $\mathcal{M}^* = \mathcal{M}_T$. Each evaluation question must then be answered solely from this persistent memory:

$$\hat{a}_i = \mathcal{R}_\phi(q_i, \mathcal{M}^*), \tag{4}$$

without access to the original stream or evidence annotations. This formulation directly evaluates whether a system can preserve useful information under causal observation before knowing what it will later be asked.

3.2 USER-DAY STREAM CONSTRUCTION

Directly collecting and releasing day-long screen recordings of real users raises substantial privacy concerns. We therefore adopt a *trace-grounded reconstruction* strategy: real-world interaction trajectories are derived from existing open-source datasets, while their visual content is recreated in controllable virtual environments. This preserves realistic behavioral structure without releasing the original users' screen recordings. Figure 2 summarizes the construction pipeline.

Trace sources and selection. PC interactions are derived from RLKWiC (Bakhshizadeh et al., 2024), while mobile interactions are derived from FingerTip 20K (Yang et al., 2025a). We compute application occurrence frequencies in the source traces and primarily select interactions involving frequently appearing applications, while also considering application diversity to broaden behavioral coverage. The resulting traces preserve realistic task and interaction trajectories grounded in real-world computer use.

Application reconstruction. Selected interactions are instantiated in executable virtual environments, using OS.js (Team, 2021) for PC applications and MobileGym-compatible applications (Wu et al., 2026; Liu et al., 2026c) for mobile devices. Rather than reproducing the original applications at the pixel level, we reconstruct the interaction semantics and visually observable information required by each trace.

Applications are implemented using reusable interface skeletons together with scenario-specific content. For applications that recur across multiple interactions, deterministic seeds instantiate different documents, messages, settings, or task contexts while preserving a common interface structure. This design allows the same reconstructed application to support diverse interaction episodes while remaining fully reproducible.

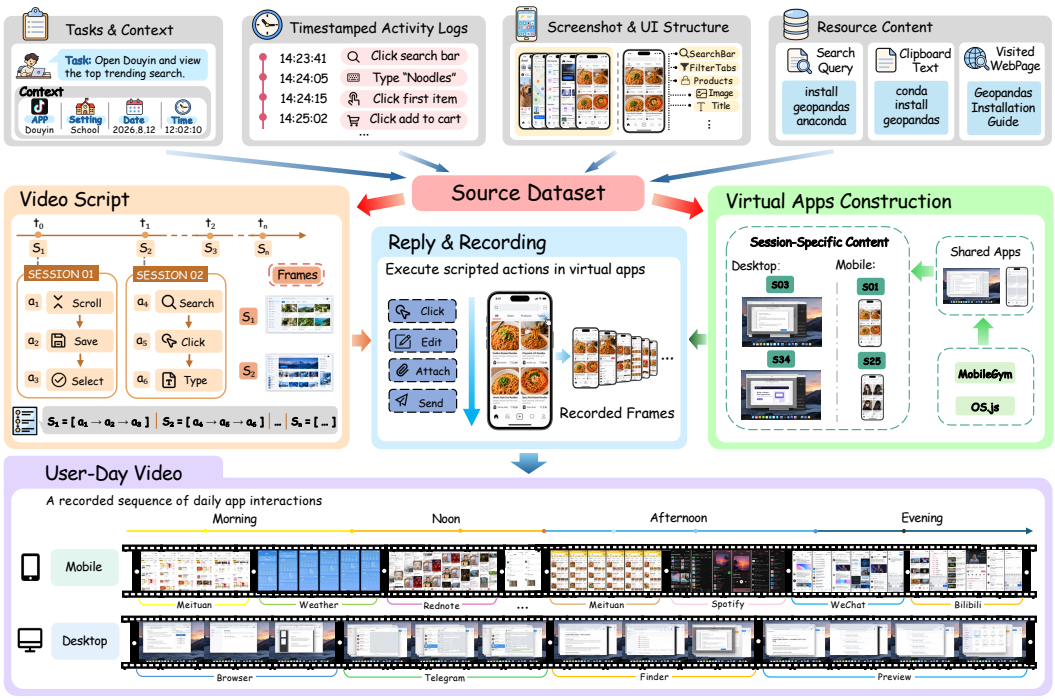


Figure 2: Trace-grounded construction of user-day streams.

Replay and temporal composition. Each selected trace is converted into an executable action script and replayed automatically while the device display is continuously recorded. Individual interaction episodes are then temporally composed into a user-day stream according to their reconstructed daily schedule.

Importantly, we retain inactive intervals between interactions rather than concatenating only active segments. The resulting streams therefore contain both meaningful interface changes and extended periods with little or no new information, requiring memory systems to distinguish salient observations from temporal redundancy during continuous ingestion.

3.3 Question Design and Annotation

The evaluation questions are designed to probe four complementary requirements of persistent computer-use memory:

- **Facts & States:** retaining factual information or an application/device state observed at a particular time, such as a filename, message content, or interface setting.
- **State Evolution:** tracking information whose valid state changes over time and distinguishing earlier states from subsequent updates.
- **Activities & Transient Events:** preserving user activities and short-lived information, such as notifications or temporary interface events, that may disappear from subsequent observations.
- **Behavioral Patterns:** aggregating evidence across temporally separated interactions to recover recurring behaviors, routines, or preferences.

Question construction follows a semi-automated annotation pipeline. Candidate question–answer pairs are generated from reconstruction scripts and interaction metadata, aligned with their supporting evidence, automatically checked with an LLM, and subsequently verified by human annotators. We retain only questions that are answerable from visually observable evidence and admit a unique, deterministic reference answer. Questions involving subjective interpretation, insufficient evidence, or multiple valid answer forms are removed.

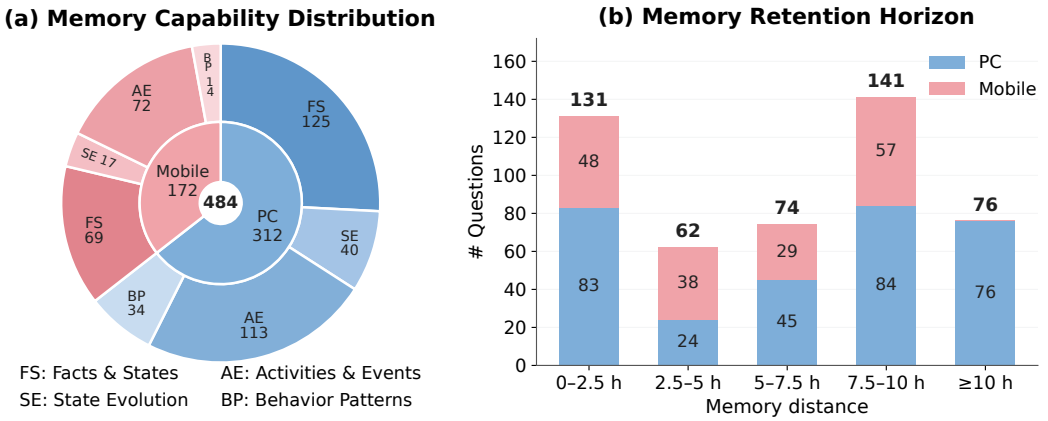


Figure 3: Characteristics of ScreenMem. (a) Distribution of evaluation questions across four memory capabilities, with the inner ring showing the PC/mobile split. (b) Distribution of questions by memory distance from supporting evidence to end-of-day evaluation.

For each question q_i, we annotate one or more supporting temporal intervals

$$\mathcal{E}_i = \{[s_{i,k}, e_{i,k}]\}_{k=1}^{K_i}. \tag{5}$$

These annotations are used only for dataset validation and analysis and are never exposed to evaluated systems.

To characterize the retention horizon required by each question, we define its *memory distance* as the elapsed time from the start of its latest supporting-evidence interval to the end of the corresponding user-day stream:

$$\Delta_i = T - \max_k s_{i,k}. \tag{6}$$

A larger Δ_i therefore indicates that the relevant information must remain accessible through a longer sequence of subsequent observations and memory updates. The resulting distribution is analyzed in Section 3.5.

3.4 Evaluation Protocol

ScreenMem strictly separates *memory formation* from *memory use*. During ingestion, observations arrive chronologically in their original temporal order, future observations are unavailable, and all evaluation questions remain hidden. A system may continuously update its persistent memory as the stream progresses, but it cannot use future questions to determine what should be retained.

After the complete user-day stream has been processed, the resulting memory is sealed. The system can no longer revisit the original stream, access reconstruction metadata or evidence annotations, or modify its memory using evaluation questions. All questions associated with the user day are subsequently answered from this same sealed memory.

This protocol ensures that benchmark performance reflects information preserved during causal ingestion rather than query-conditioned inspection of the original interaction history. It therefore evaluates the central challenge of proactive, query-agnostic memory maintenance: deciding what to retain before future information needs are known.

3.5 Benchmark Statistics

ScreenMem contains 10 user-day streams spanning 100.1 hours of PC and mobile computer use, with 1,949 interaction sessions and 484 human-verified questions. Table 2 summarizes the overall benchmark scale, while Figure 3 shows the distribution of memory capabilities and retention horizons. The questions cover all four target memory capabilities. The median memory distance is 7.9 hours on PC and 5.0 hours on mobile, and 41.3% of all questions require retaining information for at least 8 hours.

Table 2: Statistics of ScreenMem. Duration includes inactive intervals, and a session denotes a contiguous foreground-application interaction.

Platform	# Days	Duration	Avg.	# Apps	# Sessions	# Questions
PC	7	71.1 h	10.2 h	17	1,340	312
Mobile	3	29.0 h	9.7 h	10	609	172
Overall	**10**	**100.1 h**	**10.0 h**	**27**	**1,949**	**484**

4 EXPERIMENTS

We evaluate how existing memory systems perform under the query-agnostic, day-long computer-use setting introduced in Section 3.4. We first report overall benchmark performance, then analyze capability-specific behavior, performance across memory distances, and quality–efficiency trade-offs.

4.1 EXPERIMENTAL SETUP

Systems and adaptation. We evaluate 15 memory systems spanning four paradigms: recent-context memory (SimpleStream) (Shen et al., 2026), implicit visual memory (InfiniPot-V, StreamForest, ReKV, HERMES, and FluxMem) (Kim et al., 2026; Zeng et al., 2026; Di et al., 2025; Zhang et al., 2026; Xie et al., 2026), explicit multimodal memory (MIRIX, M3-Agent, WorldMM, EventMemAgent, and OASIS) (Wang & Chen, 2025; Long et al., 2026; Yeo et al., 2026; Wen et al., 2026; Liang et al., 2026), and generic text memory (A-MEM, Mem0, MemOS, and EverMemOS) (Xu et al., 2026a; Chhikara et al., 2025; Li et al., 2025; Hu et al., 2026). All systems follow the protocol in Section 3.4: they construct memory during causal ingestion and answer evaluation questions only after the resulting memory has been sealed. We preserve each method's native memory mechanism whenever possible. Model-agnostic systems use Qwen3-VL-8B-Instruct (Bai et al., 2025), with Qwen3-VL-Embedding-2B (Li et al., 2026) when an external retriever is required; methods tightly coupled to a particular backbone or KV representation retain their native models. Because generic text-memory systems cannot directly ingest continuous video, they receive a shared sequence of 10-second textual observations generated by Qwen3-VL-8B-Instruct. Complete configurations and method-specific adaptations are provided in Appendix A.

Metrics. Question-level accuracy is the primary metric. We report PC and mobile results separately and an Overall micro-average over all 484 questions. All 15 systems have complete runs on both platforms. For efficiency, we measure ingestion real-time factor $\mathrm{RTF} = T_{\mathrm{ingest}}/T_{\mathrm{stream}}$, ingestion token usage, and final persistent-memory storage; generic text-memory costs include the required video-to-text preprocessing pass. Detailed scoring and resource accounting are given in Appendices E and D.

4.2 OVERALL AND CAPABILITY-LEVEL PERFORMANCE

Table 3 summarizes both the primary benchmark leaderboard and the capability-level breakdown. PC and mobile columns report platform-level accuracy, while Overall and the four capability columns are micro-averaged over both platforms.

Day-long query-agnostic memory remains difficult. MemOS reaches 31.4% Overall accuracy versus 2.7% for SimpleStream; the next strongest systems, Mem0 and EverMemOS, reach 24.8% and 23.8%, respectively. The results also reveal substantial platform-dependent variation: OASIS rises from 15.7% on PC to 21.5% on mobile, whereas MIRIX and EverMemOS fall from 21.5% to 9.9% and from 29.2% to 14.0%. Because the two platforms contain different user days and question compositions, these changes characterize benchmark behavior rather than a controlled domain-generalization gap.

State evolution remains a key challenge. State Evolution is the weakest capability for most systems: MemOS reaches 15.8%, Mem0 and MIRIX both reach 12.3%, and no implicit visual-memory system answers one correctly. Behavioral Patterns show a different profile: because all are four-option multiple choice, 25% is a natural chance reference; recent-context and most implicit systems

Table 3: Overall and capability-level accuracy (%) on ScreenMem. Overall and capability results are micro-averaged across PC and mobile. **Bold**/underline: strongest/second strongest; ties are preserved. F&S: Facts & States; Evol.: State Evolution; Act./Event: Activities & Transient Events; Behav.: Behavioral Patterns.

Method	PC	Mobile	F&S	Evol.	Act./Event	Behav.	Overall
Recent-Context Memory							
SimpleStream	3.2	1.7	0.0	0.0	2.2	18.8	2.7
Implicit Visual Memory							
InfiniPot-V	3.8	4.1	0.0	0.0	2.2	31.2	3.9
StreamForest	7.1	2.9	2.6	0.0	4.3	29.2	5.6
ReKV	3.5	4.7	1.0	0.0	4.3	18.8	3.9
HERMES	2.6	4.1	0.0	0.0	4.3	14.6	3.1
FluxMem	3.8	0.0	0.0	0.0	3.2	12.5	2.5
Explicit Multimodal Memory							
MIRIX	21.5	9.9	21.6	12.3	9.2	37.5	17.4
M3-Agent	25.0	18.6	24.7	5.3	20.0	45.8	22.7
WorldMM	7.4	15.1	5.7	0.0	8.1	47.9	10.1
EventMemAgent	2.2	5.8	3.6	0.0	1.6	14.6	3.5
OASIS	15.7	21.5	20.1	8.8	13.5	35.4	17.8
Generic Text Memory							
A-MEM	21.8	12.2	17.0	5.3	10.8	68.8	18.4
Mem0	27.6	19.8	23.2	12.3	21.1	60.4	24.8
MemOS	**29.2**	**35.5**	**32.5**	**15.8**	**24.3**	**72.9**	**31.4**
EverMemOS	**29.2**	14.0	24.2	3.5	21.1	56.2	23.8

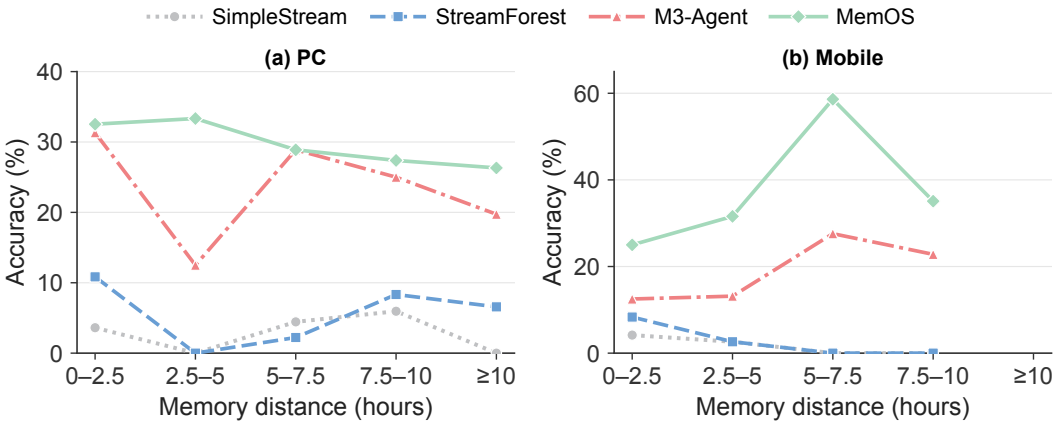


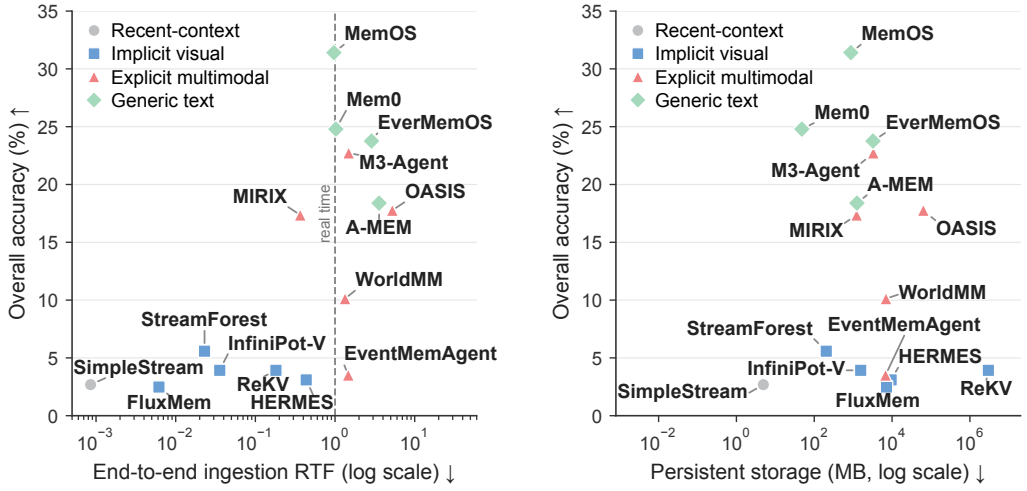
Figure 4: Distance-conditioned accuracy on **(a) PC** and **(b) mobile** for one representative per memory paradigm. Mobile has no questions at $\geq$ 10 hours. The profiles emphasize descriptive strata rather than a controlled forgetting curve.

remain near or below it, while several semantic-memory systems exceed it substantially (MemOS: 72.9%). Platform-specific and question-format breakdowns are in Appendix B.1.

4.3 Distance-Conditioned Long-Horizon Performance

We stratify accuracy by memory distance, the gap between the latest supporting evidence and end-of-day evaluation (Section 3.3). Because naturally occurring distances co-vary with question and user-day composition, we treat distance as a diagnostic of long-horizon accessibility rather than a controlled forgetting variable. Figure 4 reports one representative per memory paradigm; full results appear in Appendix C.

The profiles are non-monotonic. On PC, MemOS stays within 26.3–33.3% across all five ranges, whereas M3-Agent changes from 31.3% at 0–2.5 hours to 19.7% beyond 10 hours; mobile shows



(a) Accuracy vs. ingestion RTF **(b) Accuracy vs. persistent storage**

Figure 5: Quality–efficiency trade-offs: **(a)** Overall accuracy vs. end-to-end RTF (lower is faster; RTF $= 1$ is real time) and **(b)** vs. persistent-memory footprint. Text-memory RTF includes video-to-text preprocessing.

similar fluctuations, with MemOS reaching 58.6% at 5–7.5 hours and 35.1% at 7.5–10 hours. This does not imply improved retention at intermediate distances because bin composition varies substantially (Appendix C). Crucially, in the PC $\geq$ 10-hour bin, MemOS and M3-Agent still answer 11 and 10 short-answer Facts & States or State Evolution questions correctly. Thus, long-range accuracy is not solely multiple-choice chance, while temporal distance alone does not determine accessibility in these naturalistic streams.

4.4 Quality–Efficiency Trade-offs

Persistent computer-use memory must operate continuously, so retention quality should be considered together with ingestion and storage cost. Figure 5 compares Overall accuracy with end-to-end RTF and persistent-memory footprint for all 15 systems.

Fast ingestion does not imply effective retention: several recent/implicit systems are fast but inaccurate, while some latent/KV methods retain large states despite low RTF. Among higher-accuracy systems, the trade-off remains heterogeneous: MemOS reaches 31.4% at approximately real-time ingestion, whereas MIRIX is faster but less accurate and OASIS is substantially slower despite comparable Overall accuracy to MIRIX. Token workload also differs substantially: M3-Agent and WorldMM use about 1.6M input tokens/hour, versus 4.2M for end-to-end MemOS and 7.8M for Mem0. Per-day cost is nearly duration-proportional for fixed-rate pipelines but more variable for MIRIX (Appendix D.4). Because token units differ across mechanisms, we use them descriptively; complete resource results are in Appendix D.

5 Conclusion

We have introduced ScreenMem, a benchmark for proactive, query-agnostic memory maintenance over long-horizon computer-use streams. By separating causal memory formation from later question answering, the benchmark tests what information a system preserves before future information needs are known rather than how well it can search a history after a query arrives. Across 10 reconstructed PC and mobile user days, ScreenMem provides 100.1 hours of computer-use streams and 484 human-verified questions covering factual and device states, state evolution, activities and transient events, and behavioral patterns. Our evaluation of 15 existing memory systems shows substantial headroom in both retention quality and practical efficiency, with distinct failure modes across memory paradigms, capabilities, temporal distances, and platforms. The strongest system reaches 31.4% Overall accuracy, with 29.2% on PC and 35.5% on mobile. ScreenMem can provide a reproducible foundation for developing persistent memory systems that continuously observe everyday computer use and preserve information that remains useful hours later.

AI USE STATEMENT

We used AI tools at several stages of this work as research assistance. For manuscript preparation, AI tools were used to improve clarity and conciseness, identify redundant exposition, polish language, and provide preliminary structural drafts for selected passages. All generated text was subsequently reviewed, revised, and approved by the authors. AI-assisted retrieval was also used to discover potentially relevant literature; all cited works and bibliographic information were independently checked by the authors against their original sources.

During research development and execution, AI coding assistants were used to help draft and debug implementation and analysis code. They also assisted in constructing parts of the executable virtual applications used in our trace-grounded reconstruction pipeline. The benchmark construction process further uses LLM assistance in the semi-automated question-generation and verification pipeline described in Sec. 3.3. AI-generated code, reconstructed content, and candidate annotations were subject to human inspection and validation before being used in the benchmark or experiments.

AI tools were not used as a substitute for author judgment in determining the benchmark formulation, evaluation protocol, reported experimental results, or scientific conclusions.

REPRODUCIBILITY STATEMENT

We provide detailed specifications of the ScreenMem construction and evaluation procedure to support reproducibility. Sec. 3 defines the benchmark formulation, trace-grounded user-day construction, question annotation procedure, memory-distance definition, and causal sealed-memory evaluation protocol. Sec. 4 specifies the evaluated memory systems, common model configurations, evaluation metrics, and efficiency measurements.

The appendix provides additional implementation details, including system-specific adaptations and model serving configurations, complete platform- and capability-level results, memory-distance analyses, resource accounting, and deterministic answer-matching rules. All reported benchmark accuracy is computed using the same deterministic scoring procedure without method-specific manual correction.

We will publicly release ScreenMem together with the reconstructed benchmark data, evaluation code, scoring scripts, and configurations required to reproduce the benchmark protocol and evaluate new memory systems. Together with the implementation details documented in the paper, these resources are intended to support independent reproduction of our results and standardized evaluation of future methods.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Mahta Bakhshizadeh, Christian Jilek, Markus Schröder, Heiko Maus, and Andreas Dengel. Data collection of real-life knowledge work in context: The rlkwic dataset. In *International Conference on Information Management*, pp. 277–290. Springer, 2024.

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.

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*, 2026.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *International Conference on Learning Representations*, volume 2025, pp. 42115–42127, 2025.

Rosario Forte, Giuseppe Lando, and Antonino Furnari. Egostream: A diagnostic benchmark for streaming episodic memory in egocentric vision. *arXiv preprint arXiv:2605.31557*, 2026.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, et al. Evermemos: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853, 2026.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *Advances in Neural Information Processing Systems*, 38:138983–139013, 2026.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

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

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. Oasis: On-demand hierarchical event memory for streaming video reasoning. *arXiv preprint arXiv:2604.17052*, 2026.

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.

Guanming Liu, Yuqi Ren, Hansu Gu, Peng Zhang, Weihang Wang, Jiahao Liu, Ning Gu, and Tun Lu. Streammembench: Streaming evaluation of agent memory for future-oriented assistance. *arXiv preprint arXiv:2606.14571*, 2026b.

Guohong Liu, Jialei Ye, Pengzhi Gao, Wei Liu, Jian Luan, Yunxin Liu, and Yuanchun Li. Scalewob: Guiding gui agents with coding agents via large-scale environmental synthesis. *arXiv preprint arXiv:2605.25160*, 2026c.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *International Conference on Learning Representations*, volume 2026, pp. 146197–146246, 2026.

Yibo Lyu, Gongwei Chen, Rui Shao, Weili Guan, and Liqiang Nie. Personalalign: Hierarchical implicit intent alignment for personalized gui agent with long-term user-centric records. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36074–36089, 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.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, volume 2025, pp. 406–441, 2025.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding. *arXiv preprint arXiv:2604.02317*, 2026.

Yansong Shi, Qingsong Zhao, Tianxiang Jiang, Xiangyu Zeng, Yi Wang, and Limin Wang. River: A real-time interaction benchmark for video llms. *arXiv preprint arXiv:2603.03985*, 2026.

OS.js Team. Os.js, July 2021. URL `https://github.com/os-js/OS.js/releases/tag/3.1.12`. Version 3.1.12.

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

Siwei Wen, Zhangcheng Wang, Xingjian Zhang, Lei Huang, and Wenjun Wu. Eventmemagent: Hierarchical event-centric memory for online video understanding with adaptive tool use. *arXiv preprint arXiv:2602.15329*, 2026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

Dingbang Wu, Rui Hao, Haiyang Wang, Shuzhe Wu, Han Xiao, Zhenghong Li, Bojiang Zhou, Zheng Ju, Zichen Liu, Lue Fan, et al. Mobilegym: A verifiable and highly parallel simulation platform for mobile gui agent research. *arXiv preprint arXiv:2605.26114*, 2026.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. Fluxmem: Adaptive hierarchical memory for streaming video understanding. *arXiv preprint arXiv:2603.02096*, 2026.

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, 2026a.

Wujiang Xu, Yu Wang, Kai Mei, Kaiqu Liang, Zhenting Wang, Mingyu Jin, Han Zhang, Shi-Xiong Zhang, Wenyue Hua, Sambit Sahu, et al. Memgym: a long-horizon memory environment for llm agents. *arXiv preprint arXiv:2605.20833*, 2026b.

Qinglong Yang, Haoming Li, Haotian Zhao, Xiaokai Yan, Jingtao Ding, Fengli Xu, and Yong Li. Fingertip 20k: A benchmark for proactive and personalized mobile llm agents. *arXiv preprint arXiv:2507.21071*, 2025a.

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. In *International Conference on Learning Representations*, volume 2025, pp. 42522–42556, 2025b.

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, 2026.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, et al. Streamforest: Efficient online video understanding with persistent event memory. *Advances in Neural Information Processing Systems*, 38:75804–75835, 2026.

Haowei Zhang, Shudong Yang, Jinlan Fu, See Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8411–8430, 2026.

A Implementation Details

This section provides additional implementation details for the memory systems evaluated on ScreenMem. Our objective is to preserve the defining memory mechanism of each method while adapting model and input interfaces when necessary for day-long computer-use streams. Because several memory mechanisms are intrinsically coupled to different backbones or representations, the benchmark should be interpreted as an evaluation of end-to-end memory systems under a common protocol rather than a controlled comparison of memory algorithms under an identical neural backbone.

Table 4: Implementation summary of the evaluated memory systems. "Native" indicates that the representation or backbone required by the released implementation is retained. Generic text-memory methods receive the same 10-second caption stream generated by Qwen3-VL-8B-Instruct.

Method	Memory Type	Backbone Used	Input	Retrieval
SimpleStream	Recent context	Qwen3-VL-8B-Instruct	Visual stream	Recent window
InfiniPot-V	Implicit visual	Qwen2.5-VL-7B	Native visual input	Native
StreamForest	Implicit visual	StreamForest-Qwen2-7B	Native visual input	Native
ReKV	Implicit KV	LongVA-7B	Native visual input	Native
HERMES	Implicit visual	Qwen2.5-VL-7B	Native visual input	Native
FluxMem	Implicit visual	Qwen2.5-VL-7B	Native visual input	Native
MIRIX	Explicit multimodal	Qwen3-VL-8B-Instruct	Native visual input	Qwen3-VL-Embedding-2B
M3-Agent	Explicit multimodal	Qwen3-VL-8B-Instruct	Native visual input	Qwen3-VL-Embedding-2B
WorldMM	Explicit multimodal	Qwen3-VL-8B-Instruct	Native visual input	Qwen3-VL-Embedding-2B
EventMemAgent	Explicit multimodal	EventMemAgent-8B	Native visual input	Qwen3-VL-Embedding-2B
OASIS	Explicit multimodal	Qwen3-VL-8B-Instruct	Native visual input	Qwen3-VL-Embedding-2B
A-MEM	Generic text	Qwen3-VL-8B-Instruct	10-s captions	Qwen3-VL-Embedding-2B
Mem0	Generic text	Qwen3-VL-8B-Instruct	10-s captions	Qwen3-VL-Embedding-2B
MemOS	Generic text	Qwen3-VL-8B-Instruct	10-s captions	Qwen3-VL-Embedding-2B
EverMemOS	Generic text	Qwen3-VL-8B-Instruct	10-s captions	Qwen3-VL-Embedding-2B

A.1 System Configurations

Table 4 summarizes the principal configurations used in our experiments.

A.2 Model Serving

All model-based systems are evaluated on NVIDIA A800 GPUs with 80 GB of memory. For ingestion-time measurements, each memory system is served independently to reduce cross-system scheduling interference. We preserve each method's official model when its memory representation is tightly coupled to a particular backbone, learned parameterization, or KV-cache structure. For model-agnostic methods or implementations originally relying on proprietary APIs, we use Qwen3-VL-8B-Instruct as the common multimodal backbone.

A.3 Adaptation of Generic Text-Memory Systems

A-MEM, Mem0, MemOS, and EverMemOS were developed primarily for textual interaction histories and cannot directly ingest continuous screen recordings. To provide a common interface, each user-day stream is partitioned into non-overlapping 10-second windows. Qwen3-VL-8B-Instruct converts each window into a concise textual observation describing the visible interface state and user activity. The resulting observations preserve their original temporal order and are sequentially passed to each memory system.

All four systems receive exactly the same caption stream. For accuracy evaluation, caption generation is therefore identical across systems. For efficiency evaluation, one caption-generation pass is included in the end-to-end ingestion cost of each system because this preprocessing would be required when deploying that system independently on the original visual stream.

A.4 Method-Specific Adaptations

MIRIX. MIRIX receives screenshots sampled every 3 seconds in our long-horizon evaluation, and the same sampling interval is used for all PC and mobile runs. Its six memory stores are updated synchronously during ingestion, and question answering is performed only from the persisted memories after the stream has been sealed. For compatibility with Qwen3-VL, the additional system message in the final QA turn is passed as a user message.

ReKV. We evaluate ReKV using the LongVA-7B-based implementation provided in the released code. The evaluation stores 144 visual tokens per frame and uses a local window of 8,000 tokens, top-$k = 32$ retrieval, and a maximum of 128 cached blocks. For day-long execution, we disable pinned-memory and non-blocking CPU transfer when evicted KV states are moved from GPU to CPU. This modification affects transfer behavior but not the retained KV contents or retrieval procedure.

FluxMem. The sealed-memory protocol requires the stream to be ingested before evaluation questions become available. We therefore process each stream once, save the resulting compressed visual cache, and answer all subsequent questions solely from this cache without reopening the original stream. During QA, we perform autoregressive decoding through FluxMem's forward interface using the saved compressed visual cache.

HERMES. We retain the released question-answering procedure. When greedy decoding immediately produces an end-of-sequence token, the resulting empty response is scored as incorrect rather than rerunning the model with a modified decoding strategy.

Persistent storage. For each system, storage includes the persistent artifacts required for question answering after ingestion. For MemOS, memories are attributed to the corresponding user-day identifier rather than measuring an entire shared vector database. For MIRIX, storage includes the sealed SQLite database and image files; for M3-Agent, it includes the sealed graph and JSON metadata. For OASIS, WorldMM, and EventMemAgent, storage includes the sealed memory records together with the stored frames that question answering can retrieve. For A-MEM and Mem0, storage includes the complete persisted vector-store directory, including index files. For EverMemOS, storage includes the user-day's episodic, event-log, and foresight records in MongoDB, Elasticsearch storage apportioned by document count, and Milvus vector payloads. The Milvus component excludes unmeasured HNSW index files. For KV- and visual-cache-based systems, we measure the serialized cache states. SimpleStream stores only its final four-frame recent-context window.

B Additional Benchmark Results

The main paper reports the primary PC, mobile, and Overall leaderboard together with the cross-platform capability analysis. Here, we provide platform-specific capability results, question-format breakdowns, and per-user-day results.

B.1 Platform-Specific Capability Results

B.2 Results by Question Format

The benchmark contains both short-answer and multiple-choice questions. Because answer format is correlated with capability—in particular, all Behavioral Pattern questions are four-option multiple choice—we separately report performance by format on the PC subset. For four-option questions, 25% provides a natural chance reference, allowing us to distinguish near-chance behavior from systems that achieve substantially higher accuracy.

Multiple-choice accuracy is generally higher than short-answer accuracy, as expected from the constrained answer space. However, the spread within the same format remains large: several recent-context and implicit visual-memory systems stay near the 25% chance reference, whereas stronger explicit multimodal and text-memory systems substantially exceed it. Thus, answer format explains part of the absolute performance difference, while the within-format gaps still reveal substantial variation in retained information across memory systems.

B.3 Per-User-Day Results

Table 8 reports accuracy separately for each reconstructed user day. The seven PC user days contain 12, 21, 30, 44, 49, 73, and 83 questions. The three mobile user days contain 54, 56, and 62 questions.

Table 5: Capability-level accuracy (%) on the 312 PC questions. Facts & States, State Evolution, Activities & Transient Events, and Behavioral Patterns contain 125, 40, 113, and 34 questions, respectively.

Method	All	F&S	Evol.	Act./Event	Behav.
SimpleStream	3.2	0.0	0.0	2.7	20.6
InfiniPot-V	3.8	0.0	0.0	3.5	23.5
StreamForest	7.1	4.0	0.0	4.4	35.3
ReKV	3.5	1.6	0.0	1.8	20.6
HERMES	2.6	0.0	0.0	3.5	11.8
FluxMem	3.8	0.0	0.0	5.3	17.6
MIRIX	21.5	29.6	15.0	11.5	32.4
M3-Agent	25.0	26.4	5.0	26.5	38.2
WorldMM	7.4	0.0	0.0	8.0	41.2
EventMemAgent	2.2	0.0	0.0	2.7	11.8
OASIS	15.7	16.8	7.5	15.9	20.6
A-MEM	21.8	21.6	5.0	14.2	67.6
Mem0	27.6	24.8	15.0	26.5	55.9
MemOS	29.2	25.6	10.0	27.4	70.6
EverMemOS	29.2	29.6	5.0	26.5	64.7

Table 6: Capability-level accuracy (%) on the 172 mobile questions for all 15 systems. The four categories contain 69, 17, 72, and 14 questions, respectively.

Method	All	F&S	Evol.	Act./Event	Behav.
SimpleStream	1.7	0.0	0.0	1.4	14.3
InfiniPot-V	4.1	0.0	0.0	0.0	50.0
StreamForest	2.9	0.0	0.0	4.2	14.3
ReKV	4.7	0.0	0.0	8.3	14.3
HERMES	4.1	0.0	0.0	5.6	21.4
FluxMem	0.0	0.0	0.0	0.0	0.0
MIRIX	9.9	7.2	5.9	5.6	50.0
M3-Agent	18.6	21.7	5.9	9.7	64.3
WorldMM	15.1	15.9	0.0	8.3	64.3
EventMemAgent	5.8	10.1	0.0	0.0	21.4
OASIS	21.5	26.1	11.8	9.7	71.4
A-MEM	12.2	8.7	5.9	5.6	71.4
Mem0	19.8	20.3	5.9	12.5	71.4
MemOS	35.5	44.9	29.4	19.4	78.6
EverMemOS	14.0	14.5	0.0	12.5	35.7

The results reveal substantial day-to-day variation for several systems. For example, OASIS ranges from 10.7% on D9 to 29.6% on D8, while EverMemOS ranges from 5.4% on D9 to 24.1% on D8, reinforcing that aggregate platform scores average over heterogeneous user-day compositions.

C Additional Long-Horizon Analysis

The main paper reports representative distance-conditioned performance profiles on both PC and mobile streams. Here, we provide complete PC and mobile results for all 15 systems. These bins are benchmark strata, not matched experimental conditions: their user-day, capability, answer-format, and semantic composition can differ substantially.

Table 7: PC performance (%) by question format. The 312 PC questions contain 264 short-answer questions and 48 four-option multiple-choice questions. Random guessing on the multiple-choice subset yields 25% accuracy.

Method	Short Answer	MCQ	Overall
SimpleStream	0.0	20.8	3.2
InfiniPot-V	0.0	25.0	3.8
StreamForest	1.9	35.4	7.1
ReKV	0.8	18.8	3.5
HERMES	0.0	16.7	2.6
FluxMem	0.4	22.9	3.8
MIRIX	18.6	37.5	21.5
M3-Agent	21.2	45.8	25.0
WorldMM	0.0	47.9	7.4
EventMemAgent	0.4	12.5	2.2
OASIS	13.3	29.2	15.7
A-MEM	13.3	68.8	21.8
Mem0	20.8	64.6	27.6
MemOS	21.2	72.9	29.2
EverMemOS	21.6	70.8	29.2

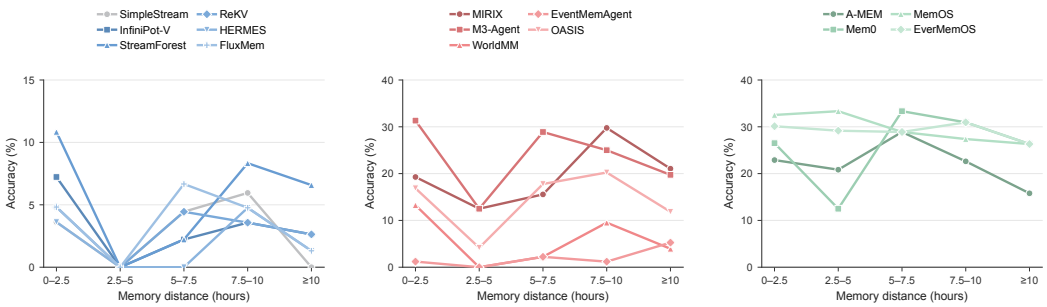
Table 8: Per-user-day accuracy (%). D1–D7 are PC user days and D8–D10 are mobile user days. PC Avg. is accuracy over all 312 PC questions; all systems are evaluated on all three mobile user days.

Method	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	PC Avg.
SimpleStream	8.3	4.8	3.3	4.5	0.0	6.8	0.0	1.9	1.8	1.6	3.2
InfiniPot-V	0.0	4.8	6.7	0.0	6.1	6.8	1.2	1.9	1.8	8.1	3.8
StreamForest	0.0	4.8	6.7	6.8	12.2	8.2	4.8	1.9	5.4	1.6	7.1
ReKV	8.3	4.8	3.3	4.5	4.1	4.1	1.2	7.4	3.6	3.2	3.5
HERMES	8.3	4.8	0.0	0.0	2.0	5.5	1.2	1.9	7.1	3.2	2.6
FluxMem	0.0	0.0	0.0	2.3	4.1	9.6	2.4	0.0	0.0	0.0	3.8
MIRIX	25.0	28.6	6.7	20.5	32.7	21.9	18.1	9.3	12.5	8.1	21.5
M3-Agent	41.7	19.0	13.3	27.3	32.7	27.4	20.5	16.7	16.1	22.6	25.0
WorldMM	16.7	9.5	3.3	2.3	8.2	11.0	6.0	18.5	12.5	14.5	7.4
EventMemAgent	0.0	4.8	0.0	0.0	2.0	1.4	4.8	5.6	3.6	8.1	2.2
OASIS	41.7	9.5	10.0	15.9	30.6	17.8	4.8	29.6	10.7	24.2	15.7
A-MEM	33.3	4.8	23.3	20.5	32.7	19.2	20.5	13.0	10.7	12.9	21.8
Mem0	50.0	33.3	26.7	25.0	16.3	34.2	25.3	22.2	12.5	24.2	27.6
MemOS	58.3	19.0	33.3	20.5	34.7	32.9	24.1	37.0	26.8	41.9	29.2
EverMemOS	50.0	23.8	30.0	25.0	30.6	35.6	22.9	24.1	5.4	12.9	29.2

C.1 Complete PC Memory-Distance Results

Figure 6 groups systems by memory paradigm to avoid an overcrowded 15-line visualization. The first panel combines the recent-context baseline with implicit visual-memory methods and uses a magnified accuracy scale because these methods occupy a substantially lower performance range. All panels use the same five 2.5-hour bins as the dataset statistics and the main PC analysis.

Performance does not decrease monotonically with memory distance, and the bin composition helps explain why a monotonic forgetting interpretation would be inappropriate. For example, 52 of the 76 questions in the PC $\geq$ 10-hour bin come from UD07, whereas 10 of the 24 questions in the 2.5–5-hour bin come from UD03; capability mixtures also change across ranges. The longest PC range nevertheless contains genuine short-answer successes: MemOS answers 11 short-answer Facts &



(a) Recent-context & implicit visual **(b) Explicit multimodal** **(c) Generic text memory**

Figure 6: Complete PC memory-distance results grouped by memory paradigm. Memory distance is measured from the latest supporting evidence to end-of-day evaluation. Panel (a) uses a magnified accuracy range because recent-context and implicit visual-memory systems occupy a substantially lower performance regime. The profiles are descriptive distance-conditioned slices rather than controlled forgetting curves because capability, answer format, and user-day composition vary across bins.

Table 9: PC accuracy (%) by memory distance. The five bins contain 83, 24, 45, 84, and 76 questions, respectively.

Method	**0–2.5 h**	**2.5–5 h**	**5–7.5 h**	**7.5–10 h**	**≥10 h**
SimpleStream	3.6	0.0	4.4	6.0	0.0
InfiniPot-V	7.2	0.0	2.2	3.6	2.6
StreamForest	10.8	0.0	2.2	8.3	6.6
ReKV	4.8	0.0	4.4	3.6	2.6
HERMES	3.6	0.0	0.0	4.8	1.3
FluxMem	4.8	0.0	6.7	4.8	1.3
MIRIX	19.3	12.5	15.6	29.8	21.1
M3-Agent	31.3	12.5	28.9	25.0	19.7
WorldMM	13.3	0.0	2.2	9.5	3.9
EventMemAgent	1.2	0.0	2.2	1.2	5.3
OASIS	16.9	4.2	17.8	20.2	11.8
A-MEM	22.9	20.8	28.9	22.6	15.8
Mem0	26.5	12.5	33.3	31.0	26.3
MemOS	32.5	33.3	28.9	27.4	26.3
EverMemOS	30.1	29.2	28.9	31.0	26.3

States or State Evolution questions correctly, while M3-Agent answers 10. Thus, nonzero long-distance accuracy is not explained solely by chance performance on multiple-choice questions.

C.2 Complete Mobile Memory-Distance Results

The mobile subset contains no questions whose memory distance reaches 10 hours. For consistency with the dataset statistics and the main analysis, we retain the same 2.5-hour bin width and report the four non-empty mobile ranges.

As on PC, accuracy is not monotonic in temporal distance. The mobile bins also differ markedly in composition: the 7.5–10-hour range contains no Behavioral Pattern questions, while 29 of its 57 questions come from UD03. The intermediate-distance peak is not purely a multiple-choice effect either: in the 5–7.5-hour bin, 12 of MemOS's 17 correct answers are short-answer Facts & States or State Evolution questions. Together, these results support interpreting memory distance as a diagnostic stratification axis for long-horizon accessibility rather than a causal forgetting variable.

Table 10: Mobile accuracy (%) by memory distance for all 15 systems. The four non-empty bins contain 48, 38, 29, and 57 questions, respectively.

Method	0–2.5 h	2.5–5 h	5–7.5 h	7.5–10 h	All
SimpleStream	4.2	2.6	0.0	0.0	1.7
InfiniPot-V	4.2	5.3	10.3	0.0	4.1
StreamForest	8.3	2.6	0.0	0.0	2.9
ReKV	12.5	2.6	3.4	0.0	4.7
HERMES	8.3	5.3	3.4	0.0	4.1
FluxMem	0.0	0.0	0.0	0.0	0.0
MIRIX	14.6	10.5	10.3	5.3	9.9
M3-Agent	12.5	13.2	27.6	22.8	18.6
WorldMM	16.7	18.4	17.2	10.5	15.1
EventMemAgent	6.3	7.9	6.9	3.5	5.8
OASIS	18.8	26.3	24.1	19.3	21.5
A-MEM	14.6	18.4	20.7	1.8	12.2
Mem0	10.4	26.3	27.6	19.3	19.8
MemOS	25.0	31.6	58.6	35.1	35.5
EverMemOS	18.8	7.9	17.2	12.3	14.0

D Detailed Efficiency Analysis

Figure 5 in the main paper summarizes the principal trade-offs between Overall accuracy, ingestion RTF, and persistent storage. Here, we provide complete per-platform resource accounting together with a decomposition of text-memory preprocessing and question-answering cost.

D.1 Efficiency Metrics

For each system, we record end-to-end ingestion wall-clock time, real-time factor, ingestion prompt/input tokens, ingestion completion/output tokens, question-answering token usage, and persistent-memory size after ingestion. For systems requiring preprocessing before their native memory mechanism can operate, preprocessing cost is included in end-to-end measurements.

Because visual tokens, language-model tokens, embedding operations, and retained KV states are not equivalent computational units, token statistics should be interpreted descriptively rather than as a universal measure of computation.

D.2 Complete PC Ingestion Efficiency

We first report complete ingestion statistics on the seven PC user-day streams, which together contain 71.1 hours of device activity. Table 11 summarizes, for each memory system, the total ingestion wall-clock time, real-time factor (RTF), ingestion token usage, final persistent-memory footprint, and benchmark accuracy. For generic text-memory systems, both time and token counts include one video-to-text captioning pass so that the reported values reflect end-to-end processing from the original visual stream.

The results illustrate substantial differences in how existing memory paradigms allocate computation and storage. Several implicit visual-memory methods process the PC streams substantially faster than real time, whereas explicit semantic and text-memory systems generally incur higher ingestion cost. Importantly, fast ingestion does not necessarily imply compact persistent memory: some KV-based methods retain very large serialized states despite relatively low RTF. We analyze the preprocessing component of generic text-memory systems separately in Section D.5.

D.3 Complete Mobile Ingestion Efficiency

We next report the corresponding efficiency statistics on the three mobile user-day streams, totaling 29.0 hours. Table 12 reports all 15 systems over the same three mobile user-day streams, ensuring that latency, storage, and accuracy are computed over an identical stream set. As in the PC evalu-

Table 11: Complete ingestion-efficiency statistics on 71.1 hours of PC streams. Time and RTF for generic text-memory systems include one 10-second video-captioning pass.

Method	Time (s)	RTF	Input (M)	Output (M)	Storage	Acc.
SimpleStream	187	0.001	0	0	3.6 MB	3.2
InfiniPot-V	9,787	0.038	14.33	0	1,034.6 MB	3.8
StreamForest	5,379	0.021	186.61	0	141.9 MB	7.1
ReKV	44,998	0.176	36.86	0	2,066.7 GB	3.5
HERMES	118,611	0.463	208.87	0	7,163.6 MB	2.6
FluxMem	1,526	0.006	0.71	0	5,450.5 MB	3.8
MIRIX	102,028	0.399	87.77	2.65	844.2 MB	21.5
M3-Agent	483,785	1.890	115.97	3.46	2,432.3 MB	25.0
WorldMM	425,912	1.664	116.63	1.73	5,007.8 MB	7.4
EventMemAgent	396,471	1.549	1.50	0.12	4,978.9 MB	2.2
OASIS	1,422,249	5.557	37.03	3.92	47,069.6 MB	15.7
A-MEM	952,096	3.720	383.32	23.32	989.8 MB	21.8
Mem0	271,189	1.059	556.46	5.00	25.0 MB	27.6
MemOS	260,188	1.017	303.53	4.64	679.4 MB	29.2
EverMemOS	778,217	3.040	394.32	12.13	2,480.2 MB	29.2

Table 12: Complete ingestion-efficiency statistics on 29.0 hours of mobile streams. Generic text-memory RTF includes one shared captioning pass per independently deployed system.

Method	Time (s)	RTF	Input (M)	Output (M)	Storage	Acc.
SimpleStream	121	0.001	0	0	1.3 MB	1.7
InfiniPot-V	3,079	0.029	6.95	0	526.2 MB	4.1
StreamForest	2,913	0.028	76.17	0	61.7 MB	2.9
ReKV	20,264	0.194	15.05	0	847.5 GB	4.7
HERMES	38,142	0.365	77.42	0	2,296.9 MB	4.1
FluxMem	693	0.007	0.34	0	1,748.5 MB	0.0
MIRIX	29,902	0.286	51.23	0.73	384.9 MB	9.9
M3-Agent	50,412	0.482	46.28	1.36	863.5 MB	18.6
WorldMM	55,211	0.528	46.91	0.67	1,966.1 MB	15.1
EventMemAgent	131,969	1.263	2.07	0.24	1,818.2 MB	5.8
OASIS	466,896	4.468	28.22	1.14	16,504.0 MB	21.5
A-MEM	335,462	3.210	137.81	5.78	261.0 MB	12.2
Mem0	97,810	0.936	227.51	1.57	23.0 MB	19.8
MemOS	88,647	0.848	117.71	1.29	195.0 MB	35.5
EverMemOS	257,452	2.463	176.22	4.63	752.6 MB	14.0

ation, the reported cost of generic text-memory systems includes the shared video-captioning stage required to transform the original visual stream into textual observations.

The mobile results exhibit the same broad quality–efficiency tension observed on PC, while also highlighting platform-dependent differences in processing cost. Several visual-memory systems remain substantially faster than real time, whereas stronger semantic-memory systems require more computation. At the same time, the relative RTF of a system can differ considerably between PC and mobile streams, reflecting differences in stream composition and the number of processed observations. The platform-specific tables therefore expose cost variation that would be obscured by a single aggregated RTF.

D.4 Per-User-Day Token Workload

Aggregate token counts can obscure how memory-maintenance cost scales across user days of different duration and interaction density. Table 13 therefore normalizes measured input-token usage by stream duration for representative systems. For MemOS and Mem0, the values include the shared video-to-text captioning pass required before their native text-memory backend.

The normalized workload is nearly constant for fixed-rate pipelines such as StreamForest and M3-Agent: StreamForest stays at 2.62–2.63M input tokens/hour across all ten user days, while M3-Agent stays at 1.63M on PC and 1.59–1.60M on mobile. MIRIX shows a more variable profile: on PC it increases from 0.32M tokens/hour on UD01 to 2.23M on UD07, while the three mobile days range from 1.02M to 2.17M tokens/hour. On PC, the increase coincides with a rise from 50 to 360 sessions across user days, consistent with an activity-dependent update workload. The text-memory pipelines are more token intensive end to end because caption generation is included. Because token

Table 13: Input-token workload per stream hour (M tokens/hour) for representative systems. Text-memory values include one captioning pass.

Platform	Day	StreamForest	M3-Agent	MIRIX	MemOS	Mem0
PC	UD01	2.62	1.63	0.32	4.44	7.97
PC	UD02	2.63	1.63	0.48	4.43	8.13
PC	UD03	2.63	1.63	1.25	4.42	8.00
PC	UD04	2.63	1.63	0.69	4.44	7.94
PC	UD05	2.62	1.63	1.44	4.42	7.92
PC	UD06	2.62	1.63	2.10	4.43	7.95
PC	UD07	2.62	1.63	2.23	3.55	7.10
Mobile	UD01	2.62	1.59	1.02	4.05	7.80
Mobile	UD02	2.62	1.59	2.17	4.05	7.82
Mobile	UD03	2.63	1.60	2.07	4.06	7.89

Table 14: Decomposition of PC end-to-end ingestion time for generic text-memory systems. Caption generation is counted once per independently deployed memory system.

Method	Caption	Memory	Total	RTF
A-MEM	256,134	695,962	952,096	3.720
Mem0	256,134	15,055	271,189	1.059
MemOS	256,134	4,054	260,188	1.017
EverMemOS	256,134	522,083	778,217	3.040

units are not equivalent across architectures, these values are descriptive workload measurements; together with RTF and storage, they expose complementary scaling behavior.

D.5 Text-Memory Preprocessing Cost

The shared 10-second captioning stage contributes substantially to the end-to-end cost of generic text-memory systems. On the PC subset, caption generation consumes approximately 256,134 seconds, 298M input tokens, and 4.6M output tokens. Table 14 separates this shared preprocessing cost from each memory backend.

D.6 Question-Answering Cost

Memory formation is not the only source of computational cost: retrieval and answer generation also consume model context after the memory has been sealed. Table 15 reports the measured token usage during PC question answering.

D.7 Persistent Storage

To normalize persistent-memory footprint across streams of different duration, we additionally report the storage rate

$$R_{\text{storage}} = \frac{S_{\text{memory}}}{T_{\text{stream}}}, \tag{7}$$

where S_{memory} is the final persistent-memory footprint and T_{stream} is the total stream duration in hours.

The storage results reinforce that ingestion speed and persistent-memory size capture different aspects of system behavior. In particular, some KV-based systems process streams substantially faster

Table 15: Question-answering token cost after memory construction on PC. This table does not report QA wall-clock time. Reported averages use the sealed token records associated with the 312 PC questions.

Method	Prompt Tok.	Completion Tok.	Avg. Tok./Question
SimpleStream	802,362	35,816	2,686.5
InfiniPot-V	10,506	36,139	149.5
StreamForest	2,667,682	2,878	8,559.5
ReKV	9,882	18,471	90.9
HERMES	9,762	26,742	117.0
FluxMem	11,130	9,374	65.7
MIRIX	2,745,877	17,264	8,856.2
M3-Agent	3,488,331	232,086	11,924.4
WorldMM	1,437,051	1,267	4,610.0
EventMemAgent	458,135	24,762	1,547.7
OASIS	1,659,553	17,568	5,375.4
A-MEM	1,920,740	8,271	6,182.7
Mem0	255,842	5,973	839.2
MemOS	591,827	6,902	1,919.0
EverMemOS	725,364	9,056	2,353.9

Table 16: Persistent-memory footprint on the 71.1-hour PC subset. Storage rate divides aggregate storage by total PC stream duration.

Method	Total Storage	MB / Hour	Avg. / User Day
SimpleStream	3.6 MB	0.05	533.5 KB
InfiniPot-V	1,034.6 MB	14.55	147.8 MB
StreamForest	141.9 MB	2.00	20.3 MB
ReKV	2,066.7 GB	29,765	295.2 GB
HERMES	7,163.6 MB	100.75	1,023.4 MB
FluxMem	5,450.5 MB	76.66	778.6 MB
MIRIX	844.2 MB	11.87	120.6 MB
M3-Agent	2,432.3 MB	34.21	347.5 MB
WorldMM	5,007.8 MB	70.43	715.4 MB
EventMemAgent	4,978.9 MB	70.03	711.3 MB
OASIS	47,069.6 MB	662.02	6,724.2 MB
A-MEM	989.8 MB	13.92	141.4 MB
Mem0	25.0 MB	0.35	3.6 MB
MemOS	679.4 MB	9.56	97.1 MB
EverMemOS	2,480.2 MB	34.88	354.3 MB

than real time while producing very large serialized memory states. Together with the latency and token measurements above, these results reveal heterogeneous efficiency profiles that are not captured by any single resource metric.

E Evaluation and Scoring Details

All benchmark accuracy values are computed using deterministic answer matching rather than an LLM judge.

Multiple-choice questions. A prediction is counted as correct only when it identifies the reference option unambiguously. Option letters must occur as independent selections rather than as substrings of unrelated English words. Predictions containing multiple competing answer options are treated as incorrect.

Short-answer questions. Responses are normalized before matching the reference answer and accepted aliases. For mobile questions, whose references are predominantly Chinese, surrounding

punctuation and quotation marks are normalized before matching. The complete normalized reference answer must occur as a contiguous answer span; predicting only a strict substring of the reference answer is insufficient. If a prediction matches any annotated hard-negative answer, it is scored as incorrect even if it also matches the reference answer or an accepted alias.

Consistency. The same deterministic scoring implementation is applied to all memory systems. No method-specific post-processing or manual correction is used when computing the reported benchmark accuracy.

F LIMITATIONS

ScreenMem is intended as a first step toward evaluating proactive, query-agnostic memory over long-horizon computer-use streams, and several limitations remain. First, although the benchmark spans both PC and mobile platforms and more than 100 hours of activity, it contains a limited number of reconstructed user days and applications. The trace-grounded reconstruction strategy improves reproducibility and avoids releasing sensitive real-user screen recordings, but cannot capture the full diversity, visual complexity, and unpredictability of real-world computer use. Future versions could incorporate broader application ecosystems, more diverse user routines, and longer observation horizons.

Second, ScreenMem focuses on information that is visually observable from device interfaces and evaluates four memory capabilities with deterministic questions. This design enables reliable and reproducible scoring, but does not cover all forms of persistent memory required by practical assistants, such as information conveyed through other modalities, highly open-ended user queries, or personalized judgments whose correct answers may be inherently ambiguous.

Finally, the evaluated systems differ in their native backbones, memory representations, and input interfaces. We preserve each method's original memory mechanism whenever possible, while generic text-memory systems require a shared video-to-text preprocessing stage. Accordingly, the reported results should be interpreted as comparisons of end-to-end memory systems under a common evaluation protocol rather than as controlled comparisons of individual memory algorithms.