Memory is Not Search: Toward Proactive, Lifelong Memory in AI

Will Xiao **Sanket Deshpande** **Guha Mahesh** **Spandan Madan** **Gabriel Kreiman**
Engramme
will@engramme.com gabriel@engramme.com

Abstract

Memory is a foundational cognitive function for any artificial or biological intelligent system. Current AI approaches to long-term memory treat it as a problem largely of data storage, summarization, and search. Drawing inspiration from cognitive science and neuroscience, this position paper argues that search is insufficient for memory. Instead, memory requires distinct computational mechanisms that can proactively recall information. We analyze the specific limitations of current AI memory approaches and highlight the core requirements needed to achieve proactive, lifelong memory. Building AI systems with human-level lifelong memory will require new datasets, appropriate evaluation methods and benchmarks, and novel algorithms.

1 Introduction

Memory is inseparable from learning and integral to intelligence [1]. Any system that aspires to emulate or surpass human-like general intelligence must be able to encode and recall memories over a long horizon of experience [2]. To adapt to novel environments, interpret input contextually, and plan for the long term, biological and artificial agents need long-term memory that supports rapid recall, continual updates, multimodal data, veracity (no hallucinations), and contextual relevance.

Current AI has demonstrated enormous success across many general intelligence tasks. Frontier models memorize general knowledge by compacting large amounts of training data into their weights. However, the same models lack long-term memory after deployment. For example, a coding agent working in a specific code base must constantly (re-)remind itself, in each new conversation, about the idioms of that code base (e.g., [3, 4]).

The problem of lifelong memory in AI has been framed as continual learning (persistent updates to a model to adapt it to a new data distribution) [5–8]. When considering in-context information, memory has also been studied under the framework of test-time training (ephemeral updates for one context) [9–11]. Both continual learning and test-time training involve effectively adapting a (pre-trained) model to new test-time context(s). Thus, both frameworks also draw insights from meta-learning, i.e., learning to learn quickly in a new context [12, 10, 13]. Continual learning is an attractive conceptual framework and remains an open research area, especially in relation to frontier models. Test-time training focuses on learning in a context window, which is orders of magnitude smaller than lifelong memory such as long-horizon plans and actions, entire code bases, multimodal data, and personal life logs. Furthermore, current models do not fully utilize even their context windows, showing degraded performance when tasked with retrieving multiple pieces of information, synthesizing dispersed information, performing multi-hop retrieval, and reasoning over long contexts [14–18].

Instead, state-of-the-art methods for long-term AI memory rely on search. A typical approach is retrieval-augmented generation (RAG; [19, 20]), which involves 1) indexing documents using conventional information-retrieval methods (lexical index, (optionally learned) vector embeddings, and/or knowledge graphs) and 2) searching for related documents based on a query or prompt.

Preprint.

Another common approach is agentic search, wherein an AI agent answers the prompt by interactively using tools (e.g., web search, app connectors, file system tools, sub-agents (e.g., deep research)) to look for relevant information [21, 22].

RAG and agentic search are both based on search, which we define here as:

Definition (Search)**.** Search refers to using an explicit query, typically a short natural language query, to retrieve information from a database. Established search techniques include dense, sparse, & graph retrieval; multi-stage retrieval & reranking; and metadata filtering.

In this position paper, we argue that proactive, lifelong memory is fundamentally unlike search. We remember without explicitly querying for information. Instead, the brain spontaneously surfaces memories relevant to the current situation (the *ambience*). Search starts with a query. Either a human in the loop must provide this query, or the system must know when to search and what to search for; both require the querying agent to already have some memory in the first place. To build AI systems that can emulate and even augment human memory, a paradigm other than search is needed. Memory systems must incorporate the ambience to automatically recall memories from lifelong data, which can be multimodal and up to petabytes in size (Section A). To make progress toward developing novel architectures for memory, it is critical to build new and more appropriate benchmarks and evaluation methods.

Below, we explicate the properties of lifelong memory (the *memorome*; Section 2), survey current AI memory algorithms and their limitations (Sections 3 & 4), articulate memory as proactive recall (Section 5), identify key gaps in current AI memory benchmarks and architectures and suggest ways to address them (Section 6), and discuss alternative views (Section 7).

2 The Memorome: Lifelong, Dynamic Data up to Multimodal Petabytes

Memory is a broad concept that encompasses working memory, short-term memory, long-term memory, and more. Here, we focus on the *lifelong* memory of a biological or artificial agent. We define the **memorome**: Analogous to the genome—the entirety of a person's genetic information—the memorome is the entire collection of *memories* from an individual lifetime. The memorome includes conversations, pictures, videos, emails, documents, and more. Thus, the memorome is naturally multimodal. In practice, we focus on the digital memorome, excluding touch, smell, etc., as well as never-recorded memories. The memorome is closely allied with the concept of a life log [23–25].

The memorome is individual-specific and excludes *world knowledge*. If an individual never knew who the prime minister of Cambodia is, this information is not part of that individual's memorome, even though this information is easily accessible on the internet. Conversely, foundational models, by definition trained on general-purpose data, lack individual-specific information, such as a particular conversation instance. The general-purpose knowledge stored in foundation models is distinct from the contents of an individual's memorome.

The size of a memorome varies from individual to individual. We estimate a typical person's memorome to be potentially up to petabytes of data (Appendix A). Even individuals with a small digital footprint have memoromes orders of magnitude larger than the context windows of current foundation models.

A person's memorome spans decades, in contrast with the lifetimes of current AI agents, which typically live from seconds to hours. Future agents will likely live longer and process more data in their lifetime, heightening the need for a memory system that can scale to large memoromes.

The memorome is dynamic and continually updated. Dynamic changes involve not only adding new knowledge and updating stale beliefs, but also reinterpreting and synthesizing accumulated information.

The vast size, time span, and dynamic nature of lifelong memory lead to challenges qualitatively different from typical concerns of current AI memory algorithms, which we review next.

3 Current Approaches to AI Memory Require Search to Scale

3.1 RAG and Agentic Search Attempt to Use Search to Scale to Large Memoromes

Current algorithms for AI memory require search to scale to life-sized memoromes. Consistent with the definition in Section 1, we use 'search-based memory' broadly to refer to retrieval using a query either written by a human or generated by an agent. Practical search-based AI memory algorithms fall into two main clusters: RAG and agentic search.

Retrieval-Augmented Generation (RAG). RAG uses conventional search methods to retrieve information from a database [19, 20, 26, 27]. Retrieval methods include semantic (dense) search, keyword (sparse) search, hybrid search, knowledge graph-based retrieval, and metadata filtering. To improve search efficiency, RAG indexes the database offline before processing the query, limiting the possible interactions between the query and memories; this is only partially mitigated by multi-stage and multi-hop retrieval. The RAG retrieval function can be learned [20, 28–30]. Practical RAG offerings are usually not adapted to the context of an individual user, further limiting their relevance.

Agentic search-based memory. Agentic search-based memory systems mimic how people organize and search for information using tools by using an LLM or an associated controller to decide what to store, summarize, link, retrieve, and update. Agentic search is often combined with a set of notes or a file-system-like organization of memories maintained by the agent [31–33, 3, 4]. Agentic search is costlier and slower than RAG. At the start of each session, the agent has no user-specific information except what is automatically injected into context or later retrieved. Thus, as the scale of the memorome increases, the agent is likely to encounter bottlenecks in how much context can be retrieved, incorporated, and used to dynamically steer the search.

3.2 Current Neural Memory Architectures Do Not Scale to Lifelong Memory

Research has also approached the memory problem with more tightly integrated neural mechanisms. To date, these mechanisms have not been demonstrated or designed to scale to lifelong memories.

Model fine-tuning. Foundation models can be fine-tuned on personal data [34, 35]; a prominent example is digital cloning [36]. Digital clones usually try to mimic the style and behavior of the cloned person. Digital clones do not explicitly solve memory, although they implicitly memorize some information about the cloned person. Model fine-tuning faces challenges in catastrophic forgetting [37] (losing, e.g., instruction following and general knowledge) and hallucinating memories.

In-context memory. The context window can be viewed as test-time memory [38–40], and neural architectures have been proposed to efficiently process long contexts [41, 42, 11, 43, 12, 44–46]. At the scale of current context windows of around 1 M tokens, memory performance remains inadequate when the context is long [14–18]. Further, in-context memory is usually considered ephemeral test-time learning rather than persistent continual learning. As such, this direction of research has not addressed scaling to lifelong memory. Long-context abilities require dedicated training [47, 48, 15], which can also be expensive to scale. Thus, it is unclear whether current approaches to in-context memory can extend to the petabyte scale of human lifelong memory.

4 Limitations of Search-Based Approaches to AI Memory

RAG and agentic search, the two main current approaches to supplement AI with long-term memory, are practical, but they are insufficient to serve as models of proactive memory.

Search requires a query or prompt. RAG and agentic search start with a search query or prompt. Formulating an effective query requires already knowing something about what to search for. This need for foreknowledge poses a bootstrapping problem. Ultimately, the user (or an agent suggesting a query) is left with the key function of formulating a query or prompt. Proactive memory recall adds the challenge of knowing when to search, i.e., whether there is useful information to search for, another piece of information that requires pre-existing memory. The brain spontaneously surfaces useful memories or suggests the need to search for information. Memories that are not spontaneously

surfaced or suggested become irretrievable, which we refer to as the *dark matter of memory*—the memory content still exists, but it can only be triggered serendipitously, with the right cues. Search-based memory systems that rely on the user or an agent to originate a query have no natural mechanism for surfacing this dark matter and will systematically under-access it.

Search struggles to perform associative recall. Search similarity metrics, whether lexical or semantic, are primarily built on culturally shared meaning that does not reflect an individual's idiosyncratic associations. In contrast, much of human memory is associative, linking together pieces of information that share no surface-level similarity—no lexical overlap, no culturally shared semantic relevance, and no entity overlap. For example, if someone once heard a piece of shocking news, they can remember the details of where they were, what they were doing, etc., even though the news and the remembered details have no other culturally shared semantic linkage (flashbulb memory; [49]). Modeling associative recall requires a context-specific, i.e., personalized, model of relevance.

5 Properties of Proactive, Lifelong Memory

Several key properties of human memory can instruct the design of proactive, lifelong AI memory.

Memory is proactive. The human brain surfaces memories spontaneously without external prompting. When someone meets an acquaintance, they (ideally) automatically recall the acquaintance's name, relationship to them, recent interactions, and so on. When an author discusses related work, they (ideally) automatically remember the key papers. People search for information, but only with effort, and we search infrequently compared with how often we proactively remember. Similarly, AI memory should automatically retrieve information instead of chasing information.

While an AI agent can search proactively, coming up with personally relevant queries requires the AI agent to know what information can be looked for and how to look for it, which in turn requires memory. As discussed in Section 4, most memories are lost as dark matter, i.e., information that we once knew and that we could potentially access, but that we are no longer aware of. By definition, memory dark matter is not searchable via easy-to-identify queries. Rather, accessing the dark matter requires a proactive memory system that can infer, from the ambience, cues with personal relevance.

Memories are recalled in real time. The brain recalls memories alongside cognition in real time. This fast memory recall is necessary in many situations. A lawyer conducting a deposition, a surgeon facing a complication, or a politician debating an opponent cannot afford minutes formulating a search prompt, waiting for search results, digesting the hits, and only then continuing their thinking. For a practical AI memory system, a search loop, such as agentic search, that takes tens of seconds is inefficient and even impractical for proactively retrieving memories in real time.

Memories are lifelong. Human memory spans decades. Years after last meeting an old friend, people can often still readily remember what they know about the friend. In addition to the long time span, lifelong memories are large, up to petabytes of data. As AI agents become increasingly longer-lived, they will also need increasingly longer-horizon memory. Besides challenging the in-context capabilities of models (as discussed in Sections 3 & 4), lifelong memory also introduces unique challenges in terms of synthesizing dispersed information and understanding dynamic trajectories across time scales.

Memory involves a specific form of compression. Memory is not lossless storage, nor is memory a summarization. Rather, memories retain rich, abstract, and often-idiosyncratically specific aspects of the original experience. When someone meets an old friend decades later, they may not remember what their friend wore the last time they met, but they may still remember the restaurant they met at. If they revisit the restaurant, they may even recognize the table they sat at. What human memory retains and recalls is highly dependent on a person's existing knowledge and context.

Table 1: Popular memory benchmarks are insufficient to distinguish general model capabilities represented by agentic search and even in-context memory from specialized memory systems. Uncertainty for our baselines is reported as standard error across question-answer pairs after averaging the five judge passes for each item.

	LoCoMo* [55]	LongMemEval [56]		BEAM† [57]	
		S	M†	1M	10M
In-context	0.977 ± 0.004	0.81 ± 0.02	0.54 ± 0.07	0.64 ± 0.02	–
Agentic	0.985 ± 0.003	0.90 ± 0.01	0.89 ± 0.04	0.71 ± 0.02	0.64 ± 0.03
Hindsight[a]	0.920	**0.946**	–	**0.739**	**0.641**
Mem0[b]	0.916	0.934	–	0.641	0.486
Honcho[c]	0.899	0.926	**0.888**†	0.631	0.406
EverMemOS[d]	**0.931**	0.830	–	–	–
Zep[e]	0.852‡	0.712	–	–	–

*LoCoMo performance for specialized memory systems corresponds to 1,540 of 1,986 total LoCoMo questions, excluding so-called adversarial questions. We identified and removed additional questions and labels that were wrong, vague, or not specific to memory, resulting in 1,334 questions. Hence, our LoCoMo performance ceiling is higher. See details in Appendix B. †A subset of questions were evaluated. See details in Appendix B. ‡Zep's official blog post reports 0.80 [60]; 0.85 is the highest public value we found [61]. BEAM performance reports mean scores on a rubric ('nugget evaluation'; [57, 62]). Sources: [a]Hindsight AMB benchmark page and BEAM post [63, 64]; [b]Mem0 memory-evaluation docs [65]; [c]Honcho benchmarking post [66]; [d]EverMemOS/EverMind blog and GitHub [67]; [e]MemoryLake benchmark repo, Zep paper, and Zep blog [60, 61, 68].

6 The Way Forward: New Datasets, Benchmarks, and Architectures

To build AI with lifelong, proactive memory, a new paradigm of datasets, benchmarks, and architectures is needed to represent and handle large memoromes, realistic data, and proactive recall.

6.1 Benchmarks Need Larger, Realistic Memoromes and Need to Test Proactive Recall

In computer vision, the MNIST dataset [50] helped develop initial algorithms, while later progress relied on the transformative role of ImageNet [51]. Analogously, while existing memory benchmarks provide useful guidance, progress toward real-world memory systems requires more natural, more complex, and much larger datasets.

Benchmarks should test much larger memoromes relevant to lifelong memory. Current benchmarks for AI memory test $O(1\text{ M})$ tokens of context. These memory benchmarks form two clusters. One stems from testing long-context models and includes needle in a haystack [52], multiple needles, multi-hop retrieval on a graph [53], long-context instruction following, and long-context reasoning [18, 54]. Although some of these benchmarks are programmatically extensible, they are typically tested up to 1 M tokens, informed by the context capacity of current models.

The other cluster originates in testing memory for chat assistants and AI agents, including LoCoMo [55], LongMemEval [56], BEAM [57], MEMTRACK [58], MemoryAgentBench [59], and others. These datasets test memory sizes from 10 K tokens to over 1 M tokens, with the recent BEAM dataset extending up to 10 M tokens of context. The sizes of agentic memory benchmarks have increased over the years in concert with increasing model context windows.

Compared to long-context benchmarks, agentic memory benchmarks emphasize temporal grounding, incremental data updates, and a memory persistent across queries and tasks, all properties relevant to lifelong memory. However, the context sizes of current agentic memory benchmarks significantly overlap the context-window sizes of current LLMs. Indeed, in-context memory is a strong baseline on popular agentic memory benchmarks (Table 1; Figure 1). Further, agentic search without tuning nearly matches the performance of specialized memory systems. Some of the top performances reported by specialized memory systems used frontier models (Hindsight and Honcho used Gemini 3(.1) Pro; see full details in Appendix B.1), while others used efficiency-oriented models. Thus, while the benchmarks may be somewhat more discriminative if all existing memory systems employed full models, available results indicate that current benchmarks—in part due to the limited memory size tested—are insufficient for evaluating improvements in frontier long-term memory abilities.

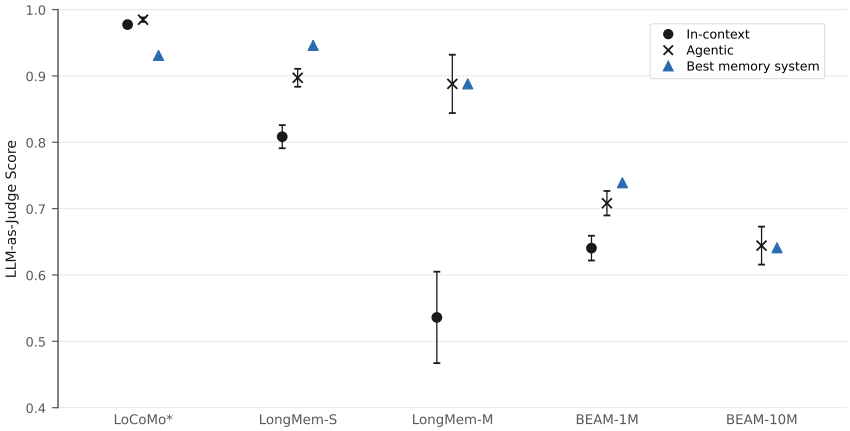

Figure 1: **Current memory benchmarks insufficiently distinguish general model capabilities represented by in-context and agentic-search baselines from specialized memory systems.** The data correspond to Table 1. The in-context prompt was minimal and not optimized (prompts shown in Appendix D). BEAM-10M was not tested in context due to context-length limits. Error bars indicate standard error across question-answer pairs. For LoCoMo*, see notes in Table 1. Across benchmarks, agentic search and, in one case, in-context memory already match or exceed public specialized memory systems.

Lifelong memory is orders of magnitude larger than the millions of tokens tested in existing memory benchmarks. For example, knowledge workers can exchange 30–50 M tokens per year of email alone. An AI-assisted coder today orchestrates millions of tokens per day, or $O(1\text{ B})$ tokens per year; an organizational context is even larger. An individual's memorome can extend to petabytes of multimodal data (Section 2). Thus, benchmarks relevant to lifelong memory need to test at least 100 M to 1 B tokens *per memory context* (e.g., one user or one timeline).

Memory datasets need more realistic statistics. Existing memory benchmarks are largely built from synthetic data. While controlled synthetic datasets like MRCR (multi-round co-reference resolution; [69, 70]) are valuable for answering specific questions, synthetic data abstract away important aspects of real data. For example, real data show small-world networks of co-occurring entities (e.g., social networks in the case of people as entities). Real data cover more diverse topics, have messier content (typos, ambiguities, and complex associations), and combine heterogeneous sources. Topics in real data are fluid, blending into each other and evolving over time (e.g., a research idea may spawn several projects over years and evolve all the while). These differences limit the transferability of results from synthetic benchmarks to real-world applications.

To quantify how existing memory benchmark data diverge from data in the wild, we compared descriptive statistics between real personal memoromes and agentic memory benchmark data (methods detailed in Appendix C). In memory benchmarks, each independent context (e.g., a conversation) constitutes a memorome. Real data show a higher proportion of unique people and people mentions, have more diverse and higher-dimensional semantic topics, show less topical clustering, contain small-world co-occurrence graphs, and use longer-tailed vocabulary (Figure 2). These qualitative differences are robust despite significant variations across differently constructed synthetic datasets on the one hand, and heterogeneous data from different people on the other hand. The complexity of real data presents both challenges for search-based memory and opportunities for developing better models of memory.

Real data, or synthetic data that better match real-data statistics, are needed to build memory benchmarks that can guide translatable model improvements. Because personal memory data, such as life logs, contain inherently sensitive information, collecting datasets of real memoromes at scale will be challenging, although there are laudable efforts in this direction (e.g., [25]). Simulated societies of interacting agents have been used to study AI memory (e.g., [31, 71]) and could generate data with

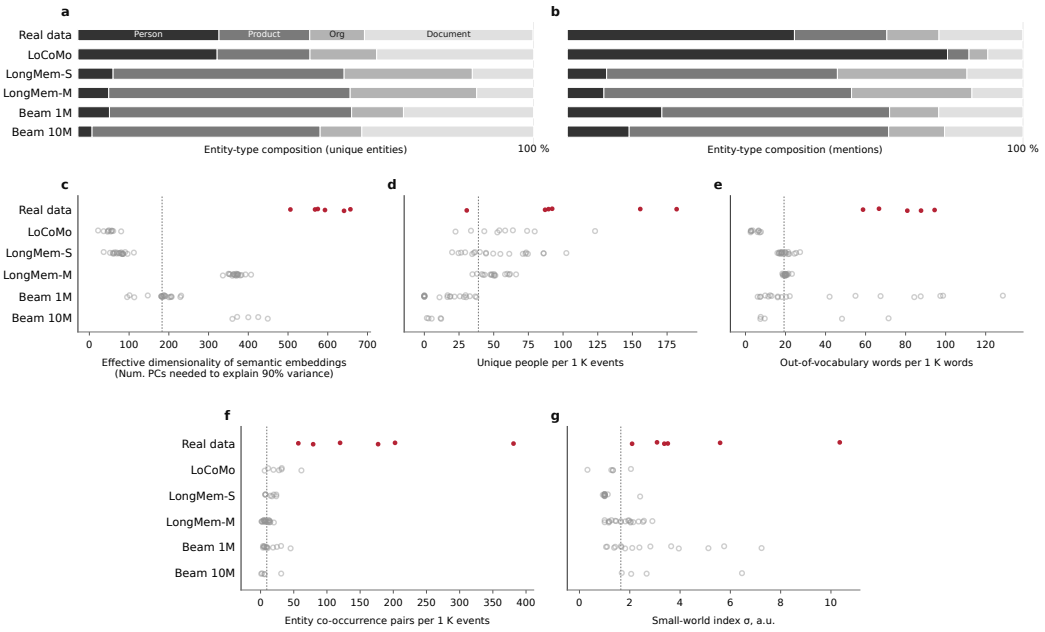

Figure 2: **Real memorome data differ statistically from current synthetic memory benchmarks**. **a, b**, Proportions of entities in each category (person, product, organization (org), and document), separately for unique entities in each memorome (**a)** and entity mentions (**b**). **c**, Effective dimensionality of semantic embeddings of events extracted from each memorome, quantified by the number of principal components needed to explain 90% of variance. **d**, Unique people per 1 K extracted events. **e**, Number of out-of-vocabulary words per 1 K words. **f**, Number of entity pairs occurring in the same event per 1 K extracted events. **g**, Index σ [72] quantifying the small-worldness of entity co-occurrence graphs. In panels **c–g**, each dot corresponds to a memorome, either a personal corpus (real data) or a complete, independent history (synthetic datasets; see Appendix C for details). The dashed vertical line indicates the median across synthetic datasets. Real memoromes differ from synthetic data in entity composition, topic dimensionality, entity & lexical variety, and co-occurrence graph structure.

more realistic statistics than synthetic conversations. However, data derived from agents with limited memory may be fundamentally less coherent than real data.

Benchmarks need to test proactive memory recall. Memory benchmarks almost exclusively test question answering. In contrast, proactive memory does not answer explicit questions but rather surfaces information based on the current holistic context (ambience) to inform the user's next action (Section 5). To test proactive memory recall, benchmarks need to include dynamic contexts as a model input, and evaluations should assess proactively surfaced memories for timeliness, informativeness, abstention, and non-redundancy. The evaluation should be relative: Can a future action be better predicted (supervised labels), or a better future action taken (reward function), with a surfaced memory vs. without it?

Additionally, benchmarks should address actual memory needs instead of artificially generated queries. More work (e.g., [73, 74]) is needed to quantify the actual memory needs of people and AI agents in realistic situations.

6.2 New Models and Architectures Are Required for Proactive, Lifelong Memory

Models need several new abilities beyond current AI memory architectures (Section 3). First, models need to be able to incorporate memoromes (i.e., test-time data) at least as large as 100 M–1 B tokens. Context sizes at this scale are beyond reach for current linear attention, test-time memory, and neural memory approaches. Ultimately, because petabyte-scale lifelong memoromes rival the scale of pretraining data, a promising approach may be to represent memory implicitly in model weights.

If so, however, a key challenge is how to efficiently adapt the model weights to a new memorome without cost-prohibitive (pre)training or catastrophic forgetting of general abilities.

Second, models should recall memories contextually given a streaming ambience without requiring explicit queries. Ambient memory recall refers to

$$\text{Recall}_\theta(M_t, \boldsymbol{x}_{(t-\tau):t}) \mapsto \{m_{t1}, m_{t2}, ..., m_{tn}\}, n \geq 0,$$

where $\boldsymbol{x}_{(t-\tau):t}$ is a (multimodal) ambience time series, M_t is a user-specific, continually updated memory corpus, and m_t is a piece of information grounded in M_t. For illustration, M_t could be a set of valid memories, and $m_t \in M_t$; in general, a grounding model $G(m_{ti}, E_{ti}, M_t) \mapsto \{0, 1\}$ can be used to determine whether memory m_{ti} is grounded in evidence $E_{ti} \subseteq M_t$. Early and recent examples of ambient memory recall systems include [75–82].

The need for contextual recall limits the direct applicability of current search-based memory systems optimized for question answering. The memory model will ideally input multimodal context including vision, audition, location, time, company, etc., all of which are vital to human memory. In contrast, RAG, agentic search, and long-context models focus on content (occasionally with a thin layer of context represented by metadata). Models, like human memory, should be informed by context at multiple time scales.

As models are developed that can represent lifelong memory, it will also be vital to ensure controllability and the right to forget. Users should be able to ask the model to 'forget' specific information. The right to forget will be an especially important algorithmic consideration for implicit (neural) memory architectures. A particularly interesting combination is to use implicit memory for in-context-scale recent recall and an external evidence store for provenance, grounding verification, and user control.

To advance algorithms for proactive, lifelong memory, we first need appropriate benchmarks to measure progress. These benchmarks will also allow for testing our position against alternative views, detailed in the next section.

7 Falsifiability and Alternative Views

Our position makes testable predictions that contrast with alternative views. The first alternative to our position is: **Search can solve (proactive, lifelong) memory.** In principle, a model can be built that predicts a verbal or latent query based on the input context. The predicted query can then be passed into a question-answering system employing, e.g., RAG or agentic search. Such a system can be made proactive by searching continually, perhaps gated on the context input and the quality of the memory output [31, 71]. The utility of search engines demonstrates that there is no fundamental difficulty in scaling information retrieval to large-scale data. The search-based alternative is thus a feasible baseline deserving rigorous comparison. There could be a continuum of alternatives, including personalized search by fine-tuning embedding models, rerankers, generative retrieval, and retrieval using queries in a learned latent space [83, 84]. Our position predicts that a search-based alternative introduces bottlenecks limiting the quality of queries, the search relevance function, and proactivity. Potential limitations include cases involving vague queries, like tip-of-tongue questions; synthesis of information sparsely distributed over many documents; and inefficiency in determining when to search and what to search for. Circumventing these potential bottlenecks calls for an integrated model that internalizes a memorome to directly map ambience to proactive memories, without explicit search.

A second alternative view is: **Current AI architectures can scale to lifelong memories**. This alternative view does not necessarily contradict our position that memory is not search, although this view contradicts some of our deductions. Existing work that scales in-context memory uses fixed-size memory representations that are either recurrently accumulated (a latent state matrix (e.g., [44, 85]) or test-time learned (model weights for neural/soft memory; e.g., [12, 46]). In the limit of very large memoromes, finite-sized memories will approximately forget everything, and lifelong memoromes are significantly larger than the context sizes considered in current test-time memory work. Moreover, (meta-)learning to memorize (learn) at test time could require a combinatorial number of training examples for a model to learn to use all parts of the context. Instead, our position is that fundamental architecture changes are needed to add extensible memory capacity, inductive biases for effective generalization to very long contexts, and more efficient test-time learning to feasibly incorporate lifelong memoromes.

A third alternative view concerns our position on the need for a new kind of datasets (Section 6.1). Alternatively: **Existing approaches suffice to construct memory datasets useful for training and testing proactive, lifelong memory**. This alternative view holds that scaling methods similar to those that constructed MRCR [70] or BEAM [57] can create the necessary datasets. In contrast, our position is that real data, or significantly more realism in synthetic data starting with the statistics we analyzed, are required to evaluate and learn proactive, lifelong memory recall. In particular, the shapes of input data and output behavior need to be fundamentally different and oriented toward ambient intelligence.

8 Conclusion

Advancing AI memory requires a shift away from search-centric paradigms toward proactive, lifelong memory. This shift necessitates new datasets, evaluation methods, and architectures.

References

[1] Eric R. Kandel, Irving Kupfermann, and Susan Iversen. Learning and memory. In Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, editors, *Principles of Neural Science*. McGraw-Hill, 4th edition, 2000.

[2] Dan Hendrycks, Dawn Song, Christian Szegedy, Honglak Lee, Yarin Gal, Erik Brynjolfsson, Sharon Li, Andy Zou, Lionel Levine, Bo Han, Jie Fu, Ziwei Liu, Jinwoo Shin, Kimin Lee, Mantas Mazeika, Long Phan, George Ingebretsen, Adam Khoja, Cihang Xie, Olawale Salaudeen, Matthias Hein, Kevin Zhao, Alexander Pan, David Duvenaud, Bo Li, Steve Omohundro, Gabriel Alfour, Max Tegmark, Kevin McGrew, Gary Marcus, Jaan Tallinn, Eric Schmidt, and Yoshua Bengio. A Definition of AGI. *arXiv preprint arXiv:2510.18212*, 2025. URL `http://arxiv.org/abs/2510.18212`.

[3] Anthropic. How claude remembers your project. Webpage, 2026. URL `https://code.claude.com/docs/en/memory`. Accessed: 2026-05-04.

[4] OpenAI. Custom instructions with agents.md. Webpage, 2026. URL `https://developers.openai.com/codex/guides/agents-md`. Accessed: 2026-05-04.

[5] Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned Language Models are Continual Learners. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.410. URL `https://aclanthology.org/2022.emnlp-main.410/`.

[6] Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards Lifelong Learning of Large Language Models: A Survey. *ACM Comput. Surv.*, 57(8), March 2025. ISSN 0360-0300. doi: 10.1145/3716629. URL `https://doi.org/10.1145/3716629`.

[7] Qisheng Hu, Quanyu Long, and Wenya Wang. When continual learning moves to memory: A study of experience reuse in llm agents, 2026. URL `https://arxiv.org/abs/2604.27003`.

[8] Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H Chi, et al. Evo-Memory: Benchmarking LLM Agent Test-time Learning with Self-Evolving Memory. *arXiv preprint arXiv:2511.20857*, 2025.

[9] Ke Alexander Wang, Jiaxin Shi, and Emily B Fox. Test-time regression: a unifying framework for designing sequence models with associative memory. *arXiv preprint arXiv:2501.12352*, 2025.

[10] Jianyu Zhang, Niklas Nolte, Ranajoy Sadhukhan, Beidi Chen, and Léon Bottou. Memory mosaics. *arXiv preprint arXiv:2405.06394*, 2024.

[11] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers Are Secretly Fast Weight Programmers. *arXiv preprint arXiv:2102.11174*, June 2021.

[12] Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

[13] Yiming Xiong, Shengran Hu, and Jeff Clune. Learning to continually learn via meta-learning agentic memory designs, 2026. URL `https://arxiv.org/abs/2602.07755`.

[14] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, Krishna Haridasan, Ahmed Omran, Nikunj Saunshi, Dara Bahri, Gaurav Mishra, Eric Chu, Toby Boyd, Brad Hekman, Aaron Parisi, Chaoyi Zhang, Kornraphop Kawintiranon, Tania Bedrax-Weiss, Oliver Wang, Ya Xu, Ollie Purkiss, Uri Mendlovic, Ilaï Deutel, Nam Nguyen, Adam Langley, Flip Korn, Lucia Rossazza, Alexandre Ramé, Sagar Waghmare, Helen Miller, Nathan Byrd, Ashrith Sheshan, Raia Hadsell, Sangnie Bhardwaj, Pawel Janus, Tero Rissa, Dan Horgan, Alvin Abdagic, Lior Belenki, James Allingham, Anima Singh, Theo Guidroz, Srivatsan Srinivasan, Herman Schmit, Kristen Chiafullo, Andre Elisseeff, Nilpa Jha, Prateek Kolhar, Leonard Berrada, Frank Ding, Xiance Si, Shrestha Basu Mallick, Franz Och, Sofia Erell, Eric Ni, Tejasi Latkar, Sherry Yang, Petar Sirkovic, Ziqiang Feng, Robert Leland, Rachel Hornung, Gang Wu, Charles Blundell, Hamidreza Alvari, Po-Sen Huang, Cathy Yip, Sanja Deur, Li Liu, Gabriela Surita, Pablo Duque, Dima Damen, Johnson Jia, Arthur Guez, Markus Mircea, Animesh Sinha, Alberto Magni, Paweł Stradomski, Tal Marian, Vlado Galić, Wenhu Chen, Hisham Husain, Achintya Singhal, Dominik Grewe, François-Xavier Aubet, Shuang Song, Lorenzo Blanco, Leland Rechis, and 3326 additional authors. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities, December 2025. URL `https://arxiv.org/abs/2507.06261`.

[15] Weizhou Shen, Ziyi Yang, Chenliang Li, Zhiyuan Lu, Miao Peng, Huashan Sun, Yingcheng Shi, Shengyi Liao, Shaopeng Lai, Bo Zhang, et al. QwenLong-L1.5: Post-Training Recipe for Long-Context Reasoning and Memory Management. *arXiv preprint arXiv:2512.12967*, 2025.

[16] OpenAI. Introducing GPT-5.4, March 2026. URL `https://openai.com/index/introducing-gpt-5-4/`. Accessed: 2026-05-04.

[17] OpenAI. Introducing GPT-5.5, April 2026. URL `https://openai.com/index/introducing-gpt-5-5/`. Accessed: 2026-05-04.

[18] Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024.

[19] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[20] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR, 2020.

[21] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[22] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[23] Jim Gemmell, Roger Lueder, and Gordon Bell. The mylifebits lifetime store. In *Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence*, pages 80–83, 2003.

[24] Morgan Harvey, Marc Langheinrich, and Geoff Ward. Remembering through lifelogging: A survey of human memory augmentation. *Pervasive and Mobile Computing*, 27:14–26, 2016.

[25] Cathal Gurrin, Liting Zhou, Graham Healy, Björn Þór Jónsson, Duc-Tien Dang-Nguyen, Jakub Lokoć, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Klaus Schöffmann. Introduction to the Fifth Annual Lifelog Search Challenge, LSC'22. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, ICMR '22, pages 685–687, New York, NY, USA, June 2022. Association for Computing Machinery. ISBN 978-1-4503-9238-9. doi: 10.1145/3512527.3531439.

[26] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 59532–59569. Curran Associates, Inc., 2024. doi: 10.52202/079017-1902. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf`.

[27] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

[28] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL `https://aclanthology.org/2020.emnlp-main.550/`.

[29] Omar Khattab and Matei Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401075. URL `https://doi.org/10.1145/3397271.3401075`.

[30] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[31] Joon Sung Park, Joseph C. O'Brien, Carrie J. 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*, UIST '23, pages 1–22. Association for Computing Machinery, 2023. doi: 10.1145/3586183.3606763.

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

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

[34] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.399. URL `https://aclanthology.org/2024.acl-long.399/`.

[35] Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6476–6491, 2024.

[36] Gordon Bell and Jim Gray. Digital immortality. *Communications of the ACM*, 44(3):28–31, 2001.

[37] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 128–135, 1999. doi: 10.1016/S1364-6613(99)01294-2.

[38] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[39] Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[40] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2021.

[41] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2978–2988, 2019.

[42] Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

[43] Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091, 2022.

[44] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[45] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[46] Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. ATLAS: Learning to Optimally Memorize the Context at Test Time. *arXiv preprint arXiv:2505.23735*, 2025.

[47] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1M Technical Report, 2025. URL `https://arxiv.org/abs/2501.15383`.

[48] Chejian Xu, Wei Ping, Peng Xu, Zihan Liu, Boxin Wang, Mohammad Shoeybi, Bo Li, and Bryan Catanzaro. From 128K to 4M: Efficient Training of Ultra-Long Context Large Language Models. *arXiv preprint arXiv:2504.06214*, 2025.

[49] Roger Brown and James Kulik. Flashbulb memories. *Cognition*, 5(1):73–99, 1977. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(77)90018-X. URL `https://www.sciencedirect.com/science/article/pii/001002777790018X`.

[50] Yann LeCun. The mnist database of handwritten digits, 1998.

[51] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[52] Greg Kamradt. Needle in a haystack - pressure testing llms. GitHub repository, 2023. URL `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`.

[53] OpenAI. Graphwalks: A multi-hop reasoning long context benchmark. Hugging Face dataset, 2025. URL `https://huggingface.co/datasets/openai/graphwalks`.

[54] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, 2024.

[55] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. [LoCoMo] Evaluating Very Long-Term Conversational Memory of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747.

[56] 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.

[57] Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J Ross Mitchell. Beyond a Million Tokens: Benchmarking and Enhancing Long-Term Memory in LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=y59hf5lrMn`.

[58] Darshan Deshpande, Varun Gangal, Hersh Mehta, Anand Kannappan, Rebecca Qian, and Peng Wang. MEMTRACK: Evaluating Long-Term Memory and State Tracking in Multi-Platform Dynamic Agent Environments. *arXiv preprint arXiv:2510.01353*, 2025.

[59] Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. *arXiv preprint arXiv:2507.05257*, 2025. doi: 10.48550/arXiv.2507.05257. URL `http://arxiv.org/abs/2507.05257`.

[60] Daniel Chalef. The retrieval tradeoff: What 50 experiments taught us about context engineering, December 2025. URL `https://blog.getzep.com/the-retrieval-tradeoff-what-50-experiments-taught-us-about-context-engineering/`. Accessed: 2026-05-05.

[61] MemoryLake AI. LoCoMo benchmark — MemoryLake. GitHub repository, n.d. URL https://github.com/memorylake-ai/memorylake-locomo-benchmark. Accessed: 2026-05-05.

[62] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework. *arXiv preprint arXiv:2411.09607*, 2024.

[63] Nicolò Boschi and Hindsight Team. Agent memory benchmark: A manifesto, March 2026. URL https://hindsight.vectorize.io/blog/2026/03/23/agent-memory-benchmark. Accessed: 2026-05-05.

[64] Ben Bartholomew and Hindsight Team. Hindsight is #1 on BEAM — the benchmark that tests memory at 10 million tokens, April 2026. URL https://hindsight.vectorize.io/blog/2026/04/02/beam-sota. Accessed: 2026-05-05.

[65] Mem0. Memory evaluation, 2026. URL https://docs.mem0.ai/core-concepts/memory-evaluation. Accessed: 2026-05-05.

[66] Ben McCormick and Courtland Leer. Benchmarking Honcho, December 2025. URL https://blog.plasticlabs.ai/research/Benchmarking-Honcho. Accessed: 2026-05-05.

[67] EverMind researchers. EverMemOS: SOTA results across four memory benchmarks and what it means for LLM agents, January 2026. URL https://evermind.ai/blogs/everos-sota-results-across-four-memory-benchmarks-and-what-it-means-for-llm-agents. Accessed: 2026-05-05.

[68] Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, January 2025. URL https://arxiv.org/abs/2501.13956. Accessed: 2026-05-05.

[69] Kiran Vodrahalli, Santiago Ontañón, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, Rohan Anil, Ethan Dyer, Siamak Shakeri, Roopali Vij, Harsh Mehta, Vinay Ramasesh, Quoc Le, Ed Chi, Yifeng Lu, Orhan Firat, Angeliki Lazaridou, Jean-Baptiste Lespiau, Nithya Attaluri, and Kate Olszewska. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *arXiv preprint arXiv:2409.12640*, 2024.

[70] OpenAI. OpenAI-MRCR v2: Multi-Round Co-reference Resolution Long-Context Evaluation. Hugging Face dataset and OpenAI model release documentation, 2025. URL https://huggingface.co/datasets/openai/mrcr. Accessed: 2026-05-06.

[71] Chuanyang Hong and Qingyun He. Enhancing memory retrieval in generative agents through llm-trained cross attention networks. *Frontiers in Psychology*, 16:1591618, 2025.

[72] Mark D Humphries and Kevin Gurney. Network 'small-world-ness': a quantitative method for determining canonical network equivalence. *PloS one*, 3(4):e0002051, 2008.

[73] Engramme. What do people need to remember?, March 2026. URL https://www.engramme.com/index/what-do-people-need-to-remember. Accessed: 2026-05-05.

[74] Margery Eldridge, Abigail Sellen, and Debra Bekerian. Memory problems at work: Their range, frequency and severity. *Rank Xerox, EuroPARC*, 1992.

[75] Bradley J. Rhodes and Thad Starner. Remembrance agent: A continuously running automated information retrieval system. In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pages 487–495, 1996. URL https://www.bradleyrhodes.com/Papers/remembrance.html. Accessed 2026-05-06.

[76] Bradley J. Rhodes. The Wearable Remembrance Agent: A System for Augmented Memory. *Personal Technologies*, 1:218–224, 1997. URL https://www.bradleyrhodes.com/Papers/wear-ra-personaltech/. Accessed 2026-05-06.

[77] Bradley James Rhodes. *Just-in-time information retrieval*. PhD thesis, Massachusetts Institute of Technology, 2000.

[78] Bradley J Rhodes and Pattie Maes. Just-in-time information retrieval agents. *IBM Systems journal*, 39 (3.4):685–704, 2000.

[79] Wazeer Deen Zulfikar, Samantha Chan, and Pattie Maes. Memoro: Using Large Language Models to Realize a Concise Interface for Real-Time Memory Augmentation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703300. doi: 10.1145/3613904.3642450. URL https://doi.org/10.1145/3613904.3642450.

[80] Indrajeet Ghosh, Kasthuri Jayarajah, Nicholas Waytowich, and Nirmalya Roy. Augmenting Personalized Memory via Practical Multimodal Wearable Sensing in Visual Search and Wayfinding Navigation. In *Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '25, page 11–21, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713132. doi: 10.1145/3699682.3728340. URL https://doi.org/10.1145/3699682.3728340.

[81] Kevin Pu, Ting Zhang, Naveen Sendhilnathan, Sebastian Freitag, Raj Sodhi, and Tanya R. Jonker. ProMemAssist: Exploring Timely Proactive Assistance Through Working Memory Modeling in Multi-Modal Wearable Devices. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720376. doi: 10.1145/3746059.3747770. URL https://doi.org/10.1145/3746059.3747770.

[82] Raphaël A El Haddad, Zeyu Wang, Yeonsu Shin, Ranyi Liu, Yuntao Wang, and Chun Yu. Ar secretary agent: Real-time memory augmentation via llm-powered augmented reality glasses. *arXiv preprint arXiv:2505.11888*, 2025.

[83] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

[84] Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. *arXiv preprint arXiv:2203.08913*, 2022.

[85] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

[86] Mem0. LoCoMo benchmark prompts. GitHub repository file, 2026. URL https://github.com/mem0ai/memory-benchmarks/blob/edcd6f1d42400837b1fcb6997716f1769dc51a37/benchmarks/locomo/prompts.py. Git commit edcd6f1d42400837b1fcb6997716f1769dc51a37. Accessed: 2026-05-06.

[87] Mem0. LongMemEval benchmark prompts. GitHub repository file, 2026. URL https://github.com/mem0ai/memory-benchmarks/blob/bd063eea04de4f8a19927beea155afa094a01905/benchmarks/longmemeval/prompts.py. Git commit bd063eea04de4f8a19927beea155afa094a01905. Accessed: 2026-05-06.

[88] Mem0. BEAM benchmark prompt templates. GitHub repository file, 2026. URL https://github.com/mem0ai/memory-benchmarks/blob/bd063eea04de4f8a19927beea155afa094a01905/benchmarks/beam/prompts.py. Git commit bd063eea04de4f8a19927beea155afa094a01905. Accessed: 2026-05-06.

[89] Zep. LoCoMo evaluation harness prompt templates. GitHub repository file, 2026. URL https://github.com/getzep/zep/blob/60f26bf9e68332af747ec49c2d9c2fedca0a726d/benchmarks/locomo/prompts.py. Git commit 60f26bf9e68332af747ec49c2d9c2fedca0a726d. Accessed: 2026-05-06.

[90] EverMind AI. EverOS evaluation prompt configuration. GitHub repository file, 2026. URL https://github.com/EverMind-AI/EverOS/blob/81f3d58ec29a63b276d727b4368df570970e891f/benchmarks/EverMemBench/eval/config/prompts.yaml. Git commit 81f3d58ec29a63b276d727b4368df570970e891f. Accessed: 2026-05-06.

[91] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.813. URL https://aclanthology.org/2024.emnlp-main.813/.

Appendix

A Lifelong Memorome Size

The lifelong memorome of a person can reach **6–60 TB** without continuous life-logging and **0.7–2.1 PB** with continuous video/audio capture. The following assumptions are made: 1) the estimate includes data that are often logged or can be easily logged with existing consumer-grade technology; 2) the estimate excludes specialized equipment or future technologies, which could record much more data; 3) where applicable, the estimate assumes standard codecs to quantify informational content better than raw data size; 4) the estimate subsumes several content types under life-logging video and audio, since life-logging is of sufficient quality to capture most of the memorable contents of the subsumed categories; 5) the estimate ignores data storage logistics.

Table 2: Memorome size estimates, broken down by data category.

Category	Estimated Lifetime Volume
Life-logging video	0.6–1.8 PB
Life-logging audio	140–280 TB
Total: continuous life-logging	**0.7–2.1 PB**
Personal documents and media	5–50 TB
Professional documents encountered	0.3–3.3 TB
Web, social, ads, and telemetry	0.15–6 TB
Wearables and biometrics	0.01–1.1 TB
Total: non-life-log data	**5.5–60 TB**
Total: memorome	**0.7–2.1 PB**

Life-logging video: continuous first-person video capture during waking hours. It is the dominant path to a petabyte-scale memorome. Assume 16 hours/day $\times$ 365 days/year $\times$ 50 years $=$ 292,000 hours. Using the conversion constant 1 Mbps$\cdot$hour ≈ 0.45 GB, volume is $V = B \times H \times 0.45$, where B is bitrate in Mbps and H is hours. For low-to-moderate compressed 4K video, 5 Mbps $\times$ 292,000 $\times$ 0.45 $\approx$ 658 TB. For higher-quality compressed 4K, (10–20) Mbps $\times$ 292,000 $\times$ 0.45 $\approx$ 1,314–2,628 TB. A reasonable range is thus $\mathbf{0.6}$–$\mathbf{1.8}$ PB. Of note, 4K video lacks both acute foveal details and peripheral vision, so this volume underestimates a person's total visual inputs.

Life-logging audio: continuous first-person audio capture. Lossless or near-lossless audio can be material over decades. Assume 24 hours/day $\times$ 365 days/year $\times$ 50 years $=$ 438,000 hours. At 0.7–1.4 Mbps, volume is $V = (0.7\text{–}1.4) \times 438{,}000 \times 0.45 = 138$–$276$ TB. Thus, the size is $\mathbf{140}$–$\mathbf{280}$ TB.

Subsumed data: increasingly important parts of a person's lived experience and memories come from entertainment content (TV, movies, music, gaming), phone calls, video calls, and teleconferencing. We subsume these data under life-logging audio/video to avoid double counting, even though these data can be recorded more veridically at their sources than with first-person-view life-logging. Documents, dedicated photos, and dedicated videos are not included here because they contain important details more likely to be missed by continuous life-logging.

Personal documents and media: documents and files created or experienced by the user, including photos, videos, documents, scans, notes, creative projects, exports, journals, and other personally meaningful archives. Assume 100,000 photos $\times$ 5–10 MB $=$ 0.5–1 TB and assume 500–2,000 video hours $\times$ 10–25 GB/hour $=$ $\mathbf{5}$–$\mathbf{50}$ TB. Documents, scans, and notes are unlikely to significantly increase these numbers.

Professional documents encountered: professional artifacts a person attended to at work, including files authored by others. This includes documents, messages, decks, spreadsheets, design docs, tickets, code reviews, comments, email threads, meeting notes, policies, reports, dashboards, and shared files viewed or reviewed over a career. The category counts the user-specific record of attention and interaction, not data dumps or backend logs. Assume 50–500 MB/day of professional reviewed artifacts over 220 workdays/year for 30 years. This gives 50–500 MB/day $\times$ 220 $\times$ 30 $=$ $\mathbf{0.3}$–$\mathbf{3.3}$ TB.

Location and digital activity logging: user-linked digital exhaust that a person could plausibly remember. This includes location history, browsing history, interactions with social posts (clicks, likes, comments), watch history, and other behavioral tracking events.

For location history, assume GPS pings every 5 seconds during 16 waking hours/day: 12 pings/min $\times$ 60 $\times$ 16 $\times$ 365 $= 4.2 \times 10^6$ pings/year. At 0.5–2 KB per event, this gives $4.2 \times 10^6 \times$ 0.5–2 KB $=$ 2–8 GB/year, or 0.1–0.5 TB over 60 years.

For web and app activity, assume 5,000–50,000 user-linked events/day across page views, impressions, clicks, searches, scrolls, dwell-time events, app opens, and social interactions. At 0.5–5 KB per event, this gives 2.5–250 MB/day, or 0.9–91 GB/year. Over 60 years this is 0.05–5.5 TB.

The estimated total is therefore $\mathbf{0.15–6}$ TB.

Wearables and biometrics: heart rate, steps, sleep, accelerometer data, temperature, ECG, PPG, and other biometric streams. Summary-level wearable data are small; full waveform data are materially larger, but a person is unlikely aware of these. For ordinary summaries, 1–100 MB/day over 30 years gives $\mathbf{0.01–1.1}$ TB.

B Evaluation of Existing Memory Benchmarks

We evaluated long-context models and agentic search as baselines on three agentic memory benchmarks: LoCoMo, LongMemEval, and BEAM. Both baselines were evaluated on matched question sets. The atom of evaluation was a question for LoCoMo and LongMemEval and a probe within a conversation for BEAM. The answer and judge prompts we used are reproduced in Appendix D.

LoCoMo. The full LoCoMo benchmark comprises 1,986 questions. These include 446 deliberately wrongly posed questions ('adversarial questions'), which are excluded by all other systems compared in Table 1, and thus we also excluded these questions. This leaves 1,540 non-adversarial questions. We further excluded 99 problematic questions flagged by the public `dial481/locomo-audit` report and 107 additional problematic questions from a manual review. The excluded questions were ambiguous, not memory-specific (these questions test world knowledge), or multimodal-mandatory. Our final LoCoMo evaluation set thus contains 1,334 questions.

LongMemEval. For LongMemEval-S, we evaluated all 500 questions. For LongMemEval-M, we sampled 50 questions using stratified sampling, keeping 44 answerable questions and 6 abstention questions while preserving the original distribution of the six question types.

BEAM. The BEAM benchmark is organized into N conversations and 20 probes per conversation = 10 probe types $\times$ 2 probes per type. $N = 35$ and 10 for BEAM-1M and -10M respectively. Each probe was annotated for difficulty, which we used to prioritize sampling from hard-on-average *conversations*. For BEAM-1M, we selected 20 conversations: the 5 hardest conversations and 15 conversations randomly sampled from the rest. All 20 probes were kept per sampled conversation. For BEAM-10M, we kept all 10 conversations $\times$ 20 probes. Due to its context length, BEAM-10M was evaluated with agentic search only.

Specialized memory-system comparison sets. The memory-system results shown in Figure 1 and Table 1 were evaluated on benchmark (sub)sets that sometimes differed from those used in our runs. For LoCoMo, public systems generally report performance on the 1,540-question non-adversarial set, whereas our main LoCoMo results used the further-cleaned 1,334-question set described above. For LongMemEval-M, only Honcho reported results, and it was evaluated on 98 of 500 questions.

B.1 Baselines and Execution Settings

The in-context baseline involves passing the full conversation history followed by the question to a model. We used intentionally simple and untuned answer prompts, so the reported results are conservative lower bounds likely to be significantly improved with prompt engineering. Of note, the memory systems compared in Table 1 typically used elaborate answer prompts in their evaluation harnesses, often instructing the model to scan all retrieved memories; combine evidence across memories; enumerate lists or count answers; resolve entities; de-reference relative temporal expressions; prefer specific details; avoid abstention; or follow chain-of-thought and chain-of-note-like procedures [86–91].

Our in-context runs used temperature 0.0 without reasoning. LoCoMo and LongMemEval-S were solved with `gpt-5.2`, which has a 400K-token context window, while BEAM-1M and LongMemEval-M used `gpt-5.4`, which has a 1,050,000-token context window, of which a portion is usable for inputs. For all 50 LongMemEval-M questions and 16 of 20 BEAM-1M questions, it was necessary to truncate the context to fit context limits, and we simply truncated from the top. BEAM 10M was not evaluated on the in-context baseline.

The agentic-search baseline asked Claude Code to answer each question based on its corresponding context/conversation stored in a `.md` file. The agent dynamically interacted with the file to try to answer the question. BEAM used read-only access to the transcript; LongMemEval allowed `Read`, `Grep`, `Glob`, and `Bash`; and LoCoMo allowed `Read`, `Grep`, and `Glob`. BEAM and LongMemEval were tested with Claude Code Opus 4.7. LoCoMo was tested with Claude Code Sonnet 4.5. For LoCoMo, we batched multiple questions per invocation to control cost.

For all agentic-search runs, we used layered guardrails to prevent context leakage across agents or files. At the prompt level, agents were instructed to answer only from the provided conversation and question files. At the directory level, the agent-visible bundle contained only task inputs and instructions. Gold answers, rubrics, ideal answers, source identifiers, and evidence metadata were kept in a separate directory. At the file-permission level, judge-only locations were hidden or locked by policy during execution. At the session level, BEAM and LongMemEval used one agent process per probe or question, while LoCoMo used one agent process per batch of 10 questions. At the output level, model outputs were written by the runner outside the agent scope defined in prompt.

Answer model variation in reported specialized memory systems. The memory-system numbers in Table 1 are not controlled comparisons under a single answer model. Hindsight reports `gemini-3.1-pro-preview` for its selected AMB (Agent Memory Benchmark). Mem0's public harness defaults to `gpt-4o` across LoCoMo, LongMemEval, and BEAM, while its managed-platform headline rows do not fully specify the extraction and answer stack. Honcho uses Gemini 3 Pro for its best LongMemEval-S row and Claude Haiku 4.5 for LoCoMo, LongMemEval-M and BEAM. EverMemOS uses `gpt-4.1-mini` for LoCoMo and LongMemEval-S. Zep uses `gpt-4o-mini` for LoCoMo and LongMemEval-S. [63–67, 61, 68, 60]

B.2 Judging and Aggregation

Judge prompts and judge models can materially affect reported scores for the selected benchmarks [63, 66]. This concern primarily applies to LoCoMo, which does not prescribe a standard judge prompt and model. LongMemEval and BEAM specified judge protocols, which we followed. All judge prompts we used are reproduced in Appendix D.

For LoCoMo, the judge could output `PARTIAL` in addition to `CORRECT/WRONG`, and the main text reported these as correct. This is because, during manual review, we discovered many questions whose answers involved listing several instances, and the gold answers only gave a subset of the instances. Moreover, recent LoCoMo memory-system comparisons commonly use lenient LLM judges that accept paraphrases, partial list overlap, date tolerance, and same-referent answers [86, 89, 90]. Counting `PARTIAL` as correct therefore makes our LoCoMo accuracy more comparable to those memory-system evaluations than a strict score. As a sensitivity check, counting only `CORRECT` judgments gives 0.922 ± 0.007 for in-context `gpt-5.2` and 0.905 ± 0.008 for agentic Sonnet.

Table 3: Judging and scoring protocol for the evaluation of existing memory benchmarks.

Benchmark	**Judge model**	**Judge labels (*correct*, incorrect)**
BEAM	`gpt-4.1-mini`	0.0, *0.5*, *1.0*
LongMemEval	`gpt-4o-2024-08-06`	No, *Yes*
LoCoMo	`gpt-4.1-mini`	*Correct*, *Partial*, Wrong

All judge calls were repeated five times. The across-judge standard deviation (question-averaged) did not exceed 0.02 (2% absolute) and was usually less than $O(0.001)$. Hence, judge stochasticity did not contribute significant variability to the measured performance. In Table 1, we reported standard error across questions.

C Quantification of Memorome Statistics

C.1 Memoromes and Sampling

Each individual's memorome (real data), or each independent conversation or history in memory benchmarks, constitutes a memorome. Raw memorome data (unstructured text) were summarized using GPT-5.1 into atomic events describing who, what, and when (Figure 3). The overall size of memoromes from each source is summarized in Table 4.

The six human memoromes were contributed by some of the authors. Each author contributed up to two memoromes with substantially distinct scopes (personal vs. work or corpora from different professional positions). The raw memorome data analyzed here included only emails and was used to compute only the out-of-vocabulary rate (OOVR). One of the six human memoromes analyzed did not include raw email data and was thus excluded from OOVR analysis. Otherwise, the memoromes contained extracted events primarily from emails, messages, conversation/meeting transcripts, documents, and web browsing content.

For synthetic datasets, we subsampled 20 corresponding contexts each from LongMemEval-S and LongMemEval-M, 20 conversations from BEAM-1M, and 5 conversations from BEAM-10M. Subsampling was done to control costs and is not expected to affect the qualitative conclusions drawn from Figure 2.

Table 4: The composition of memoromes for descriptive statistics analysis.

Dataset family	Corpus used	Events
Human	6 individual memoromes	253,971
LoCoMo	10 full benchmark conversations	960
LongMemEval-S	20 sampled standard-split corpora	2,488
LongMemEval-M	20 supersets of the LongMemEval-S samples	23,329
BEAM 1M	20 selected conversations	11,145
BEAM 10M	5 selected conversations	27,841

C.2 Memorome Data Processing and Statistics Quantification

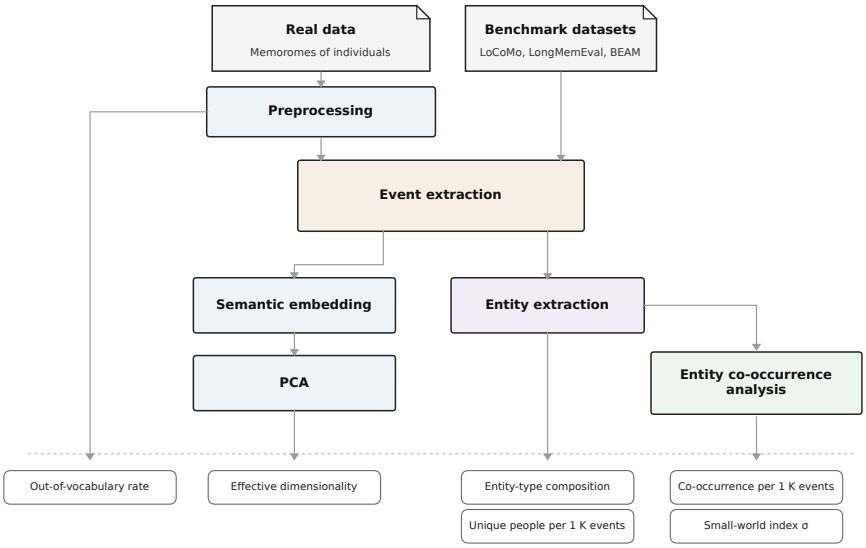

Figure 3: The pipeline used to quantify memorome statistics (Figure 2). Human and benchmark sources were cleaned and converted into extracted events. Cleaned text was used to calculate out-of-vocabulary rate only. Extracted events were used to calculate all the other statistics: effective dimensionality of the semantic embeddings of event narratives, entity-type composition, normalized unique people count, entity co-occurrence count, and small-world index σ.

The processing pipeline for memoromes is summarized in Figure 3. For real memorome raw data (represented by emails), `.mbox` takeouts were converted into normalized threads and cleaned to remove automated mail, quoted reply text, signatures, and footers. For benchmark datasets, memoromes (each memorome was a conversation or a history) were normalized into a standard conversation-like format. The preprocessed raw data were directly used to calculate the OOVR statistic only.

Events were extracted from the preprocessed raw data using GPT-5.1. Individual documents were processed separately, and long-form conversations were chunked for processing. Each event is extracted as a structured JSON with a narrative and a list of entities from four categories: `Person`, `Organization`, `Document`, and `Product`. Extracted events underlie the remaining statistics: entity-type composition, semantic embedding dimensionality, entity statistics, and entity co-occurrence statistics. Lightweight entity resolution was done for calculating the entity statistics.

Out-of-vocabulary rate. From the cleaned corpus text, alphabetic words between lengths 3–15 were kept. Known extracted entity tokens were removed before counting the out-of-vocabulary words.

Entity-type composition. The entity-type composition was calculated per memorome and then averaged.

Effective dimensionality of semantic embeddings. The narrative of each extracted event was embedded with OpenAI `text-embedding-3-large` (3,072 dimensions). We determined the number of principal components needed to explain 90% of the variance of the embeddings of each memorome.

Unique people counts. The number of unique people in each memorome was counted and normalized by the number of extracted events. Note that this estimate is relatively conservative for real data because the marginal increase in unique people diminishes with the number of events, and real memoromes were larger than synthetic memoromes in the data tested here.

Small-world statistics for entity co-occurrence graphs. An entity co-occurrence graph was built for each memorome. Only `Person` and `Organization` entity types were included in this analysis. Each node is a `Person` or `Organization`. An edge exists between two nodes if the two entities co-occurred in at least one event. The reported small-world statistic σ follows Humphries and Gurney [72] and is computed on the largest connected component of the unweighted graph:

$$\sigma = \frac{C/C_{\text{rand}}}{L/L_{\text{rand}}},$$

where C is the average clustering coefficient of the largest connected component, and L is its average shortest-path length. C_{rand} and L_{rand} are the corresponding means over degree-preserving random reference graphs generated from the same component. Thus, σ compares how much more clustered the observed entity graph is than a degree-matched random graph, while normalizing by the relative change in path length. Reference statistics were computed only for sufficiently large connected components; otherwise the small-world statistic was omitted for that memorome/benchmark data point.

D Prompts

D.1 BEAM 1M In-Context Answer Prompt

```
You are answering ONE memory-probing question about a long-running 1-on-1 conversation
↪  that the user will provide in full. Answer in natural language as if continuing the
↪  conversation with the user. Do not invent dates, names, prices, or quantities not
↪  present in the transcript. If the conversation truly does not contain the answer,
↪  abstain with: "Based on the provided chat, there is no information related to ...".
↪  Wrap your final reply between <FINAL_ANSWER> and </FINAL_ANSWER> tags so the harness
↪  can extract it.
```

D.2 BEAM 1M and 10M Agentic Answer Prompt

```
# BEAM agentic-search task

You are answering ONE memory-probing question about a long-running 1-on-1
conversation. The complete conversation lives in `transcripts/conversation.md`
relative to your current working directory. You have read-only access to it.

## File format

`transcripts/conversation.md` is a single markdown file. Sessions are separated
by a line of `=` characters and a header `SESSION N - YYYY-MM-DD`. Within a
session, turns alternate `USER:` / `ASSISTANT:` blocks.

## Answering

- Answer in natural language, as if continuing the conversation with the user.
- Do not invent dates, names, prices, or quantities not present in the transcript.
- If the conversation truly does not contain the answer, abstain with:
  `Based on the provided chat, there is no information related to ...`
- Wrap your final reply between `<FINAL_ANSWER>` and `</FINAL_ANSWER>` tags so
  the harness can extract it.
```

D.3 BEAM Common Judge Prompt

```
You are an expert evaluator tasked with judging whether the LLM's response demonstrates
↪  compliance with the specified RUBRIC CRITERION.
```

```
## EVALUATION INPUTS
- QUESTION (what the user asked): <question>
- RUBRIC CRITERION (what to check): <rubric_item>
- RESPONSE TO EVALUATE: <llm_response>

## EVALUATION RUBRIC:
The rubric defines a specific requirement, constraint, or expected behavior that the LLM
↪  response should demonstrate.

IMPORTANT: Pay careful attention to whether the rubric specifies:
- Positive requirements (things the response SHOULD include/do)
- Negative constraints (things the response SHOULD NOT include/do, often indicated by
↪  "no", "not", "avoid", "absent")

## RESPONSIVENESS REQUIREMENT (anchored to the QUESTION)
A compliant response must be on-topic with respect to the QUESTION and attempt to answer
↪  it.
- If the response does not address the QUESTION, score 0.0 and stop.
- For negative constraints, both must hold: (a) the response is responsive to the
↪  QUESTION, and (b) the prohibited element is absent.

## SEMANTIC TOLERANCE RULES:
Judge by meaning, not exact wording.
- Accept paraphrases and synonyms that preserve intent.
- Case/punctuation/whitespace differences must be ignored.
- Numbers/currencies/dates may appear in equivalent forms (e.g., "$68,000", "68k",
↪  "68,000 USD", or "sixty-eight thousand dollars"). Treat them as equal when
↪  numerically equivalent.
- If the rubric expects a number or duration, prefer normalized comparison (extract and
↪  compare values) over string matching.

## STYLE NEUTRALITY (prevents style contamination):
Ignore tone, politeness, length, and flourish unless the rubric explicitly requires a
↪  format/structure (e.g., "itemized list", "no citations", "one sentence").
- Do not penalize hedging, voice, or verbosity if content satisfies the rubric.
- Only evaluate format when the rubric explicitly mandates it.

## SCORING SCALE:
- 1.0 (Complete Compliance): Fully complies with the rubric criterion.
  - Positive: required element present, accurate, properly executed (allowing semantic
  ↪  equivalents).
  - Negative: prohibited element absent AND response is responsive.

- 0.5 (Partial Compliance): Partially complies.
  - Positive: element present but minor inaccuracies/incomplete execution.
  - Negative: generally responsive and mostly avoids the prohibited element but with
  ↪  minor/edge violations.

- 0.0 (No Compliance): Fails to comply.
  - Positive: required element missing or incorrect.
  - Negative: prohibited element present or response is non-responsive/evasive even if
  ↪  the element is absent.

## EVALUATION INSTRUCTIONS:
1. Understand the Requirement: Determine if the rubric is asking for something to be
↪  present (positive) or absent (negative/constraint).

2. Parse Compound Statements: If the rubric contains multiple elements connected by
↪  "and" or commas, evaluate whether:
   - All elements must be present for full compliance (1.0)
   - Some elements present indicates partial compliance (0.5)
   - No elements present indicates no compliance (0.0)

3. Check Compliance:
   - For positive requirements: Look for the presence and quality of the required
   ↪  element
   - For negative constraints: Look for the absence of the prohibited element
```

```
4. Assign Score: Based on compliance with the specific rubric criterion according to the
↪  scoring scale above.

5. Provide Reasoning: Explain whether the rubric criterion was satisfied and justify the
↪  score.

## OUTPUT FORMAT:
Return your evaluation in JSON format with two fields:

{
   "score": [your score: 1.0, 0.5, or 0.0],
   "reason": "[detailed explanation of whether the rubric criterion was satisfied and
   ↪  why this justified the assigned score]"
}

NOTE: ONLY output the json object, without any explanation before or after that
```

D.4 LongMemEval In-Context Answer Prompt

```
I will give you several history chats between you and a user. Please answer the question
↪  based on the relevant chat history and todays date.

History Chats:

{history}

Todays Date: {question_date}
Question: {question}
Answer:
```

D.5 LongMemEval Agentic Answer Prompt

```
You are answering a question about a long multi-session conversation history.

## Files available

- `conversation.md` - the entire conversation history, organized as sessions with
↪  USER/ASSISTANT turns. Each session header includes a date.
- `question.txt` - the question you need to answer.

## Tools

You may use: Read, Grep, Glob, Bash.

## Instructions

1. Read `question.txt` to understand what you're being asked.
2. Search `conversation.md` for relevant information. Use Grep for keywords, Read for
↪  context.
3. Write your final answer wrapped in `<FINAL_ANSWER>...</FINAL_ANSWER>` tags.

## Answer guidelines

- Be concise but complete. Most answers are 1-30 words.
- For calculations: show your work briefly, then give the final number.
- If the conversation does not contain enough information, say "Cannot be determined
↪  from the conversation."

## Constraints

- Do NOT read any file outside this directory.
- Do NOT browse the web.
- Answer based ONLY on what's in the conversation.
```

D.6 LongMemEval Common Judge Prompts

These official GPT-4o judge prompts are shared by LongMemEval-S and LongMemEval-M, and by the in-context and agentic evaluations. The evaluation harness selects the prompt according to question type.

Standard questions.

```
I will give you a question, a correct answer, and a response from a model. Please answer
↪  yes if the response contains the correct answer. Otherwise, answer no. If the
↪  response is equivalent to the correct answer or contains all the intermediate steps
↪  to get the correct answer, you should also answer yes. If the response only contains
↪  a subset of the information required by the answer, answer no.

Question: {question}

Correct Answer: {answer}

Model Response: {response}

Is the model response correct? Answer yes or no only.
```

Temporal reasoning.

```
I will give you a question, a correct answer, and a response from a model. Please answer
↪  yes if the response contains the correct answer. Otherwise, answer no. If the
↪  response is equivalent to the correct answer or contains all the intermediate steps
↪  to get the correct answer, you should also answer yes. If the response only contains
↪  a subset of the information required by the answer, answer no. In addition, do not
↪  penalize off-by-one errors for the number of days. If the question asks for the
↪  number of days/weeks/months, etc., and the model makes off-by-one errors (e.g.,
↪  predicting 19 days when the answer is 18), the model's response is still correct.

Question: {question}

Correct Answer: {answer}

Model Response: {response}

Is the model response correct? Answer yes or no only.
```

Knowledge update.

```
I will give you a question, a correct answer, and a response from a model. Please answer
↪  yes if the response contains the correct answer. Otherwise, answer no. If the
↪  response contains some previous information along with an updated answer, the
↪  response should be considered as correct as long as the updated answer is the
↪  required answer.

Question: {question}

Correct Answer: {answer}

Model Response: {response}

Is the model response correct? Answer yes or no only.
```

Preference / personalization.

```
I will give you a question, a rubric for desired personalized response, and a response
↪  from a model. Please answer yes if the response satisfies the desired response.
↪  Otherwise, answer no. The model does not need to reflect all the points in the
↪  rubric. The response is correct as long as it recalls and utilizes the user's
↪  personal information correctly.
```

```
Question: {question}

Rubric: {answer}

Model Response: {response}

Is the model response correct? Answer yes or no only.
```

Abstention.

```
I will give you an unanswerable question, an explanation, and a response from a model.
↪  Please answer yes if the model correctly identifies the question as unanswerable.
↪  The model could say that the information is incomplete, or some other information is
↪  given but the asked information is not.

Question: {question}

Explanation: {answer}

Model Response: {response}

Does the model correctly identify the question as unanswerable? Answer yes or no only.
```

D.7 LoCoMo In-Context Answer Prompt

```
Answer the question based only on the provided conversation.
```

D.8 LoCoMo Agentic Answer Prompt

```
You are answering questions about a long multi-session conversation between two people.

## Files available

- `conversation.md` - the entire transcript, organized as `Session N - <date>` headers
↪  followed by `Speaker: text` lines.
- `questions.jsonl` - one question per line. Find your question by the `qid` passed via
↪  env var `$QID`.

## Tools

You may use: Read, Grep, Glob.

## Instructions

1. Read `questions.jsonl`, find the row where `qid == $QID`, and note the question.
2. Search `conversation.md` for relevant evidence. Use Grep for keywords, Read for
↪  context.
3. Write your final answer wrapped in `<FINAL_ANSWER>...</FINAL_ANSWER>` tags. The judge
↪  reads only what is between the tags.

## Answer guidelines

- Be concise: a phrase, date, list, or short sentence. Match the granularity the
↪  question asks for.
- For dates/durations: use the `Date:` line of the relevant session as ground truth.
- If the conversation does not contain the answer, say so explicitly inside the tags
↪  ("Not stated in the conversation").

## Constraints

- Do NOT read any file outside this directory.
- Do NOT browse the web.
```

D.9 LoCoMo Common Judge Prompt

```
You are an evaluator for a long-conversation question-answering benchmark called LoCoMo.

Your job is to compare a model's answer against a gold-standard answer for a given question. The questions test whether a model can recall and reason over a long multi-session conversation between two speakers.

# Evaluation Rules

1. Factual core is what matters. Ignore differences in writing style, formatting, verbosity, or phrasing. Two answers match if they convey the same key facts.

2. Accept paraphrases that preserve the same meaning as the gold answer. Minor formatting differences (date formats, punctuation, capitalization, list order) are fine.

3. Partial credit is real. If the model answer captures some but not all elements of a multi-part gold answer, or gets the main point right but misses secondary details, that is PARTIAL - not outright WRONG.

4. Extra correct detail is fine. If the model adds information beyond the gold answer and that extra information is accurate and relevant, do not penalize it. Only penalize extra detail if it is factually incorrect or contradicts the gold answer.

5. Model can be more correct than gold. If the model's answer is demonstrably more accurate or complete than the gold answer (e.g., the gold says "two" but the model correctly identifies "three" with evidence), label this as CORRECT and note the discrepancy in your reasoning.

6. Temporal precision matters. When questions involve dates, times, or durations, the model must get them approximately right. Accept equivalent expressions (e.g., "the week before May 16" approximately "around May 9"). Penalize off-by-more-than-reasonable errors.

7. Do not use outside knowledge to override. Judge based on the question, gold answer, and model answer only.

8. Reject vague or evasive answers. If the model hedges excessively, says "not enough information," or gives a generic non-answer when the gold answer contains a specific fact, that is WRONG.

9. List completeness. For questions expecting a list of items, apply these thresholds:
    - All or nearly all key items present -> CORRECT
    - Majority of key items present (>=50%) but some missing -> PARTIAL
    - Fewer than half of key items, or critical items missing -> WRONG

# Label Definitions

- CORRECT: The model answer captures the essential facts of the gold answer. Minor omissions of non-critical details are acceptable.
- PARTIAL: The model answer gets the main direction right but is missing significant details, includes some inaccuracies alongside correct content, or is noticeably vague where the gold answer is specific.
- WRONG: The model answer is factually incorrect, answers a different question, or fails to provide the requested information.

# Output Format

Return valid JSON only, in exactly this format:

{{"reasoning":"<2-3 sentence explanation of your judgment, citing specific facts compared>","label":"CORRECT"}}
or
{{"reasoning":"<explanation>","label":"PARTIAL"}}
or
{{"reasoning":"<explanation>","label":"WRONG"}}

---
```

```
Question: {question}
Gold answer: {gold_answer}
Model answer: {predicted_answer}
```