BRANCHMEM: PERSISTENT CANDIDATE BRANCHES FOR LONG-HORIZON SEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon search agents accumulate evidence across retrieval rounds, form intermediate judgments, and use earlier findings to guide subsequent actions. As the interaction history grows beyond a single context window, memory must preserve a state that remains useful for future decisions. An executed search trajectory can support several unresolved candidate interpretations, each with its own intermediate judgments. We introduce BranchMem, a branch-structured memory framework that maintains each task-relevant candidate interpretation as a persistent branch. New observations can extend a branch or revise its local judgments; subsequent decisions read the candidate branches and their associated evidence to reassess unresolved interpretations. We evaluated BranchMem on all 221 MemoryArena Search task groups.

1 INTRODUCTION

Long-horizon search requires an agent to find candidates, connect evidence, and verify conditions over multiple retrieval rounds. Intermediate judgments guide the next query, while new observations can change earlier judgments. ReAct established a general pattern of interleaving reasoning with external actions (Yao et al., 2023b). MemoryArena studies how information acquired in one session informs subsequent decisions in interdependent tasks (He et al., 2026). In this setting, the agent's accumulated interaction history eventually exceeds the context available for a single decision. Long-context models can also be sensitive to where relevant evidence occurs (Liu et al., 2024). A persistent memory must carry forward evidence that helps the agent reconsider earlier interpretations when new clues arrive.

Existing systems organize this history in several ways. Mem0 extracts and consolidates salient information from interactions (Chhikara et al., 2025); ReSum periodically condenses long search trajectories to support continued exploration (Wu et al., 2025b). MAGE maintains hierarchical execution state through recent traces, subgoal summaries, and revision operations (Chen et al., 2026). ReTree associates search claims with their sources and repairs downstream state after confirmed conflicts (Yang et al., 2026). BeliefMem retains alternative conclusions and their evidence support under partial observability (Liao et al., 2026). These contributions establish memory as an evolving part of an agent's decision process.

An executed search trajectory can contain several competing interpretations. Consider a search for the unusual plant ingredient used in a later waffle study by an author of a 2021 paper on food-quality indicators. A retrieved review contains two plausible paper lineages. The first links a strongly title-matched paper through an author and a cookie study to spent coffee grounds. The second links another paper through a different author and a filled-waffle study to Jerusalem artichoke powder. Both lineages arise from the same document read and the same executed history. The agent initially follows the stronger title match. A later clue about the target paper's corresponding author favors the second lineage. The correct ingredient then follows from the second paper–author–study–ingredient chain.

This example exposes two representational requirements. A candidate that is temporarily less salient may become decisive after delayed evidence, so its supporting chain must remain available. The relations within each candidate must also survive memory updates: independently plausible paper and ingredient values need a common evidential chain before they support an answer. A memory state that retains only the initially favored lineage can lose a useful alternative. A state that retains

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

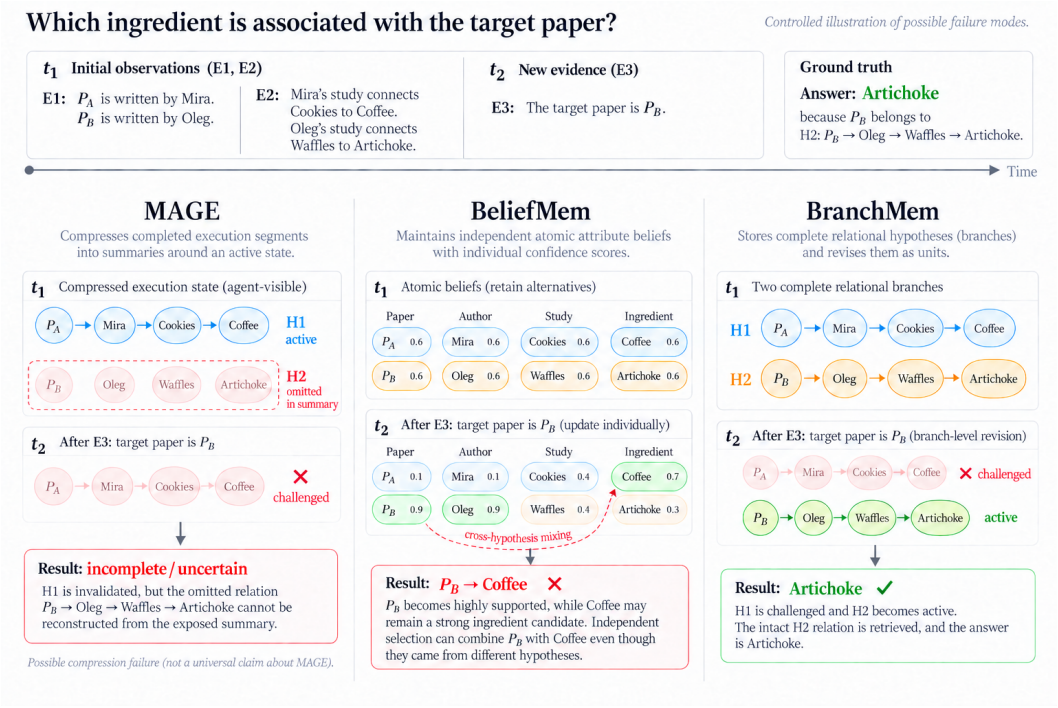


Figure 1: Controlled illustration of candidate retention and relational membership under delayed evidence. The same initial observations support two candidate chains. Later evidence selects the second chain. The panels illustrate how the available memory representation affects subsequent answer construction. Names and ingredient labels are simplified for display.

individual values without their candidate membership can combine the second paper with the first lineage's ingredient. Figure 1 illustrates these two risks in a controlled example.

We propose BranchMem, a persistent candidate-branch memory for long-horizon search. A branch groups a task-relevant candidate with local judgments, source references, and revision records. The search model can create or extend branches during ordinary interaction. When evidence challenges one judgment, the corresponding branch receives a local update while other recorded information remains available for later inspection. At the next stage, the agent reads the retained branches together with selected source excerpts and decides which conditions to verify. Multiple branches share the observed search history and the same tool budget.

Our evaluation covers all 221 MemoryArena Search task groups under a local, model-driven retrieval protocol. BranchMem answers 158 final questions correctly, a 71.49% success rate. We additionally inspect stage-level records to trace how earlier candidate information enters later queries and how local judgments change after new evidence. Our contributions are a persistent representation for candidate interpretations, an incremental update and readout interface integrated into search, and a full-collection evaluation accompanied by trace-level analysis.

2 Related Work

Interactive search and long contexts. ReAct interleaves reasoning and tool use so that observations guide later actions (Yao et al., 2023b). IRCoT alternates retrieval and reasoning for multistep questions (Trivedi et al., 2023). Effective use of a long context depends on the position and accessibility of relevant information, as demonstrated by Lost in the Middle (Liu et al., 2024). ReSum periodically summarizes progress to continue long-horizon search (Wu et al., 2025b). MemoRAG uses a global memory of a document collection to generate clues for subsequent retrieval (Qian et al., 2024). These approaches connect retained context to future information acquisition.

Agent memory also supports learning from earlier interactions. Reflexion stores verbal reflections on feedback for subsequent trials (Shinn et al., 2023); Voyager accumulates a library of executable skills (Wang et al., 2023). ReasoningBank extracts reusable reasoning strategies from successful and unsuccessful trajectories (Ouyang et al., 2025). Together, they illustrate memories of observations, feedback, and procedures that can influence later actions.

Persistent and relational memory. Generative Agents records experiences and reflections for planning (Park et al., 2023). MemoryBank maintains conversational memories with time- and importance-dependent updating (Zhong et al., 2023). MemGPT manages access across context and storage tiers (Packer et al., 2023); MemoryOS and MIRIX organize persistent information through storage levels and specialized memory types (Kang et al., 2025; Wang & Chen, 2025). Mem0 extracts and consolidates salient facts, with a graph variant for entity relationships (Chhikara et al., 2025). A-MEM dynamically indexes and links notes (Xu et al., 2025). SimpleMem combines structured compression, memory consolidation, and query-aware retrieval (Liu et al., 2026).

Graph structures organize connections across retrieved information. HippoRAG associates evidence through a knowledge graph (Jiménez Gutiérrez et al., 2024); HippoRAG 2 integrates passage information and online model processing to support factual and associative retrieval (Jiménez Gutiérrez et al., 2025). GraphRAG builds an entity graph and community summaries for corpus-level question answering (Edge et al., 2024). Zep maintains temporal entity relations (Rasmussen et al., 2025), while APEX-MEM combines an entity–event property graph, append-only records, and retrieval tools for evolving information (Banerjee et al., 2026). BranchMem organizes task-relevant relations around the candidate interpretations that remain under consideration during a search.

Execution state and revision. MAGE organizes recent trajectories and completed subgoals into hierarchical execution state, supporting growth, compression, maintenance, and revision (Chen et al., 2026). ReTree attaches evidence provenance to search state and repairs affected nodes and descendants after contradictory evidence (Yang et al., 2026). Tree of Thoughts explores intermediate reasoning paths, and LATS combines tree search with action and environmental feedback (Yao et al., 2023a; Zhou et al., 2024). These structures represent executed subgoals, evidence-dependent state, and exploration paths. BranchMem maintains coexisting interpretations that can arise inside one shared execution step.

Uncertainty and candidate judgments. Maintaining several explanations has a long history in symbolic reasoning. Assumption-based truth maintenance tracks supporting assumption sets and inconsistent environments while exploring multiple possible solutions (de Kleer, 1986). In language-model agents, BeliefMem retains attribute-level candidates and updates their evidence-support information (Liao et al., 2026). Belief-State Engine maintains explicit Bayesian beliefs under a specified partially observable environment model (Chattopadhayay & Halder, 2026). DeMem studies which distinctions compressed memory must preserve for future decisions (Zou et al., 2026). BranchMem uses model-proposed candidate records to preserve judgment membership, source references, and revision information for subsequent search decisions.

Evaluating memory over time. LoCoMo evaluates question answering, event summarization, and dialogue generation over long multi-session conversations (Maharana et al., 2024). LongMemEval examines extraction, multi-session and temporal reasoning, knowledge updates, and abstention (Wu et al., 2025a). MemoryAgentBench uses incremental interactions to assess retrieval, test-time learning, long-range understanding, and selective forgetting (Hu et al., 2025). MemoryArena places memory in interdependent agent tasks whose later actions use earlier interactions (He et al., 2026). This evaluation setting motivates our measurements of final success, stage progress, and candidate use across search stages.

3 BranchMem: Persistent Candidate Branches

BranchMem maintains candidate interpretations over a shared search history. The state retains both the observations produced by actual tool use and the task-relevant explanations that the agent has formed from those observations. This section defines the state, its incremental updates, and its readout.

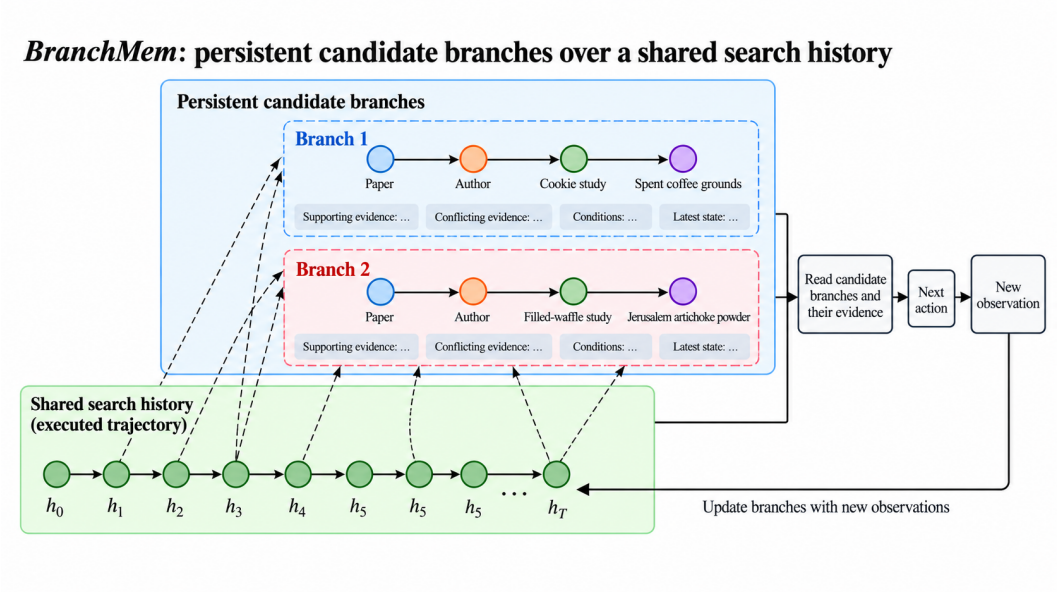


Figure 2: Overview of BranchMem. One shared action–observation history supplies evidence for persistent candidate branches. At stage boundaries, the agent proposes judgment and revision updates. Dashed gray links denote source references, and colored outlines indicate challenged and active judgments. Selected history, source excerpts, and visible branches are rendered together for the next search decision. Entity chains schematically depict relations expressed by the branch's natural-language judgments.

Figure 2 summarizes branch formation, local revision, and joint readout over a shared search history.

3.1 Search process and memory state

Let Q denote an overall question with related stages $q_1, \ldots, q_T$. Within stage t, the search agent chooses an action $a_{t,k}$ after an interaction trace $\tau_{t,<k}$. With persistent memory M_{t-1} from earlier stages, the action distribution is

$$a_{t,k} \sim \pi_\theta(\cdot \mid Q, q_t, \tau_{t,<k}, R(M_{t-1})), \tag{1}$$

where R renders memory for the agent. New tool observations enter the within-stage trace. At the end of a nonfinal stage, the agent may emit a memory update Δ_t alongside its stage answer $\hat{y}_t$:

$$M_t = U(M_{t-1}, q_t, \tau_t, \hat{y}_t, \Delta_t), \qquad t < T. \tag{2}$$

The persistent state is $M_t = (\mathcal{P}_t, \mathcal{E}_t, \mathcal{B}_t)$. $\mathcal{P}_t$ contains stage questions, answers, and queries; $\mathcal{E}_t$ contains selected records of retrieved sources; and $\mathcal{B}_t$ contains candidate branches. Different branches may cite the same source. They refer to one realized interaction history and share its tool budget.

3.2 Candidate branches and judgment membership

For candidate h, a branch at stage t is

$$b_t = (h, \rho_t, \ell_t, \mathcal{J}_t, \mathcal{V}_t), \tag{3}$$

where ρ_t describes the candidate's role in the task, ℓ_t explains its relation to the current or overall question, $\mathcal{J}_t$ contains local judgments, and $\mathcal{V}_t$ records revisions. A candidate can represent a target paper, a person, an answer entity, or a useful partial explanation. The collection can include related and overlapping entries.

A judgment $j = (p, s, D)$ stores a statement p, an evidence status s, and source references D. Statements preserve the relevant subjects, objects, and qualifiers in natural language. The status can be supported, contradicted, unresolved, or unverified. In the food-paper example from the introduction,

one branch can retain the links from the second paper to its author, the waffle study, and Jerusalem artichoke powder. The ingredient remains attached to the candidate whose study supplies its evidential context. Branches can also be partial: the agent may record a paper and an author before it finds their later study.

Source facts and task-level interpretations have different revision scopes. An observation that a paper was written by a particular author can remain valid after new evidence challenges that paper's identity as the target. BranchMem records local judgments separately from the branch's current applicability to the task.

3.3 Incremental formation and revision

The search model that chooses tool actions also proposes optional branch updates at stage boundaries. An update can create a candidate, add a judgment to an existing candidate, record a source, or revise the candidate's status. The recorder matches existing branches using normalized candidate text. The model supplies the natural-language relation to the task. If it emits no candidate update, previously recorded branches persist and the ordinary history and source records can still advance.

For judgments marked supported or contradicted, the recorder checks whether the cited source identifiers occur in the available retrieval or citation set. A judgment lacking a valid identifier is stored as unverified. The source–claim relationship is subsequently evaluated by the search model when it reads the source and branch together. Semantic validation is performed through that later model read.

Let $\Delta\mathcal{J}_t$ and $\Delta\mathcal{V}_t$ be judgment and revision updates to one branch. The recorder applies

$$\begin{aligned} \mathcal{J}_t &= \mathrm{AppendDedup}(\mathcal{J}_{t-1}, \Delta\mathcal{J}_t), \\ \mathcal{V}_t &= \mathrm{AppendDedup}(\mathcal{V}_{t-1}, \Delta\mathcal{V}_t). \end{aligned} \tag{4}$$

Unmentioned roles and status fields carry forward, and the latest revision record determines the branch's current status. Supported observations and later challenges can therefore remain visible together. A revision may address a single judgment, such as the associated film for an otherwise plausible person, or the applicability of an entire candidate to the task. The model assesses the implications of these records during later reads.

3.4 Joint readout for the next decision

At the start of a stage, BranchMem renders the selected history and sources together with retained candidate branches:

$$R(M_t) = \mathrm{Render}(\mathrm{Select}(\mathcal{P}_t, \mathcal{E}_t), \mathcal{B}_t)\,. \tag{5}$$

The source selection prioritizes referenced or actively inspected material and supplements it with search snippets. Each visible branch includes its candidate, relation to the task, judgments, source references, and latest revision. Reading these records together helps the agent inspect whether the evidence supports a connected explanation, identify a missing link, and decide whether to query, reopen a document, or answer. Tool use and final answer selection remain actions of the search model.

The evaluated capacity retains recent stage history and the most recently inserted candidate records. Updating an existing candidate preserves its insertion position. Appendix A gives the concrete limits. Algorithm 1 summarizes the stage loop.

4 Experiments

We evaluate final task completion, stage progress, and token consumption, then examine the contributions of candidate retention and revision information.

4.1 Tasks, protocol, and metrics

We use all 221 task groups of MemoryArena Search (He et al., 2026), containing 1,420 intermediate stages and 221 final stages. Each stage provides the overall question and its current question. All methods use DeepSeek V4.1 Flash with thinking disabled, the same task collection, local

Algorithm 1 BranchMem across search stages

Require: Overall question Q, stage questions $q_{1:T}$, search policy π_θ
Initialize $\mathcal{P}, \mathcal{E}, \mathcal{B}$ as empty
for $t = 1, \ldots, T$ **do**
  $C \leftarrow R(\mathcal{P}, \mathcal{E}, \mathcal{B})$
  $(\hat{y}_t, \tau_t, \Delta_t) \leftarrow \mathrm{Search}(\pi_\theta, Q, q_t, C)$
  if $t = T$ **then**
    Save the run record and **return** $\hat{y}_t$
  end if
  Record the question, answer, queries, and sources in $\mathcal{P}, \mathcal{E}$
  for each candidate update $\delta \in \Delta_t$ **do**
    Validate judgment statuses and available source identifiers
    Match or create a branch by normalized candidate text
    Append judgments and revision records with duplicate handling
  end for
  Retain the configured number of recent history and candidate records
end for

Table 1: Results on all 221 MemoryArena Search groups. Tokens include search and memory-maintenance input and generation over each complete task trajectory.

Method	SR (%)	PS (%)	Tokens/group (k)
Long context	63.35	49.67	5937.00
ReasoningBank	52.49	36.58	1887.00
MemoryOS	53.85	38.12	1796.00
Mem0	54.30	35.30	2414.00
HippoRAG 2	55.20	39.44	2259.00
MemoRAG	61.09	46.35	3506.00
BeliefMem	60.18	47.92	1207.00
MAGE	64.25	48.73	1641.00
BranchMem	**71.49**	**59.23**	**812.59**

SQLite full-text retrieval index, and search budgets. Searches return up to ten hits with 256 retrieval-interface tokens per snippet. Each stage permits ten new search calls, twenty document operations, and thirty interaction iterations. BranchMem reads up to twelve stage records and three candidate branches. The actor also proposes branch updates. All reported configurations share the evaluation implementation. Appendix A gives the system configuration.

Metrics. For task group i with T_i stages, let $c_{i,t} \in \{0, 1\}$ indicate whether stage t is answered correctly. We report final success rate (SR) and task progress score (PS):

$$\mathrm{SR} = \frac{100}{N} \sum_{i=1}^{N} c_{i,T_i}, \qquad \mathrm{PS} = \frac{100}{N} \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} c_{i,t}. \tag{6}$$

PS includes the final stage and weights task groups equally. Qwen3.8 Flash evaluates answer equivalence. Missing answers contribute zero. Token consumption sums input and generation across search and memory-maintenance model calls in a task group, including writing and later reads. Judge usage is recorded separately.

Baselines. We compare with long-context prompting, ReasoningBank (Ouyang et al., 2025), MemoryOS (Kang et al., 2025), Mem0 (Chhikara et al., 2025), HippoRAG 2 (Jiménez Gutiérrez et al., 2025), MemoRAG (Qian et al., 2024), BeliefMem (Liao et al., 2026), and MAGE (Chen et al., 2026). The actor, task collection, retrieval environment, search budgets, and evaluation version are shared across methods.

Table 2: Memory component ablations on MemoryArena Search under the same agent and evaluation configuration. Tokens sum input and generation across the task trajectory.

Configuration	SR (%)	PS (%)	Tokens/group (k)
History only	62.90	45.82	1175.00
Single candidate	64.71	49.56	896.00
Without revision fields	66.97	51.13	826.48
BranchMem	**71.49**	**59.23**	**812.59**

4.2 Overall search performance

BranchMem correctly answers 158 of 221 final questions, achieving 71.49% SR with a Wilson 95% interval of 65.21–77.04%. It exceeds MAGE by 7.24 percentage points, long-context prompting by 8.14 points, and BeliefMem by 11.31 points. Across the baseline set, SR differences range from 7.24 to 19.00 points. BranchMem also achieves 59.23% PS, exceeding MAGE by 10.50 points and long-context prompting by 9.56 points. Its mean consumption of 812.59k tokens is the lowest among the evaluated methods.

4.3 Memory component ablations

We compare the complete BranchMem configuration with three memory variants under the same search-agent implementation, task collection, and evaluation version. Each variant is applied throughout the multistage task. *History only* retains the ordinary interaction history and omits the candidate-branch memory. *Single candidate* retains one candidate interpretation for subsequent decisions. *Without revision fields* retains candidate content while omitting the explicit revision fields.

Candidate memory. Omitting candidate branches reduces SR by 8.59 points and PS by 13.41 points. History-only memory consumes 1,175k tokens per group, compared with 812.59k for BranchMem. This comparison measures the contribution of retaining task-related interpretations alongside the executed interaction history.

Multiple candidates. Restricting memory to one candidate reduces SR by 6.78 points and PS by 9.67 points. The result supports keeping alternatives available across stages as evidence accumulates and the agent reevaluates its current interpretation.

Revision information. Omitting revision fields reduces SR by 4.52 points and PS by 8.10 points. The full configuration also uses 13.89k fewer tokens per group. Revision information contributes to both final completion and intermediate progress when candidate content remains available.

4.4 Performance by task length

The full run achieves 64.56% SR on groups with at most six stages, 71.11% on seven-to-eight-stage groups, and 82.69% on groups with at least nine stages (Figure 3a). Stage count also changes task composition and the number and specificity of clues.

4.5 Token use

Across all 221 scheduled groups, actor usage totals 177,100,719 input and 2,481,411 output tokens, including the interrupted group's recorded calls. Per-group means are 801,361 input and 11,228 output tokens, totaling 812,589. Table 1 reports baseline means of 1,207k–5,937k tokens. BranchMem uses 50.48% fewer tokens than MAGE and 86.31% fewer than long-context prompting. In the component comparison, BranchMem uses 30.84% fewer tokens than history-only memory and 9.31% fewer than the single-candidate configuration.

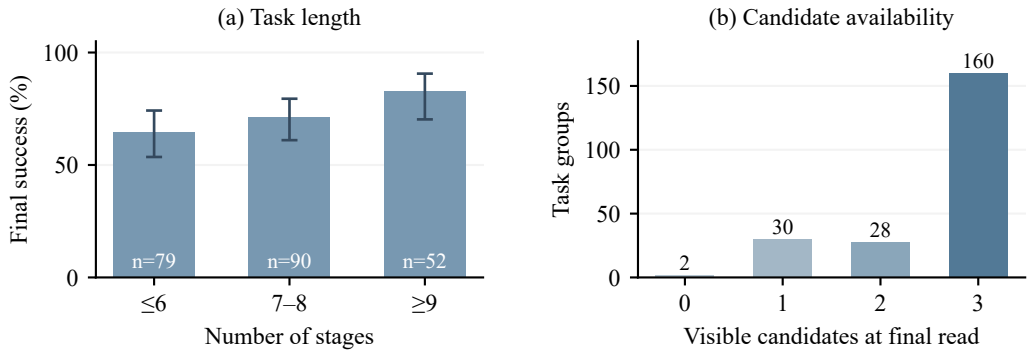


Figure 3: Recorded full-run statistics. (a) Final SR by stage count, with Wilson 95% intervals. The strata contain 79, 90, and 52 groups. (b) Visible candidate counts at the final read in 220 completed groups.

5 Analysis and Case Studies

5.1 Candidate availability and revision activity

We inspect the first recorded request at each of 1,633 stages in the 220 groups with final answers. At the final read, 188 groups (85.45%) expose multiple candidates: 160 expose three and 28 expose two. Thirty expose one and two expose none (Figure 3b). Final readouts contain an average of 9.47 local judgments. Across stage reads, 8,813 of 9,539 judgment occurrences (92.39%) include source identifiers.

We observe 39 lifecycle-state changes across 29 groups by comparing successive visible records for the same candidate text. In 111 groups, a candidate marked challenged, superseded, or abandoned remains visible at some later read. These records provide the actor with both candidate content and its current task status. Appendix C defines the counting procedure.

5.2 Observed search cases

The following cases connect a retained judgment to a subsequent search action, local revision, or answer decision. The food-publication case asks for a paper title; the introductory controlled illustration asks for an ingredient.

A retained author guides retrieval (q1206). The first answer favored *Use of spent coffee grounds as food ingredient in bakery products*. A candidate record also preserved a quality-indicator paper (document 75010) and the author Cherevychna. At the start of stage two, the four selected raw excerpts came from one document, and candidate readout supplied the additional author clue. The seventh search used that clue and subsequent retrieval reached document 7581. The final answer was *Development of waffle with fat filling using unconventional vegetable raw materials*.

A film relationship is revised (q1141). The person remained relevant while director and plot evidence changed the associated film to *An American in Paris*. Earlier and later records remained visible, allowing the actor to interpret the changed relationship while retaining useful information about the person.

A candidate retains an author list (q209). The task asks how many authors wrote a study discussed in a Psychology Today article. A candidate record links the study title to a nine-author PubMed list (document 68214) and its discussion in the article (87700). The retained record places the author list in the context of the study discussed by the article, preserving the relation between the target study and the count used in the final answer.

5.3 Scope and limitations

Candidate construction and source interpretation depend on the actor. Source identifiers establish record availability, and the model evaluates semantic support during later reads. Candidate-text matching leaves aliases and same-name entities to the model. The capacity bound retains the most recently inserted candidates, so older entries can leave the persistent state. The experiments cover one search collection and actor configuration. Additional repetitions and further environments would characterize the variability and transfer of the measured effects.

6 Conclusion

BranchMem maintains task-relevant candidate interpretations as persistent branches over a shared search history. Local judgments, source references, and revisions remain associated with their candidates and are read together at later decisions. On all 221 MemoryArena Search groups, BranchMem answers 158 final questions correctly (71.49%), exceeding the strongest evaluated baseline by 7.24 percentage points. Component ablations show reductions in success and progress when candidate memory, multiple candidates, or revision fields are removed. Recorded cases trace candidate information through retrieval, local revision, and answer selection.

Reproducibility Statement

The task denominator, scoring rules, and token accounting are specified in Section 4. Appendix A describes the retrieval configuration, memory interface, and capacity limits. Section 4.3 describes the memory component variants. Appendix B provides selected trace observations, and Appendix C defines the mechanical trace statistics.

AI Use Statement

Generative AI tools assisted with literature discovery and summarization, experiment implementation and result analysis, qualitative inspection of recorded traces, manuscript drafting and translation, reference formatting, and the creation of the method illustration. The authors are responsible for the cited sources, numerical results, technical claims, and manuscript.

References

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, and Ankit Chadha. APEX-MEM: Agentic semi-structured memory with temporal reasoning for long-term conversational AI. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16470–16489, 2026. doi: 10.18653/v1/2026.acl-long.749. URL https://aclanthology.org/2026.acl-long.749/.

Arnab Chattopadhayay and Debdipta Halder. Belief-state engine: Augmenting LLMs for principled planning under partial observability, 2026. URL https://arxiv.org/abs/2609.10036.

Yaoqi Chen, Haibin Lai, Yuru Feng, Chuyu Han, Qianxi Zhang, et al. Beyond semantic organization: Memory as execution state management for long-horizon agents, 2026. URL https://arxiv.org/abs/2606.06090.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, 1986. doi: 10.1016/0004-3702(86)90080-9. URL https://www.sciencedirect.com/science/article/pii/0004370286900809.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization, 2024. URL https://arxiv.org/abs/2404.16130v1.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. MemoryArena: Benchmarking agent memory in interdependent multi-session agentic tasks, 2026. URL https://arxiv.org/abs/2602.16313.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions, 2025. URL https://arxiv.org/abs/2507.05257.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.14831.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models, 2025. URL https://arxiv.org/abs/2502.14802.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent, 2025. URL https://arxiv.org/abs/2506.06326.

Junfeng Liao, Qizhou Wang, Jianing Zhu, Bo Du, Rui Yan, and Xiuying Chen. Belief memory: Agent memory under partial observability, 2026. URL https://arxiv.org/abs/2605.05583.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient lifelong memory for LLM agents, 2026. URL https://arxiv.org/abs/2601.02553.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents, 2024. URL https://arxiv.org/abs/2402.17753.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, et al. ReasoningBank: Scaling agent self-evolving with reasoning memory, 2025. URL https://arxiv.org/abs/2509.25140.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

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, 2023. URL https://arxiv.org/abs/2304.03442.

Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. MemoRAG: Moving towards next-gen RAG via memory-inspired knowledge discovery, 2024. URL https://arxiv.org/abs/2409.05591.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://arxiv.org/abs/2212.10509.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. URL https://arxiv.org/abs/2305.16291.

Yu Wang and Xi Chen. MIRIX: Multi-agent memory system for LLM-based agents, 2025. URL https://arxiv.org/abs/2507.07957.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=wIonk5yTDq.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, et al. ReSum: Unlocking long-horizon search intelligence via context summarization, 2025b. URL https://arxiv.org/abs/2509.13313.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents, 2025. URL https://arxiv.org/abs/2502.12110.

Aijun Yang, Qianxue Guo, Ziyi Huang, Yuxuan Chen, Shiyou Qian, and Jian Cao. Self-correcting long-horizon search agents via tree-structured memory, 2026. URL https://arxiv.org/abs/2608.10676.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.03629.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory, 2023. URL https://arxiv.org/abs/2305.10250.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models, 2024. URL https://arxiv.org/abs/2310.04406.

Mingxi Zou, Zhihan Guo, Langzhang Liang, Zhuo Wang, Qifan Wang, et al. Remember the decision, not the description: A rate-distortion framework for agent memory, 2026. URL https://arxiv.org/abs/2605.10870.

A Implementation and Evaluation Details

A.1 Stage-level memory interface

The evaluated system uses a search-evidence memory backend with a rendered history and candidate view. Each stage receives the overall question, its current stage question, the tool instructions, and the selected persistent state. The same actor produces tool actions, the stage answer, and an optional structured `memory_update`. There is no additional generation model for branch writing. The final stage returns an answer and a run record; the preceding stages can update the persistent state.

The history renderer retains up to twelve recent stage records. Each record contains the previous stage question and answer, search strings, and selected source snippets. Referenced or actively inspected evidence is prioritized; up to four excerpts per prior stage are retained, each capped at 800 characters. The recorder retains the three most recently inserted branches and the renderer presents them in insertion order. Updates to an existing branch preserve its insertion position. A branch stores candidate text, role, relation to the overall answer, claims with statuses and source IDs, optional excerpts, a lifecycle state, revision reason, and a compact update history. Supported roles include

target, alternative, bridge, counterexample, and unknown. The lifecycle vocabulary includes active, challenged, superseded, and abandoned. History and candidate counts and individual field lengths are limited. Each update accepts at most 24 judgments; statement and claim-excerpt fields are capped at 500 characters, candidate text at 180, and task relations at 240. Judgment lists accumulate within retained branches. The component conditions use these record and character limits, with actual token consumption measured for each condition.

Table 3: Frozen actor and retrieval configuration for the reported run.

Component	Setting
Actor	DeepSeek V4.1 Flash; thinking disabled
Final-answer judge	Qwen3.8 Flash, with documented equivalence review
Retrieval	Local SQLite full-text search, model-driven
Search results	Top 10; 256 retrieval-interface tokens per snippet
Per-stage operations	10 search; 20 document; 30 interaction iterations
Prompt history	At most 12 stage records
Prompt candidate view	At most 3 branches
First request output cap	1,400 tokens

Temperature and seed were left unspecified in the recorded provider requests. The 1,400-token first-request output cap and a separate 4,096-token stage budget occupy different layers of the runtime configuration. Candidate records are matched by normalized candidate text. Alias resolution and same-name candidates therefore require interpretation by the model.

A.2 Recorder operation

The recorder first preserves the ordinary stage history and observed sources. For each proposed candidate update, it matches or creates a branch and appends new judgments and revision records with duplicate handling. A supported or contradicted judgment must cite an available source identifier; otherwise its status becomes unverified. Source-ID availability is a syntactic check. A cited excerpt and its associated claim remain visible for the actor to examine in a later stage. The actor performs later relation checks and final-answer selection from the rendered state.

The following stage-level pseudocode summarizes the interface:

```
initialize history, sources, branches as empty
for each stage t:
  render selected history, sources, and branches
  run actor on overall question and stage question
  collect answer, trace, and optional memory update
  if final stage:  save run and return answer
  record stage question, answer, queries, and sources
  validate candidate statuses and source identifiers
  match or create branches; append judgments and revisions
  apply history and candidate capacity limits
```

B Trace Inspection

Three illustrative traces from the recorded full run supplement the candidate-memory cases discussed in the main text. Table 4 records their relevant progression. These observations can be inspected against the saved requests and tool logs.

In q1206, the four selected raw excerpts at the start of the later stage all referred to the same document. The candidate view supplied an additional paper-and-author clue. This trace shows an information path through the candidate record. In q1141, revision concerned the film associated with an already identified person; older film records remained present in the state, and the actor had to interpret the later evidence. The q579 case illustrates direct reuse of prior sources at answer time.

Table 4: Selected stage-level observations. Each row describes an observed sequence within one successful task group.

Group	Recorded progression
q1206	An early alternative paper and author remained in a candidate record; the later stage queried that author and found the filled-waffle study.
q1141	The person was identified early; a later film judgment changed after director and plot evidence arrived.
q579	The final stage used earlier source identifiers and date evidence, answered correctly, and made no additional tool calls.

Table 5: Recorded cases linking memory content to later decisions.

Group	Earlier state	Later action or evidence	Outcome
q1206	Coffee-ground paper favored; another paper and author retained.	A query used Cherevychna and reached the filled-waffle study.	Correct publication title.
q1141	Georges Guétary linked to an incorrect 1955 film.	Director and plot evidence supported the 1951 film *An American in Paris*.	Correct answer after local revision.
q209	Branch record preserved the study title and nine-author list.	The retained PubMed list supplied the study-specific author count.	Correct count: nine.

C Mechanical Trace Statistics

The analysis inspects one initial request per recorded stage. When memory appears in both system and user messages, the user-side copy supplies the counting unit. Candidate occurrences and judgment occurrences count each rendered record once per stage. Source-bearing judgments have a nonempty source-ID attribute. This count measures provenance availability. Lifecycle changes compare the visible state of identical candidate text between successive appearances. A candidate can be absent between these appearances. Nonactive visibility counts whether any recorded read includes a challenged, superseded, or abandoned candidate. Semantic entailment and human event-level agreement are outside these mechanical statistics.