← Back to **Author Console** (/group?id=ICLR.cc/2026/Conference/Authors#your-submissions)

EVALUATING MEMORY IN LLM AGENTS VIA INCRE- MENTAL MULTI-TURN INTERACTIONS

(/pdf?id=DT7JyQC3MR)

Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1), Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1)

Published: 26 Jan 2026, Last Modified: 26 Jan 2026 ICLR 2026 Everyone
Revisions (/revisions?id=DT7JyQC3MR) BibTeX

Keywords: LLM Agents; Agents with Memory; Memory Agents Benchmark; Evaluation for Memory
TL;DR: We introduce a unified evaluation framework designed for memory agents.

Abstract:
Recent benchmarks for Large Language Model (LLM) agents primarily focus on evaluating reasoning, planning, and execution capabilities, while another critical component—memory, encompassing how agents memorize, update, and retrieve long-term information—is under-evaluated due to the lack of benchmarks. We term agents with memory mechanisms as memory agents. In this paper, based on classic theories from memory science and cognitive science, we identify four core competencies essential for memory agents: accurate retrieval, test-time learning, long-range understanding, and selective forgetting. Existing benchmarks either rely on limited context lengths or are tailored for static, long-context settings like book-based QA, which do not reflect the interactive, multi-turn nature of memory agents that incrementally accumulate information. Moreover, no existing benchmarks cover all four competencies. We introduce MemoryAgentBench, a new benchmark specifically designed for memory agents. Our benchmark transforms existing long-context datasets and incorporates newly constructed datasets into a multi-turn format, effectively simulating the incremental information processing characteristic of memory agents. By carefully selecting and curating datasets, our benchmark provides comprehensive coverage of the four core memory competencies outlined above, thereby offering a systematic and challenging testbed for assessing memory quality. We evaluate a diverse set of memory agents, ranging from simple context-based and retrieval-augmented generation (RAG) systems to advanced agents with external memory modules and tool integration. Empirical results reveal that current methods fall short of mastering all four competencies, underscoring the need for further research into comprehensive memory mechanisms for LLM agents.

Primary Area: datasets and benchmarks
Code Of Ethics: I acknowledge that I and all co-authors of this work have read and commit to adhering to the ICLR Code of Ethics.
Submission Guidelines: I certify that this submission complies with the submission instructions as described on https://iclr.cc/Conferences/2026/AuthorGuide (https://iclr.cc/Conferences/2026/AuthorGuide).
Reciprocal Reviewing Author: Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1)
Reciprocal Reviewing Exemption: We do not need an exemption.
Resubmission: Yes
Student Author: Yes
Anonymous Url: I certify that there is no URL (e.g., github page) that could be used to find authors' identity.
No Acknowledgement Section: I certify that there is no acknowledgement section in this submission for double blind review.
Large Language Models: Yes, to aid or polish writing. Details are described in the paper.
Submission Number: 15836

Paper Decision

Decision by Program Chairs 26 Jan 2026, 00:46 (modified: 26 Jan 2026, 04:39) Program Chairs, Authors

Revisions (/revisions?id=o1D3SWn2Vl)

Decision: Accept

Meta Review of Submission15836 by Area Chair 7vkr

Meta Review by Area Chair 7vkr 22 Dec 2025, 21:44 (modified: 26 Jan 2026, 05:27)

Senior Area Chairs, Area Chairs, Authors, Program Chairs Revisions (/revisions?id=RwE8eXFlXJ)

Summary:
This paper introduces MemoryAgentBench, a benchmark evaluating memory agents across four competencies: Accurate Retrieval, Test-Time Learning, Long-Range Understanding, and Selective Forgetting. The benchmark repurposes existing long-context datasets into incremental multi-turn interactions and introduces two new datasets, EventQA and FactConsolidation. The primary strength lies in its comprehensive evaluation framework that moves beyond static long-context assessment to simulate realistic incremental information intake across diverse memory architectures including long-context models, RAG systems, and agentic memory agents. The systematic categorization of memory competencies grounded in memory science provides a principled foundation. However, weaknesses include concerns about the Selective Forgetting task appearing somewhat artificial with synthetic serial numbers and unclear real-world motivation, the Test-Time Learning operationalization being more about retrieval from long histories than genuine online learning, and potential architectural bias where TTL tasks inherently favor long-context models over RAG systems. The benchmark also focuses exclusively on language-based QA tasks without covering action-heavy or procedural domains. Despite these limitations, the paper makes a valuable contribution by establishing a unified evaluation protocol with sufficient scale and diversity, revealing important limitations in current memory systems, and providing actionable insights through comprehensive ablation studies. I would recommend weak acceptance given the timeliness of the problem and the benchmark's potential to anchor future research, though with reservations about certain design choices.

Reviewer Concerns:
The authors provided thorough rebuttals addressing most reviewer concerns. For Reviewer nPYL, they added a RAG benchmarks subsection to related work, justified Selective Forgetting as addressing capacity constraints in memory systems with ablation showing the task is solvable in short contexts, and defended Test-Time Learning terminology by demonstrating zero-shot baselines perform significantly worse. For Reviewer zBBu, they explained the focus on language tasks as necessary to decouple memory from reasoning confounds, provided zero-shot experiments confirming genuine learning rather than prior knowledge, and added detailed cost analysis showing MIRIX achieves better cost-performance tradeoffs than raw long-context models. For Reviewer Mcjo, they justified LLM-as-judge by citing prior validation studies, explained chunking as reflecting realistic quantized input processing, and clarified that dialogue format is essential for memory agents though equivalent for long-context baselines. For Reviewer RnwS, they conducted strict compute-matched comparisons revealing RAG efficiency at low budgets but long-context capacity advantages at high budgets, and tested explicit overwrite policy prompts showing marginal improvements that validate Selective Forgetting cannot be solved by prompting alone. The rebuttal successfully addressed evaluation validity concerns and architectural fairness questions. Outstanding concerns include whether Selective

Forgetting generalizes beyond the synthetic setting and whether the benchmark scope limitation to language tasks reduces broader applicability claims. These responses demonstrate good faith engagement and strengthen confidence in the technical soundness, though some conceptual debates about task design remain unresolved.

Reviewer Scores:

Based on reviewer engagement, Reviewer nPYL would likely maintain their score of 4, as while the rebuttal addressed technical gaps like missing related work, their fundamental concern about Selective Forgetting being boutique remains philosophically unresolved despite reasonable justification. Reviewer zBBu provided a score of 6 and the comprehensive cost analysis plus zero-shot experiments likely solidify or slightly improve their position to a strong 6, as their main fairness and validity concerns were directly addressed with new experiments. Reviewer Mcjo gave an initial 8 and appears fully satisfied based on the tone of questions rather than criticisms, suggesting they would maintain their strong accept at 8. Reviewer RnwS at 6 would likely stay 6 or move to 8 given the authors conducted exactly the compute-matched experiments requested and provided the prompt policy grid tests, demonstrating responsiveness to their core methodological concerns.

Official Review of Submission15836 by Reviewer nPYL

Official Review by Reviewer nPYL 02 Nov 2025, 21:25 (modified: 11 Nov 2025, 21:41) Everyone

Revisions (/revisions?id=FDayACmaND)

Summary:

This paper introduces a benchmark (datasets and evaluation metrics) for evaluating so-called "memory agents" - essentially sophisticated versions of RAG-like tasks that need retrieval before (maybe some reasoning and then) generation.

There are 4 categories of tasks; 3 of them somewhat standard, and one of them ("Selective Forgetting") that seems newly made in this paper. The datasets therein are a mix of re-purposed (about 70%) existing, and new (about 30%), datasets.

The authors evaluate several existing methods on this framework.

Soundness: 3: good
Presentation: 2: fair
Contribution: 3: good

Strengths:

Tests memory beyond simple retrieval, supporting diverse tasks, multi-turn access etc.

The authors have made a comprehensive evaluation of a large number of different styles of RAG systems and agents.

The authors have spent the effort to make some new datasets, and re-purpose existing work into a uniform framework.

Weaknesses:

There should be a subsection called "RAG Benchmarks" in Section 2, covering related work in that area as well.

The "Selective Forgetting" task seems like a boutique, somewhat fabricated one. It should be justified why measuring this is specifically important.

The "test-time-learning" title seems a bit of a misnomer, since the actual task is just learning from long histories/context.

Questions:

Why devise the "selective forgetting" question ? Seems like a boutique task, so why was this particular boutique task chosen in this paper ?

Is it possible to have a simple clear motivating example that shows where existing benchmarks fail to capture a specific particular way modern "memory agents" work ?

What exactly is a "memory agent" ? The term seems to have been left somewhat intentionally vague so as to be broad, but it would be better to clearly explain it.

Flag For Ethics Review: No ethics review needed.

Rating: 4: marginally below the acceptance threshold. But would not mind if paper is accepted
Confidence: 3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
Code Of Conduct: Yes

Rebuttal to Reviewer nPYL (Part1)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 02:19 Everyone

Comment:

> **W1:** There should be a subsection called "RAG Benchmarks" in Section 2, covering related work in that area as well.

Response to W1:

Thank you for pointing out the shortcomings in our related work section. We will provide a new subsection for "RAG Benchmarks" to the related work of our paper.

Benchmarks for Retrieval-Augmented Generation. Beyond pure long-context evaluation, a line of benchmarks targets retrieval-augmented generation (RAG) for knowledge-intensive tasks such as open-domain QA, fact checking, and document ranking over fixed corpora, e.g., KILT[1] and BEIR[2]. More recent work explicitly evaluates end-to-end RAG systems under long-context or application-specific scenarios, including LaRA[3], LONG^{2}RAG[4], FRAMES[5], and CRUD-RAG[6]. Large-scale benchmarks such as RAGBench[7], RAGTruth[8], FreshLLM[9], and T^2-RAGBench[10] further extend the evaluation space to industrial manuals, hallucination detection, time-sensitive web QA, and text-and-table financial reports, respectively. However, existing RAG benchmarks typically assume a static or slowly changing knowledge base and short-lived interactions, emphasizing retrieval accuracy and grounding, thereby failing to adequately assess other critical dimensions of Agent Memory, such as Test-Time Learning, Long Range Understanding, and Selective Forgetting.

Reference

[1] Petroni, Fabio, et al. "KILT: a benchmark for knowledge intensive language tasks." Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021.

[2] Thakur, Nandan, et al. "Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models." arXiv preprint arXiv:2104.08663 (2021).

[3] Li, Kuan, et al. "LaRA: Benchmarking Retrieval-Augmented Generation and Long-Context LLMs--No Silver Bullet for LC or RAG Routing." arXiv preprint arXiv:2502.09977 (2025).

[4] Qi, Zehan, et al. "Long ^{2} RAG: Evaluating Long-Context & Long-Form Retrieval-Augmented Generation with Key Point Recall." arXiv preprint arXiv:2410.23000 (2024).

[5] Krishna, Satyapriya, et al. "Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation." Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). 2025.

[6] Lyu, Yuanjie, et al. "Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models." ACM Transactions on Information Systems 43.2 (2025): 1-32.

[7] Friel, Robert, Masha Belyi, and Atindriyo Sanyal. "Ragbench: Explainable benchmark for retrieval-augmented generation systems." arXiv preprint arXiv:2407.11005 (2024).

[8] Niu, Cheng, et al. "Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models." Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024.

[9] Vu, Tu, et al. "Freshllms: Refreshing large language models with search engine augmentation." Findings of the Association for Computational Linguistics: ACL 2024. 2024.

[10] Strich, Jan, et al. "T ^{2}-RAGBench: Text-and-Table Benchmark for Evaluating Retrieval-Augmented Generation." arXiv preprint arXiv:2506.12071 (2025).

Rebuttal to Reviewer nPYL (Part2)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 02:21 (modified: 25 Nov 2025, 02:22) Everyone

Revisions (/revisions?id=JrTryTBzmh)

Comment:

> **W2:** The "Selective Forgetting" task seems like a boutique, somewhat fabricated one. It should be justified why measuring this is specifically important.

> **Q1:** Why devise the "selective forgetting" question ? Seems like a boutique task, so why was this particular boutique task chosen in this paper ?

Response to W2 & Q1: Rationale for the "Selective Forgetting" Task

We appreciate the reviewer's question regarding the motivation behind this task design. While the "Selective Forgetting" task may appear specialized, we designed it to address a fundamental and universal challenge in long-term memory systems: **maintaining context efficiency and reducing interference.**

1. Distinction from Previous Settings, e.g. Knowledge Updating:

While prior work exists in "Knowledge Updating" (focusing on overwriting old facts with new ones), our work uniquely emphasizes the explicit removal of non-essential information to free up cognitive space. We view this work as a foundational step. We are actively exploring methodologies to extend this mechanism to more complex, real-world historical data in our future work.

2. Theoretical Necessity (Beyond "Boutique"):

For any memory system—biological or artificial—capacity is finite. So the ability to autonomously discard (or "selectively forget") outdated information is not a niche requirement but a prerequisite for keeping memory concise and robust. Our task serves as a proxy to evaluate this critical capability.

3. Justification for the Controlled Setting:

We acknowledge that the current task setup involves synthetic elements. As it is very hard to construct a naturalistic dataset with long-term history (like more than 100K tokens) and precise ground-truth annotations for what should be forgotten is inherently difficult and often ambiguous.

4. Feasibility of the Task:

We would like to highlight that our proposed task, although somewhat synthetic, is definitely valid and solvable. As demonstrated in our ablation study (Section 4.3.4), agents(O4-mini) equipped with strong reasoning capabilities perform exceptionally well on short-context(6K tokens) versions of the dataset. This suggests that the current struggle with selective forgetting in long-history inputs stems not from the task definition, but from the limitations of current agents in long-range reasoning—specifically, their inability to accurately discern which error information should be forgotten.

> **W3:** The "test-time-learning" title seems a bit of a misnomer, since the actual task is just learning from long histories/context.

Response to W3: Justification for the Term "Test-Time Learning"

We acknowledge the reviewer's point that "Test-Time Learning" (TTL) in our benchmark relies on the mechanism of learning from long histories (or In-Context Learning). However, we believe the terminology is appropriate for the following reasons:

1. Skill Acquisition vs. Information Retrieval:

We deliberately chose the term "Learning" to distinguish these tasks from others like "Long-Range Understanding" or "Retrieval." In tasks like Accurate Retrieval, the agent fetches a static fact. In our TTL tasks (e.g., Banking77, Movie Recommendation), the agent must induce patterns and classification rulesfrom the dialogue history and apply them to novel inputs. This mirrors the definition of learning (updating behavioral policy based on experience) occurring at test time.

2. Operationalizing Online Learning:

We agree with the reviewer that a fully dynamic TTL setting would ideally involve an interleaved loop of (memory update $\rightarrow$ retrieval $\rightarrow$ feedback $\rightarrow$ update). However, evaluating such dynamic loops introduces significant noise and makes it difficult to isolate memory capacity from reasoning errors. To provide a robust and reproducible metric, we simplified this process into a two-step protocol:

- Acquisition Phase: The agent processes a sequence of labeled examples (simulating the "learning" process over time).
- Evaluation Phase: The agent is tested on its ability to apply the learned rules.

3. Foundational Nature:

Given that research on benchmarking Agentic Memory is still in a nascent stage, we view our current setup as a foundational operationalization of TTL designed to facilitate reproducibility and ensure a stable evaluation environment for the community.While this protocol simplifies the 'online' aspect into a sequential batch for stability, it successfully captures the core competence: *can the agent improve its performance on a specific task solely by 'memorizing' past interactions?* We plan to extend this to more complex, fully interleaved online learning scenarios in future work."

Rebuttal to Reviewer nPYL (Part3)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 02:45 (modified: 25 Nov 2025, 02:45) Everyone

Revisions (/revisions?id=sCHmmMHJLF)

Comment:

> **Q2:** Is it possible to have a simple clear motivating example that shows where existing benchmarks fail to capture a specific particular way modern "memory agents" work ?

Response to Q2:

Thank you for this excellent suggestion. We agree that a concrete, simplified example is essential to intuitively demonstrate why existing benchmarks fall short of evaluating modern memory agents but ours could evaluate them comprehensively.

1. Systematic Analysis of Existing Benchmarks

Beyond this specific example, we categorize the limitations of existing benchmarks as follows:

A. Long-Context Benchmarks (e.g., LongBench, HELMET)

Static vs. Dynamic: These benchmarks provide the entire context in a single block (static), whereas memory agents should accumulate knowledge interactively (incremental/chunking).

Competency Coverage: Even if adapted to an incremental format, they do not cover all four core competencies. For example, HELMET, despite being comprehensive, only covers Accurate Retrieval (AR), Test-Time Learning (TTL), and Long-Range Understanding (LRU), but **lacks** Selective Forgetting (SF), which is crucial for handling dynamic updates like the example above.

B. RAG Benchmarks (e.g., RGB, CRUD-RAG)

Limited Scope: These focus almost exclusively on Accurate Retrieval (AR).

Agentic Gap: A memory agent is not just a search engine; it must perform higher-order reasoning such as learning from examples (TTL) and resolving conflicts (SF), which standard RAG benchmarks do not evaluate.

C. Previous Memory Agent Benchmarks

While some previous benchmarks attempt to evaluate memory agents, they are either outdated in scale or incomplete in scope:

Insufficient Context Depth: Benchmarks like MemoryBank and LoCoMo have context depths of only ~5k-10k tokens. This is insufficient for evaluating modern LLM backbones that support 128k+ or 1M+ context windows.

Incomplete Competencies: RealTalk and LongMemEval scale to longer contexts (100k+), but they primarily assess AR and TTL (e.g., simulating a persona), ignoring global reasoning (LRU) and memory updates (SF). StoryBench theoretically includes LRU, but the dataset is not publicly available.

2. Comparison Table on previous Memory Agent Benchmarks.

Table 1 summarizes how our work MemoryAgentBench is the first to cover all four competencies with significant context depth.

Table 1: Comparison of Memory Agent Benchmarks

Benchmark	Number of Question	Context Depth	Evaluation Competency
MemoryBank	194	5k	AR, TTL
LoCoMo	7512	10k	AR
PerLTQA	8593	1M*	AR
LongMemEval	500	115k, 1.5M	AR, TTL
RealTalk	728	375k*	AR, TTL
StoryBench	86	-	AR, LRU
MemoryAgentBench	2071	103k-1.44M	AR, TTL, LRU, SF

Note: AR=Accurate Retrieval, TTL=Test-Time Learning, LRU=Long-Range Understanding, SF=Selective Forgetting. (*) denotes estimated context depth.

3. A Motivating Example from Selective Forgetting.

Previous memory agent benchmarks have typically covered only one or two of the four competency categories, leaving the competencies like 'Selective Forgetting' unaddressed. Therefore, we provide a motivating example of 'Selective Forgetting' here and will include additional examples in the camera-ready version.

Consider a user interacting with a personal assistant agent over the course of a month regarding her eating habit (An example from Figure 1 in our paper).

- Session 1 (Day 1): User says, "Hi. ...My favorite fruit is the pear..."

- Session 10 (Day 15): User says, "Hello.....I recently found out that I mistyped the fruit that I like. Actually I love peas..... "
- Session 50 (Day 30): User asks, "Based on what you know about my eating habit, what is my favorite fruit?"

From the perspective of data processing and input protocols. For this same interaction history, existing long context benchmarks typically concatenate the entire sequence into a single long text for evaluation, whereas RAG benchmarks generally segment historical data from different days into separate documents to construct vector indices for retrieval. However, both evaluation protocols overlook the dynamic nature and complexity inherent in real-world long-term history. This highlights a critical gap: **memory is not just about storage (Long-Context) or retrieval (RAG), but about management.** Without evaluating the ability to discard obsolete information (Selective Forgetting), we cannot truly assess a memory agent's effectiveness in dynamic environments. Our work MemoryAgentBench is uniquely designed to capture these higher-order reasoning failures.

Official Review of Submission15836 by Reviewer zBBu

Official Review by Reviewer zBBu 31 Oct 2025, 19:48 (modified: 11 Nov 2025, 21:41) Everyone

Revisions (/revisions?id=uR3qANeEDh)

Summary:
This paper introduces MemoryAgentBench, a benchmark suite designed to systematically evaluate memory agents along four core competencies: accurate retrieval, test-time learning, long-range understanding, and selective forgetting. This benchmark repurposes existing long-context datasets and introduces two new datasets to address shortcomings in existing memory benchmarks. A wide range of agents, including long-context models, RAG-based agents, and commercial memory systems, are evaluated using a unified protocol. The results highlight that current memory agents exhibit significant limitations, particularly in selective forgetting and long-range understanding.

Soundness: 3: good
Presentation: 3: good
Contribution: 3: good
Strengths:

- The paper tackles a timely task and an under-evaluated aspect on LLM-based agents. It introduces a broad, multi-faceted benchmark that evaluates memory beyond traditional long-context QA, encompassing lesser-explored but crucial capabilities like selective forgetting.
- Introduces two new datasets, EventQA and FactConsolidation, to evaluate accurate retrieval and selective forgetting. Selective existing long-context datasets are carefully reconstructed into multi-turn interactions, simulating more realistic memory use in incremental settings.
- Comprehensive empirical evaluation across diverse agent architectures under a unified framework.

Weaknesses:

- Most benchmark tasks are language-focused QA or summarization; it lacks memory evaluations in action-heavy or procedural domains (e.g., coding tasks like SWE-Bench or tool-use environments).
- While covering four competencies, some of the chosen evaluation tasks may inherently favor certain types of models, raising questions of whether all approaches are on equal footing. For instance, the test-time learning tasks (e.g. few-shot text classification and movie recommendation through a long dialogue) essentially test in-context learning ability. Large context models with expansive memory windows can simply absorb all provided examples or dialogue turns, whereas retrieval-based agents might struggle to learn new tasks since they mainly fetch past knowledge. The result that long-context models outperform others on these TTL tasks is unsurprising, but it underscores that the benchmark might be conflating memory length with learning ability. In practice, an agent with a smaller context but a clever memory update algorithm might still be disadvantaged in such a setup. It just shows that current RAG systems are not very good at adapting. But it also suggests that the TTL evaluation might be favoring raw context length over true adaptation. A related concern is whether the information or skills being learned at test time were truly unknown to the base model; if not carefully controlled, stronger LLMs might answer correctly without genuine on-the-fly learning.

Questions:

1. For test-time learning, did you measure a baseline performance without providing any in-context examples, to confirm that it improves significantly only when it learns from the given in-context examples? Understanding how novel the tasks are to the model would help interpret whether improved performance indeed reflects on-the-fly learning and not just prior knowledge or spurious pattern-matching.
2. Can the authors also report the associated API costs for conducting different experiments? This will offer insights on the practical limitations of each agent architecture under the benchmark conditions.
3. Minor Typo: Line 91: “fuor” to “four”

Flag For Ethics Review: No ethics review needed.
Rating: 6: marginally above the acceptance threshold. But would not mind if paper is rejected
Confidence: 4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.
Code Of Conduct: Yes

Rebuttal to Reviewer zBBu (Part1)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 01:51 Everyone

Comment:

> **W1:** Most benchmark tasks are language-focused QA or summarization; it lacks memory evaluations in action-heavy or procedural domains (e.g., coding tasks like SWE-Bench or tool-use environments).

Response to W1:

We acknowledge the reviewer’s valid point regarding action-heavy environments (e.g., SWE-Bench). While extending to such domains is a promising future direction, we deliberately focused on language-focused tasks in our Benchmark for three methodological reasons:

- **Decoupling Memory from Reasoning:** Action-heavy tasks introduce significant confounding factors, such as planning deficits or execution errors. By focusing on language-centric tasks, we isolate and rigorously measure pure memory competencies (e.g., Accurate Retrieval, Selective Forgetting) without the "noise" of reasoning failures inherent in complex environments.
- **Coverage of Procedural Learning:** Our Test-Time Learning (TTL) competency effectively serves as a proxy for procedural memory. In tasks like multi-class classification, agents must learn "rules" and "procedures" from historical interactions. This captures the cognitive core required for action tasks internalizing procedural knowledge without the complexity of a code execution environment.
- **Question Answering as a Prerequisite:** We view standard QA as the prerequisite of agentic capabilities. Tasks like Accurate Retrieval in our benchmark test the fundamental memory operations required for the real-world complex downstream application. If an agent cannot reliably answer these basic memory-specific questions, it lacks the foundational capacity to function in complex action environments with the long-term memory, making evaluations in such domains premature.

We will explicitly discuss this distinction and highlight action-heavy evaluation as a key area for future work in the final version.

Rebuttal to Reviewer zBBu (Part2)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 01:55 (modified: 25 Nov 2025, 02:15) Everyone

Revisions (/revisions?id=29PbAehwOz)

Comment:

> **W2 :** While covering four competencies ..... A related concern is whether the information or skills being learned at test time were truly unknown to the base model; if not carefully controlled, stronger LLMs might answer correctly without genuine on-the-fly learning.

> **Q1:** For test-time learning, did you measure a baseline performance without providing any in-context examples, to confirm that it improves significantly only when it learns from the given in-context examples? Understanding how novel the tasks are to the model would help interpret whether improved performance indeed reflects on-the-fly learning and not just prior knowledge or spurious pattern-matching.

Response to W2 & Q1:

We thank the reviewer for scrutinizing the fairness and validity of the TTL competency. We address the concern regarding "prior knowledge" (Q1) and "context length bias" (W2) together, as they both pertain to the interpretation of TTL results.

1. Addressing Prior Knowledge & Genuine Learning (Response to Q1 & W2)

To verify that performance improvements stem from on-the-fly learning rather than prior knowledge, we conducted zero-shot evaluations (without in-context examples) as a baseline.

- **Results:** As shown in the **Table 1** below, the zero-shot performance on tasks like Multi-Class Classification (MCC) and **Movie Recommendation** was significantly lower compared to the full-memory setting.
- **Conclusion:** This stark contrast confirms that the base models do not possess sufficient prior knowledge to solve these specific long-tail tasks. The performance gains are explicitly driven by the agent's ability to access and utilize the provided historical examples.

Table 1: Performance on Test-Time Learning (TTL) tasks with zero-shot setting. Compared to the full memory setting, we also reported the performance degradation under the zero-shot setting.

	MCC	Recom.	Avg.
GPT-4o-mini	0.6	6.1	3.4
GPT-4.1-mini	0.8	5.7	3.3
Gemini-2.0-Flash	0.0	5.5	2.8
gpt-4o-mini w/ context	82.0	15.1	48.6

2. Addressing Fairness between Long-Context vs. RAG (Response to W2)

The reviewer raises a valid point that long-context models (LC-LLMs) naturally excel at TTL by fitting all examples into their window, while RAG systems struggle. However, we argue this is not a bias of the benchmark, but a revelation of a critical limitation in current RAG systems.

- **The Failure Mode:** The poor performance of RAG agents on TTL tasks highlights that current retrieval-based methods only retrieve part of the history and usually struggle with the global implicit understanding of the history. This limitation extends to current memory systems as most advanced memory systems (such as Mem0[1], Letta[2], MIRIX[3]) heavily rely on retrieval of past history to perform downstream tasks.
- **Benchmark Goal:** Our goal is to evaluate the end-to-end competency of the memory system. If a memory mechanism (like standard RAG) cannot support the information consolidation needed for in-context learning, it is objectively a less capable memory system for that specific competency. This

result encourages the community to develop better memory systems to power self-evolving or self-learning agents as putting everything in context will always break the context window when the task requires long-term reasoning.

References

[1] Chhikara, Prateek, et al. "Mem0: Building production-ready ai agents with scalable long-term memory." arXiv preprint arXiv:2504.19413 (2025).

[2] Letta team. “Letta (formerly MemGPT): Platform for Building Stateful Agents.” GitHub repository (2025).

[3] Wang, Yu, and Xi Chen. "Mirix: Multi-agent memory system for llm-based agents." arXiv preprint arXiv:2507.07957 (2025).

Rebuttal to Reviewer zBBu (Part3)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 02:14 Everyone

Comment:

> **Q2:** Can the authors also report the associated API costs for conducting different experiments? This will offer insights on the practical limitations of each agent architecture under the benchmark conditions.

Response to Q2:

We thank the reviewer for highlighting the importance of cost analysis for real-world deployment. We have conducted a detailed cost estimation for inference across four representative datasets (MH-Doc QA, MCC, Detective QA, FC-SH) that vary in context length and reasoning complexity.

1. Methodology:

To provide a realistic assessment of "practical limitations," our cost calculation assumes Context Caching is enabled (a standard feature in modern APIs like OpenAI), which significantly reduces the cost for Long-Context models processing shared histories. We compare three architectures: Long-Context (LC), RAG Agents, and the Agentic approach (MIRIX).

2. Table Caption & Description

Cost-Performance Analysis on Representative Tasks. We report the amortized inference cost per query (USD) and the corresponding performance metric (Accuracy/Score) in **Table 2**.

- **Pricing Basis:** Costs are calculated based on OpenAI's latest pricing (Nov. 2025): GPT-4o-mini (0.15 USD / 1M input, 0.60 USD / 1M output) and GPT-4.1-mini (0.40 USD / 1M input, 1.60 USD / 1M output).
- **Context Caching:** We apply Cached Input pricing (50% discount for 4o-mini, 75% discount for 4.1-mini) for Long-Context Agents, assuming sequential questioning on shared contexts.
- **Settings:** RAG agents use Top-K=10. Embedding indexing costs are excluded as one-time expenses.

Table 2: Estimated Amortized Cost ($) vs. Performance per Query

		MH-Doc QA	MCC	Detective QA	FC-SH
GPT-4o-mini	Est. Cost (USD)	$0.01	$0.008	$0.01	$0.01
	Performance	43.0	82.0	63.4	45.0
GPT-4.1-mini	Est. Cost (USD)	$0.043	$0.011	$0.013	$0.027
	Performance	66.0	75.6	56.3	36.0
RAG Agents (BM25 + 4o-mini)	Est. Cost ($)	<$0.001	$0.006	$0.006	<$0.001

		MH-Doc QA	MCC	Detective QA	FC-SH
	Performance	56.0	75.4	52.1	48.0
MIRIX (4.1-mini)	Est. Cost ($)	$0.016	$0.010	$0.011	$0.019
	Performance	75.0	61.0	62.0	20.0

Note: Costs are amortized over the question set sharing the same context. Performance scores are metrics as defined in the paper.

Key Insights on Practical Limitations:

- **The "Efficiency-Reasoning" Trade-off in RAG:** RAG agents like BM25 are cost-efficient (< 0.001 USD – 0.006 USD) but suffer in tasks requiring global reasoning (e.g., Detective QA, scoring 52.1 vs. 63.4 for LC). This limits their utility in complex analytical scenarios.
- **Cost Prohibitiveness of Long-Context Scaling:** While powerful, Long-Context models face steep cost increases with stronger backbones. Upgrading from 4o-mini to 4.1-mini roughly quadruples the cost (0.010 USD → 0.043 USD for MH-Doc QA), making high-end long-context deployment expensive even with caching.
- **Agentic Memory (MIRIX) as the Optimal Middle Ground:** A crucial finding is that MIRIX (with 4.1-mini) achieves a lower cost (0.016 USD) than the raw GPT-4.1-mini Long-Context (0.043 USD) on memory-intensive tasks like MH-Doc QA, while delivering superior performance (75.0 vs. 66.0). This demonstrates that agentic memory mechanisms could successfully decouple performance from linear context costs, offering a scalable solution for high-performance applications.

We will include this detailed cost-performance breakdown in the final version to assist future researchers in selecting architectures based on their budget and accuracy requirements.

> **Minor Typo:** Line 91: “fuor” to “four”

Thanks for pointing out this issue. we will revise it in our next version paper.

Official Review of Submission15836 by Reviewer Mcjo

Official Review by Reviewer Mcjo 30 Oct 2025, 02:30 (modified: 11 Nov 2025, 21:41) Everyone

Revisions (/revisions?id=gc1zCo7ekH)

Summary:
This paper introduces MemoryAgentBench, a new benchmark for evaluating the memory management capabilities of large language model agents in multi-turn interactions. The authors outline four essential competencies for memory agents: accurate retrieval, test time learning, long range understanding, and conflict resolution. To test these competencies, the authors adapt several existing datasets and create two new ones, EventQA and FactConsolidation, to expose weaknesses in current memory systems. They evaluate a wide range of memory types, including context-based models, retrieval augmented systems, and agentic memory agents. The results show that while some methods perform well on isolated tasks, no agent performs consistently across all four competencies. This suggests that memory management remains a significant open challenge for the next generation of intelligent agents.

Soundness: 4: excellent
Presentation: 3: good
Contribution: 3: good
Strengths:

1. Compared to previous benchmarks, this paper has a more comprehensive advantage in terms of data token length, problem diversity, and evaluation diversity.
2. The paper presents a comprehensive and systematic framework that evaluates four memory-related competencies of LLM agents, including a novel aspect of Selective Forgetting. This enables consistent testing

across a variety of scenarios by combining existing and newly created datasets.
3. The paper conducts a broad comparison across various types of memory under a consistent evaluation protocol, clarifying the strengths and weaknesses of each approach, and also provides detailed analyses with ablation studies on various memory settings, offering practical insights into the efficiency and scalability of memory construction and query processing for agents.
4. The paper is clearly written and well-structured, with informative tables and figures. The inclusion of detailed prompts and ablations supports reproducibility.

Weaknesses:
See questions.

Questions:
1. How confident should we be that the evaluation metrics and the usage of LLM-as-a-judge in this paper are appropriate (e.g., for summarization or recommendation)?
2. Why does this benchmark primarily use chunking as the main interaction method between the user and the Memory agent? Does this accurately reflect real-world interaction scenarios?
3. Could you explain the difference between the dialogue-based task designed in the article and the direct document question answering task from the perspectives like task definition and model performance?
   For example, considering the document Question Answering task in the Accurate Retrieval category, what is the essential difference between transforming it into a dialogue and directly answering based on the original dataset? Is there a difference in the downstream model's performance?

Flag For Ethics Review: No ethics review needed.
Rating: 8: accept, good paper (poster)
Confidence: 3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
Code Of Conduct: Yes

Rebuttal to Reviewer Mcjo

Official Comment
by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))
25 Nov 2025, 01:20 Everyone

Comment:

> **Q1.** How confident should we be that the evaluation metrics and the usage of LLM-as-a-judge in this paper are appropriate (e.g., for summarization or recommendation)?

Reply:

Thanks for pointing out your concern on LLM-as-a-judge. We agree that model-based evaluation may not be the most appropriate choice for our specific task context. Here we adopted model-based scoring on both LongMemEval and ∞Bench-Sum in order to remain consistent with prior work [1,2]. In LongMemEval, the questions admit clear, objective ground-truth answers with minimal subjectivity. In fact, Wu et al. [1] report in Table 6 that a prompt-engineered GPT-4o judge achieves 98.0% accuracy, demonstrating very high stability. Likewise, Yen et al. [2] validate in Appendix B.6 of the HELMET benchmark that GPT-4o judgments “mostly align” with human evaluations. These findings suggest that model-based evaluation could be reasonably good so that it can reflect the human evaluation to some extent.

We also acknowledge that incorporating human evaluations or measuring inter-rater agreement can yield more accurate performance assessments. Human evaluations might be costly, but inter-rater agreement is an excellent suggestion and we will sincerely consider that in our next version.

> **Q2.** Why does this benchmark primarily use chunking as the main interaction method between the user and the Memory agent? Does this accurately reflect real-world interaction scenarios?

Reply:

In practical applications, memory-enabled agents—such as personalized memory assistants—must handle more than just conversational data. Real-world inputs may arrive through diverse media modalities. While certain benchmarks (e.g., the NIAH task) are synthetic, complete narratives—such as full novels—provide a more realistic testing ground. This setting can be used to simulate the scenario where the personal assistant is observing the user reading a book (for instance, from AI glasses or Desktop Monitors), they might be seeing the books and they also need to remember them for future discussion with the user.

As for the chunking, we acknowledge that real-world personal assistant may have streaming input (e.g. 24 seconds per second like human eyes). However, in practice, it is necessary to quantize real-world inputs before feeding them into a language model (e.g. we may accumulate 24 images after one second and make one call). Thus inputting chunks into the agent is a natural strategy to handle streaming input in real worlds. To this end, we believe adopting a chunking-based input format could more faithfully reflects real-world conditions. We will add more clarifications about chunking design in our paper.

> **Q3.** Could you explain the difference between the dialogue-based task designed in the article and the direct document question answering task from the perspectives like task definition and model performance?

Reply:

Our paper emphasizes the multi-turn nature of user–agent interactions.

For a long-context baseline, the difference is minimal: one can simply concatenate all past dialogue or document content into a single context window. In this case, transforming the task into a dialogue format versus direct document QA makes little practical difference.

However, for memory-based agent systems like MIRIX[3], Letta[4], or Mem0[5], the distinction becomes crucial. These systems are designed to process information incrementally — receiving fragmented inputs over time, much like real-world user interactions where information arrives sequentially rather than all at once. This setup introduces new challenges: maintaining compactness of stored representations, organizing the information from fragmented inputs, maintaining compact memory even though the input is fragmented, and reasoning over temporally distributed information.

In short, while for long-context baselines the two settings are effectively equivalent, for memory agents the dialogue-based formulation is essential, as it better reflects realistic conditions and tests the agent's ability to organize and utilize information accumulated over multiple turns.

References

[1] Di Wu et al., "LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory," ICLR 2025 (Table 6).

[2] Howard Yen et al., "HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly," arXiv:2410.02694 (Appendix B.6).

[3] Wang, Yu, and Xi Chen. "Mirix: Multi-agent memory system for llm-based agents." arXiv preprint arXiv:2507.07957 (2025).

[4] Letta team. "Letta (formerly MemGPT): Platform for Building Stateful Agents." GitHub repository (2025).

[5] Chhikara, Prateek, et al. "Mem0: Building production-ready ai agents with scalable long-term memory." arXiv preprint arXiv:2504.19413 (2025).

Official Review of Submission15836 by Reviewer RnwS

Official Review by Reviewer RnwS 28 Oct 2025, 14:30 (modified: 11 Nov 2025, 21:41) Everyone

Revisions (/revisions?id=nhV045liUm)

Summary:

The paper proposes MemoryAgentBench, a benchmark for LLM memory agents that emphasizes incremental and multi-turn information intake. It operationalizes memory along four competencies: Accurate Retrieval, Test-Time Learning, Long-Range Understanding, and Selective Forgetting, grounding these choices in classic memory science. It converts or reconstructs several long-context datasets into chunked, time-ordered interactions and adds two new datasets, EventQA and FactConsolidation, to close gaps around temporal recall and conflict resolution. The study evaluates long-context buffers, three classes of RAG, and agentic memory systems including commercial offerings, under a unified protocol.

This paper offers an important contribution because it reframes memory evaluation around how agents actually encounter information, chunk by chunk, across sessions, rather than flooding a context window and calling it memory. The four-competency map separates skills that are often conflated, the datasets stress realistic patterns like temporal ordering and conflict updates, and the results give a clean and actionable picture. The framework is reproducible and extensible, so it can anchor future work on agent memory.
Soundness: 3: good
Presentation: 3: good
Contribution: 3: good
Strengths:
The conversion of long-context resources into incremental dialogues is the key design choice and it is done consistently. The inclusion of commercial memory agents alongside RAG and long-context baselines provides a market-realistic snapshot.

EventQA probes temporal recall under narrative structure, and FactConsolidation isolates resolution and forgetting. The dataset validation with shorter contexts is a nice touch that shows solvability in principle and surfaces context-length brittleness in practice.

Weaknesses:
Some claims would be stronger with strict compute-matched comparisons across families. For example, matching total retrieved tokens, prompt tokens, number of model calls, and wall-clock between a tuned RAG loop and a long-context buffer would reduce confounds when declaring winners per competency. Current results trend in the right direction but still leave room for “budget effects.”

Moreover, the benchmark tells agents to “memorize” chunks. The exact prompt templates and any guardrails for overwriting or abstention deserve more foregrounding in the main paper, since small prompt deltas can shift Selective Forgetting outcomes. Much of this is in the appendix; a concise main-text summary would help.

Questions:
(1) Under equalized prompt tokens, generated tokens, and forward passes, how do the best RAG, best long-context, and best agentic memory settings compare on TTL and LRU?

(2) Can you expose the exact overwrite prompts and test a tiny policy grid, for example “always prefer later facts” or “prefer later only if explicit negation”?

Flag For Ethics Review: No ethics review needed.
Details Of Ethics Concerns:
The benchmark uses license-compliant corpora and releases safety-screened artifacts, with no personal data.

Rating: 6: marginally above the acceptance threshold. But would not mind if paper is rejected
Confidence: 4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.
Code Of Conduct: Yes

Rebuttal to Reviewer RnwS (Part 1)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 01:31 (modified: 25 Nov 2025, 02:10) Everyone

Revisions (/revisions?id=qjjTXB1tdo)

Comment:

> **W1** Some claims would be stronger with strict compute-matched comparisons across families. For example, matching total retrieved tokens, prompt tokens, number of model calls, and wall-clock between a tuned RAG loop and a long-context buffer would reduce confounds when declaring winners per competency. Current results trend in the right direction but still leave room for "budget effects."

> **Q1** Under equalized prompt tokens, generated tokens, and forward passes, how do the best RAG, best long-context, and best agentic memory settings compare on TTL and LRU?

Response to W1 & Q1: Strict Compute-Matched Comparisons.

We appreciate the reviewer's rigorous suggestion. We agree that "budget effects" confound the comparison between architectures. To address W1 and Q1, we conducted a strict compute-matched ablation study on Banking77 (TTL) and Summarization (LRU) using the best backbone (GPT-4.1-mini). We report the experiment results on **Table 1**.

1. Experimental Setup:

- Low (~4K) and Medium (~40K): Constrain Long-Context (LC) to truncated contexts. Limit RAG and Agentic Method (MIRIX) to Top-K chunks trying to match token counts.
- High : Scaling up RAG and Agentic Method (MIRIX) to retrieve High-TopK chunks to match the LC full-context budget.
- Subset: For Book Summarization, we evaluate on a random 30-book subset.

Table 1: Experiment results on TTL *(Banking77)* and LRU *(Summ.)*. (Accuracy/Score %)

Task	Budget Setting	Long-Context (GPT-4.1-mini)	RAG (BM25)	Agentic (MIRIX)
	Low (~4K / Top-K=1)	74.0	83.0	52.0
TTL *(Banking77)*	Medium (~40K / Top-K=10)	90.0	89.0	65.0
	High (~104K)	93.0	88.0	67.0
	Low (~4K / Top-K=1)	8.2	7.9	8.4
LRU *(Summ.)*	Medium (~40K / Top-K=10)	16.4	15.8	18.7
	High (~113K)	39.7	38.0	38.8

We analyze the Budget Effects on TTL and LRU. We summarize our results as follows:

2. TTL: The Efficiency-Capacity Trade-off.

- **Low-Budget Dominance of RAG:** At the Low (~4K) budget, RAG (83.0) significantly outperforms LC (74.0). This confirms that RAG possesses superior structural efficiency for pattern matching. It precisely retrieves relevant examples while LC suffers heavily from truncation.
- **Crossover & Saturation:** As the budget increases to Medium (~40K), performance equalizes (~90.0). At High (~104K), LC scales to 93.0, whereas RAG saturates and slightly degrades (88.0) due to retrieved noise. This demonstrates that while RAG is efficient, LC has a higher capacity ceiling when the budget permits.

3. LRU: The Information Threshold Effect.

- **Threshold Behavior:** For global reasoning, performance is binary. At Low and Medium budgets, all architectures fail (<20.0), confirming that partial information is insufficient for summarization regardless of the method.
- **Convergence at High Budget:** Only at the High budget (full text) do models succeed (~39.0), with RAG catching up to LC. This proves that for LRU, success is determined by meeting the information threshold (processing the full text), not by the architecture itself.

4. Forward Passes & Generation Costs.

Regarding the reviewer's concern on "forward passes" and "generated tokens," we argue that strict equality in pass counts (e.g., forcing Agentic models to 1 pass) would strip them of their reasoning capability. Instead, we compare Total Compute Load:

- Long-Context (1 "Big" Pass): Processes a massive sequence (e.g., ~100K tokens) in a single pass.
- Agentic (Many "Small" Passes): Uses multiple loops (e.g., 5 passes), but each processes a condensed context (e.g., 2K~4K tokens).
- Comparison: Mathematically, the Agentic model's Total Processed Tokens (TPT) is often lower than LC's single massive pass (e.g., $5\times4K \ll 1\times100K$). Thus, Agentic models (MIRIX) operate under a competitive, often lower, total compute budget despite using more passes. Their performance in **Table 1** reflects the different computational allocation rather than unfair total compute usage.

Our controlled experiments reveal that RAG could be efficient (Very Low-Budget TTL), while Long-Context is the winner for capacity (High-Budget TTL). For global tasks (LRU), a high compute budget is an absolute prerequisite for all architectures. Furthermore, Agentic models justify their multiple forward passes by maintaining a lower Total Compute Load than full-context processing, trying to make a balance between reasoning depth and computational cost.

Replying to Rebuttal to Reviewer RnwS (Part 1)

Rebuttal to Reviewer RnwS (Part 2)

Official Comment

by Authors (Yu Wang (/profile?id=~Yu_Wang24), Julian McAuley (/profile?id=~Julian_McAuley1), Yuanzhe Hu (/profile?id=~Yuanzhe_Hu1))

25 Nov 2025, 01:41 Everyone

Comment:

> **W2** Moreover, the benchmark tells agents to “memorize” chunks. The exact prompt templates and any guardrails for overwriting or abstention deserve more foregrounding in the main paper, since small prompt deltas can shift Selective Forgetting outcomes. Much of this is in the appendix; a concise main-text summary would help.

Response to W2:

We thank the reviewer highlighting the importance of foregrounding our prompt design. We agree that in agentic evaluations, the prompt template could affect how memory should be updated or overwritten, and this transparency is vital for interpreting the results.

- Revisions in the Main Text: As suggested, we will add a specific paragraph in **Section 3.3 (Datasets and Agents Formulation)** titled *"Prompt Formulation and Interaction Protocol"*. The following is the full text of this section:
- **Prompt Formulation and Interaction Protocol.** Unlike standard long-context evaluations that input raw text, we wrap all input chunks within a simulated User-Assistant dialogue to explicitly trigger the agent's memory mechanism. Each input chunk c_i is preceded by a memorization instruction (e.g., "Please memorize ...") to establish a clear intent for information storage. Simultaneously, for each specific dataset, we carefully curated the instructions to ensure agents accurately comprehend the task intent and execute the required actions. Crucially, for the Selective Forgetting competency, we introduce explicit guardrails in the prompt. We explicitly instruct agents that facts are indexed by serial numbers, and that "*newer facts have larger serial numbers*". The agents are mandated to solve conflicts by finding the newest fact (see Appendix D for full templates).
- Robustness of the Evaluation: We clarify that while prompt deltas can indeed shift outcomes, our benchmark applies a unified and standardized prompt template across all evaluated agents (Long-context, RAG, and Agentic). This ensures that the performance gaps observed (e.g., the failure of RAG agents in multi-hop SF) are attributed to the limitations of their memory mechanisms rather than prompt inconsistency.

> **Q2** Can you expose the exact overwrite prompts and test a tiny policy grid, for example “always prefer later facts” or “prefer later only if explicit negation”?

Response to Q2:

We sincerely thank the reviewer for this constructive suggestion. To rigorously test whether explicit instructions can mitigate the forgetting/overwriting issue, we conducted additional ablation studies using the exact overwrite prompts on the GPT-4.1-mini baseline. We tested two settings: **Policy A ("Always Prefer Later")** and **Policy B ("Conservative/Explicit Negation")**.

- Key sentence used on Policy A (Always Prefer Later): "Crucial Rule: Treat the facts as a chronological update stream. If there is ANY conflict between facts, you must ALWAYS overwrite the earlier fact with the one having the larger serial number."
- Key sentence used on olicy B (Conservative/Explicit Negation): "Crucial Rule: Be conservative with updates. ONLY discard or overwrite an earlier fact if the fact with the larger serial number explicitly negates it or explicitly states the previous information is incorrect."

Table 2: Experiment results on Selective Forgetting (Accuracy %)

	FC-SH	FC-MH	Avg.
GPT-4.1-mini (Baseline)	36.0	5.0	20.5
GPT-4.1-mini (Policy A)	40.0	4.0	22.0 (+1.5 ↑)
GPT-4.1-mini (Policy B)	28.0	4.0	16.0

The results in **Table 2** reveal two critical insights:

- **Limited Generalization with Aggressive Updates:** While Policy A slightly improves performance on single-hop tasks (FC-SH score increases from 36.0 to 40.0), it fails to generalize to complex multi-hop reasoning (FC-MH drops to 4.0). This suggests that while prompting helps with simple fact retrieval, it cannot effectively propagate updates through reasoning chains.
- **Performance Degradation with Conservative Constraints:** Surprisingly, Policy B significantly degrades the average performance (-4.5 points). The complex conditional instruction (checking for "explicit negation") appears to increase the model's cognitive load or induce overly cautious behavior, preventing the retrieval of valid updates.

These findings serve as a sanity check that validates our motivation. They demonstrate that Selective Forgetting cannot be solved by prompt engineering alone.