Know Where You Stand: Memory-Source Choice in Long-Context Dialogue Agents

Caishen Zhou, Yihong Tang, Xuefeng Bai, Kehai Chen*, Min Zhang
Harbin Institute of Technology, Shenzhen
cszhou@stu.hit.edu.cn, {baixuefeng,chenkehai}@hit.edu.cn

Abstract

Memory enables long-context dialogue agents to maintain user state across many turns, but it also introduces a fundamental source-choice problem: grounding the response in the immediate context or retrieved prior memory? A wrong choice incurs two complementary costs: neglecting valid aged-out evidence, or overwriting sufficient context with stale memory. Existing systems and benchmarks largely conflate retrieval with source choice, relying on the model to resolve it implicitly and leaving this critical decision largely unexplored. In this paper, we introduce MSBench, a controlled benchmark that isolates source choice by pairing questions answerable exclusively from context with those requiring prior memory. Under this protocol, strategies that depend on the model's implicit judgment exhibit sharply degraded performance, exposing the limits of leaving source choice ungoverned. To this end, we propose the Memory-Source Reasoner (MSR), which frames source choice as an explicit metacognitive decision: it reasons about contextual sufficiency, answers directly when appropriate, and selectively retrieves missing evidence otherwise. Experimental results show that MSR achieves the best overall accuracy and selection accuracy across four answer backbones. On the primary GPT-5-mini run, it improves over the best non-MSR baselines by 12.6 points in overall accuracy and 6.3 points in selection accuracy. Our benchmark, code, prompts, and evaluation scripts will be released at `https://github.com/Jockeylove/MSBench`.

1 Introduction

Long-context dialogue agents must maintain user preferences, task history, and personal state across dozens or hundreds of turns [14, 25, 35, 36], while preserving role consistency [32] and adapting to evolving user needs [5, 37]. This continuity is typically realized through memory modules that store and incrementally update past turns as retrievable representations [4, 28, 34, 41]. Despite these advances, memory introduces a source-choice problem during response generation: the immediate dialogue context and retrievable prior memory may both seem pertinent to the current query, yet only one may provide the correct grounding. Choosing the incorrect source gives rise to two complementary failures (Figure 1): missed prior-memory use, where aged-out but correct evidence is ignored, and unnecessary prior-memory use, where stale or off-branch memory overwrites sufficient visible context.

Existing memory-augmented dialogue systems and long-context benchmarks largely conflate retrieval with the source-choice decision. On the one hand, memory frameworks such as MemGPT [28], mem0 [4], LightMem [7], Nemori [24], and MemOS [22] develop increasingly capable pipelines for storing, consolidating, and retrieving long-term memory, while adaptive RAG methods such as FLARE [16], Self-RAG [1], and Adaptive-RAG [13] learn whether to consult or critique external evidence for generation. These approaches typically concatenate retrieved memory with the

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

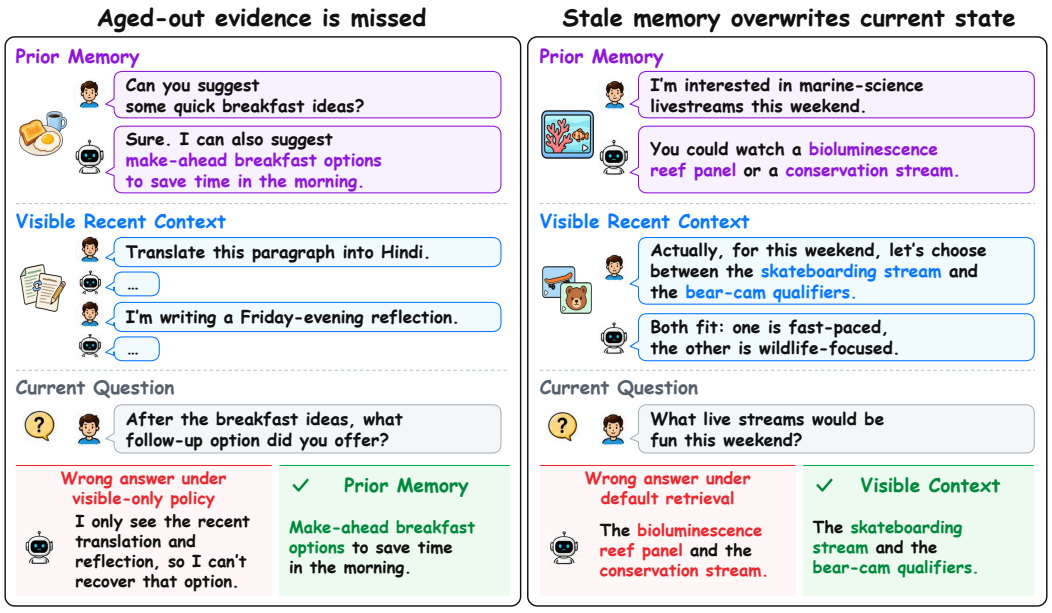

Figure 1: Missed and unnecessary prior-memory use under source ambiguity.

immediate dialogue context and delegate the final source boundary to the generator, offering no explicit mechanism for resolving tension between the two sources. On the other hand, long-dialogue benchmarks such as LoCoMo [25], PersonaMem [14], LongMemEval [35], and AMemGym [17] primarily measure whether past information can be recovered and used; they do not systematically test when retrieval should be suppressed because the immediate dialogue context already provides the correct grounding. This conflation limits our understanding of source-choice behavior, and in turn constrains mitigation of missed prior-memory use and unnecessary prior-memory use.

To study this behavior directly, we introduce **MSBench**, a controlled replay benchmark that fixes the immediate dialogue context and pairs it with two question types: **Recent-context** questions answered by the visible context, and **Prior-memory** questions requiring aged-out evidence. This design turns missed and unnecessary prior-memory use into measurable outcomes. Under this protocol, fixed policies fail in opposite regimes: No Memory reaches 50.7% on Recent-context questions but only 5.3% on Prior-memory questions, whereas always-retrieve RAG improves Prior-memory accuracy to 53.6% but lowers Recent-context accuracy to 41.3%. Prompted LLM Retrieval Gate improves source decisions over fixed policies but remains limited (46.8% overall accuracy and 75.4% Selection Acc.), suggesting that generic retrieval gating does not fully resolve source choice.

Motivated by this, we propose **MSR** (**Memory-Source Reasoner**), which treats source choice as an explicit metacognitive decision inspired by monitoring-and-control theories of metamemory [27]. MSR first checks which answer requirements are supported by visible context and a hidden memory preview. It retrieves prior memory when the preview supports requirements missing from the visible context; otherwise, it answers from visible evidence. Across four answer backbones, MSR achieves the best overall accuracy and source-selection accuracy, improving over the strongest non-MSR baseline by 12.6 points in overall accuracy and 6.3 points in source-selection accuracy on GPT-5-mini.

Our main contributions are as follows:

1. **Problem and benchmark.** We formulate memory-source choice as an answer-time behavior in long-context dialogue agents and introduce **MSBench**, a paired replay benchmark that exposes when agents should trust visible context versus retrievable prior memory. MSBench reveals two systematic failure modes: missed prior-memory use and unnecessary prior-memory use.

2. **Method.** We present **MSR**, an evidence-guided control framework that infers an atom-level source-monitoring state and reuses it for source selection and source-specific grounding: an-

choring visible answers when visible support is complete and expanding preview-supported exchanges when prior memory closes support gaps.

3. **Empirical validation.** Across answer backbones, ablations, error diagnostics, and memory-system references, MSR improves both source choice and answer accuracy, showing that reusable source monitoring helps control how the selected source is used.

2 Related Work

Retrieval control. Retrieval-augmented generation grounds responses in external evidence [20]. FLARE anticipates upcoming content and retrieves when a predicted sentence contains low-confidence tokens [16]. Self-RAG learns reflection tokens to control retrieval and assess passage relevance, evidential support, and response quality [1]. Beyond retrieval, GA-Rollback checks agent actions and triggers rollback when errors are detected [21]. Adaptive-RAG instead trains a question-complexity classifier to choose among no retrieval, single-step retrieval, and iterative retrieval [13]. SearchSkill makes query planning explicit through reusable search skills [10]. UI-Copilot trains GUI agents to request memory retrieval on demand [23]. In dialogue, the same question can require prior-memory retrieval at one checkpoint but have sufficient visible evidence at another.

Dialogue memory systems. Long-term conversational memory preserves information across sessions through stored history, summaries, and reusable representations [34, 36, 41]. SCM uses a memory controller to govern memory updates and access [33]. MemGPT manages movement between memory tiers to extend a limited context window [28], while mem0 extracts, consolidates, and retrieves salient conversational information [4]. LightMem separates lightweight filtering, topic-based consolidation, and offline memory updates [7]. Nemori integrates interactions into episodic narratives and distills semantic knowledge [24]; MemOS provides unified abstractions for representing and managing different memory types [22]. Memory granularity also affects the usefulness of retrieved evidence: SeCom builds topically coherent conversation segments and applies compression-based denoising [30]. Coherent exchanges preserve context that isolated turns lose, but coarse units may introduce unrelated material. Memory organization and answer-time source choice thus address complementary aspects of evidence use.

Memory-use benchmarks. Long-input and multi-source benchmarks evaluate evidence use over extended contexts and heterogeneous records [2, 3, 6, 26]. Conversational benchmarks assess multi-session recall, personalization, and user-state updates [9, 11, 14, 15, 18, 19, 25]; agent benchmarks evaluate memory in incremental interactions and downstream tasks [8, 12, 31, 38]. LifeDialBench evaluates continuous conversational lifelogs with temporally ordered online queries [39]. LongMemEval separates information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention [35]. AMemGym evaluates interactive, on-policy conversations with structured user-state evolution [17]. ConvoMem compares full-context and retrieval-based systems, finding full-context approaches competitive on shorter histories [29]. That comparison varies how history is supplied; MSBench fixes the visible/prior boundary and pairs questions requiring different sources at the same replay checkpoint to test both beneficial and unnecessary memory use.

3 MSBench: Measuring Memory-Source Choice

MSBench makes answer-time memory-source choice observable under replay: at a fixed checkpoint, does the agent ground its answer in visible recent context or in retrievable prior memory?

3.1 Source-Bounded Replay Setup

At each checkpoint, the agent receives a question, visible recent context, and an incrementally updated memory index. Recent-context questions have authoritative evidence in the visible window, making stale or off-branch prior memory a potential distractor. Prior-memory questions require evidence outside that window, so answering from visible context alone misses the decisive support. Table 1 compares benchmarks along four dimensions. *Explicit source boundary* separates visible context from retrievable prior memory. *Incremental replay* exposes dialogue checkpoint by checkpoint, with future turns hidden. *State updates* change the available memory as replay advances and older turns leave the visible window. *Same-checkpoint paired contrast* asks two questions at the same replay state, one requiring visible context and the other prior memory.

Table 1: Comparison of dialogue-memory benchmarks. The four dimensions are visible/prior source separation, chronological replay with future turns hidden, memory updates as turns leave the visible window, and paired questions requiring different sources at the same checkpoint.

Benchmark	Explicit source boundary	Incremental replay	State updates	Same-checkpoint paired contrast
LongBench / NoLiMa [2, 26]	×	×	×	×
LoCoMo / PersonaMem [14, 25]	△	×	△	×
LongMemEval [35]	△	×	✓	×
AMemGym [17]	△	△	✓	×
ConvoMem [29]	△	×	△	×
MSBench	✓	✓	✓	✓

✓: direct focus; △: partial or indirect; ×: not central.

3.2 Construction and Quality Control

We construct checkpoints from PersonaMem v2 dialogue scenes [15]. At checkpoint t_i, the visible source contains the latest $w = 16$ turns; earlier turns enter prior memory as replay advances, and future turns remain hidden. Each checkpoint pairs one **Recent-context** question with one **Prior-memory** question under the same local frame, varying which source supplies the decisive evidence. The release comprises 900 items from 450 checkpoints across 45 scenes and four source-sensitive question families: single-detail, multi-part, referenced-item, and updated-state questions.

Questions and gold answers are drafted with **gpt-5.4** and manually reviewed. Review verifies that Prior-memory items require aligned non-visible evidence, while Recent-context items are grounded in the latest visible state. Retrieval accesses only pre-window turns in the incrementally updated store, rather than a full-scene prebuild. Appendix B.2 gives the manual review checklist.

Prior-memory questions include recent context about the same topic but a different instance or version; Appendix B.3 reports the audit.

3.3 Evaluation Interface and Metrics

All policies receive the same question, visible context, and access to the same checkpoint-specific prior-memory store, with future turns hidden. Each policy returns an answer and, when applicable, a source decision indicating the source branch selected for answering the current question.

We report answer accuracy on **Recent-context Qs**, **Prior-memory Qs**, and the mixed workload (**Overall**). An answer is correct when it covers the reference's required facts and constraints, without requiring identical wording. **Selection Acc.** evaluates the chosen source: visible context for Recent-context questions and prior memory for Prior-memory questions. Evaluation follows `run` → `judge` → `score`: unless otherwise noted, gpt-4o-mini judges each answer three times and majority vote determines correctness. Section 5.2 reports shortcut, self-consistency, and human-audit checks. The benchmark, code, prompts, and evaluation scripts will be released here.

3.4 Empirical Diagnosis of Fixed Source Policies

We analyze two representative fixed source policies on MSBench: staying with the visible context (No Memory) and always consulting prior memory (RAG). Figure 2 shows a clear crossover pattern on the primary GPT-5-mini run. No Memory remains competitive when the visible context already contains the authoritative evidence (0.507 Recent-context accuracy), but nearly fails when the answer depends on aged-out evidence (0.053 Prior-memory accuracy). Always-retrieve RAG exhibits the opposite behavior: it improves Prior-memory accuracy to 0.536, but lowers Recent-context accuracy to 0.413 by admitting unnecessary prior memory.

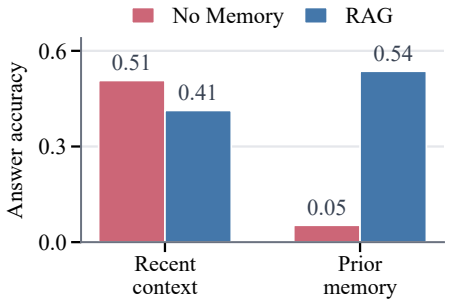

Figure 2: Answer accuracy of fixed source policies on GPT-5-mini.

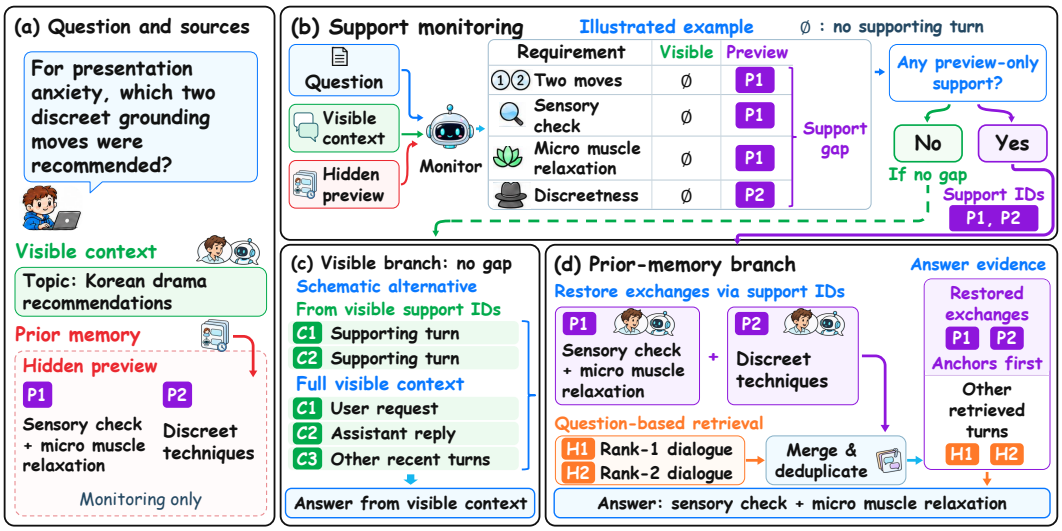

Figure 3: Overview of MSR. A reusable source-monitoring state guides source selection and the construction of branch-specific grounding contexts.

This crossover motivates monitoring which source supplies the required support before answering, rather than treating retrieval quality alone as the bottleneck (Section 4).

4 MSR: Structured Source Monitoring

The preceding analysis motivates tracking which answer requirements are supported by each source before generating an answer. MSR (**Memory-Source Reasoner**) decomposes the question into these requirements, identifies their supporting dialogue turns in the visible context and a hidden memory preview, and selects prior memory when the preview supplies support missing from the visible context. The same support state organizes evidence for the selected answer branch.

Figure 3 summarizes the pipeline. We first define the two sources and the preview, then describe support monitoring and source choice, and finally explain how the selected evidence is prepared for answering. Appendix C provides the output schema, complete monitor prompt, inference algorithm, and three worked examples linking support identifiers to source choice and grounding.

4.1 Source-Bounded Setup and Hidden Preview

At checkpoint t, MSR receives the question q_t, the visible recent dialogue C_t, and access to prior memory $M_{<t}$. Prior memory contains earlier turns outside the visible window in the same incrementally updated store. The two sources are disjoint, and future turns are unavailable.

Before source selection, MSR retrieves a small hidden preview P_t from prior memory. Preview and answer-time retrieval share the same store and scorer, but the preview uses a smaller retrieval budget k_p and excludes visible turns. The preview lets the monitor assess whether earlier dialogue contains support missing from C_t. It is used for source monitoring rather than passed directly to answer generation; supporting turns are subsequently recovered through their identifiers when prior memory is selected to construct the prior-memory answer context.

4.2 From Support State to Route

In one LLM call, the source monitor reads (q_t, C_t, P_t) and decomposes the question into atoms $\mathcal{A}(q_t)$: the facts and constraints a correct answer must satisfy. Each atom preserves relevant instance, temporal, quantity, and ordering requirements. Support is assessed against these requirements rather than topical overlap alone, distinguishing evidence for the requested fact from related discussion.

Each visible turn and previewed memory item has a fixed identifier. For each atom a, the monitor selects at most one visible support identifier $e_t^C(a)$ and at most one preview support identifier $e_t^P(a)$, recording $\emptyset$ when no support is identified in that source. These identifiers point to existing dialogue turns, not generated evidence text. The resulting source-monitoring state is

$$Z_t = \{(a, e_t^C(a), e_t^P(a)) : a \in \mathcal{A}(q_t)\}.$$

The monitor judges semantic support. Deterministic checks enforce unique atom identifiers, membership of evidence identifiers in the corresponding allowed source, and at most one support identifier per source for each atom. Appendix C.4 reports the human audit and support-to-route cases.

MSR selects prior memory when the preview supports an atom that the visible context does not support. These actionable support gaps form the set

$$F_t = \{a \in \mathcal{A}(q_t) : e_t^C(a) = \emptyset,\ e_t^P(a) \neq \emptyset\}.$$

We summarize their proportion with the normalized preview gain,

$$G_t(q_t) = \frac{|F_t|}{|\mathcal{A}(q_t)|}.$$

MSR selects the prior-memory branch iff $F_t \neq \emptyset$ (equivalently, $G_t(q_t) > 0$). An atom supported by both sources does not create a gap; an atom supported by neither remains unresolved and does not itself trigger retrieval under the binary rule. Other atoms with actionable gaps still determine whether the question takes the prior-memory branch. The full state Z_t is retained to guide evidence construction after selection through visible support anchors or restored prior-memory exchanges.

4.3 Selection-Conditioned Grounding

If $F_t = \emptyset$, MSR takes the visible-context branch. It builds a visible support pack S_t^C from the dialogue turns referenced by the visible support identifiers in Z_t. The answer model is instructed to ground factual content in this pack, while using the full visible context to interpret local references and maintain conversational coherence. Requirements unsupported by either source remain unresolved.

If $F_t \neq \emptyset$, MSR retrieves prior memory for answering. Let k_r denote the number of prior-memory items retained after ranking by relevance to the question; the main experiments use $k_r = 20$. The question-ranked evidence list is

$$Q_t = \mathrm{TopK}_{k_r}(q_t, M_{<t}).$$

For each preview support identifier associated with an atom in F_t, MSR follows the pointer to its prior dialogue turn and restores the short user–assistant exchange containing that turn. This preserves the request and response context around the missing evidence. Denoting this recovery operation by $\mathrm{ExpandExch}$ gives

$$X_t = \mathrm{ExpandExch}(\{e_t^P(a) : a \in F_t\}, M_{<t}).$$

The final retrieval context is a priority merge of the two ordered evidence lists,

$$R_t = \mathrm{Merge}^{X \prec Q}(X_t, Q_t).$$

Here $\mathrm{Merge}^{X \prec Q}$ deduplicates the two lists, places support-anchored exchanges before question-only hits, preserves local exchange order, and then preserves question-retrieval rank. Both lists contain only prior-memory evidence: the visible context C_t is not appended to R_t.

The same backbone generates the answer in a second LLM call, using the prompt and inputs of the selected branch:

$$\hat{y}_t = \begin{cases} \mathrm{Ans}_C(q_t, S_t^C, C_t), & F_t = \emptyset, \\ \mathrm{Ans}_M(q_t, R_t), & F_t \neq \emptyset. \end{cases}$$

Here Ans_C and Ans_M denote the two branch-specific generation configurations, not different backbones. MSR thus reuses its support state both to choose the answer source and to organize evidence within that source. Appendix C.5 examines extensions to partial retrieval and abstention.

Table 2: Answer and source-selection accuracy on MSBench. Prompted Gate denotes Prompted LLM Retrieval Gate. Bold values mark the best score for each metric within a backbone.

Backbone	Method	Answer accuracy ↑			Selection Acc. ↑
		Recent-context	Prior-memory	Overall	
GPT-5-mini	No Memory	0.507	0.053	0.280	0.500
	RAG	0.413	0.536	0.474	0.500
	Prompted Gate	0.522	0.413	0.468	0.754
	MSR (ours)	**0.617**	**0.583**	**0.600**	**0.817**
Gemini 3.1 Flash-Lite	No Memory	0.383	0.017	0.200	0.500
	RAG	0.350	0.400	0.375	0.500
	Prompted Gate	**0.467**	0.233	0.350	0.667
	MSR (ours)	0.433	**0.450**	**0.442**	**0.792**
Qwen3 235B-A22B	No Memory	0.400	0.033	0.217	0.500
	RAG	0.317	**0.450**	0.383	0.500
	Prompted Gate	0.400	0.083	0.242	0.550
	MSR (ours)	**0.417**	0.417	**0.417**	**0.725**
DeepSeek V4 Pro	No Memory	0.533	0.017	0.275	0.500
	RAG	0.367	0.500	0.433	0.500
	Prompted Gate	0.500	0.483	0.492	0.817
	MSR (ours)	**0.600**	**0.517**	**0.558**	**0.825**

5 Results and Analysis

All methods answer the same 900 paired items from 450 replay checkpoints under the protocol in Section 3.3. Before each model call, we construct the visible context from the latest 16 dialogue turns and restrict retrievable prior memory to earlier, non-visible turns. Future turns are unavailable. Retrieval uses `text-embedding-3-small`, cosine similarity, and the top 20 items from the incrementally updated prior-memory index. The source boundary is set by this shared input construction rather than by a provider's maximum context length. Appendix D.2 reports the input-token distributions across backbones and methods under the shared visible/prior source boundary.

We compare **No Memory**, which answers from visible context without retrieval; **RAG**, which always retrieves prior memory and supplies it alongside visible context; **Prompted LLM Retrieval Gate**, abbreviated as **Prompted Gate** in tables and figures; and **MSR**. The prompted gate uses the answer backbone itself to assess whether the question can be answered from the visible dialogue. It either answers directly or invokes the memory-retrieval tool once, then answers using the returned evidence and visible dialogue. It requires no separately trained router. Appendix D gives its full prompt and tool interface, including the permitted query and fixed retrieval width.

Table 2 compares the four policies under each answer backbone using the same questions and judging protocol. Diagnostics and ablations use GPT-5-mini unless otherwise noted. We examine both source selection and answer quality to distinguish routing gains from improvements in evidence use within the selected source, separating source-choice errors from failures during answer construction.

5.1 Main Results

Table 2 confirms the crossover across backbones: No Memory favors Recent-context questions, whereas RAG improves Prior-memory accuracy but hurts Recent-context accuracy.

MSR gives the best Recent-context, Prior-memory, Overall, and Selection Acc. on GPT-5-mini, improving both source regimes rather than trading Recent-context accuracy for Prior-memory recall. Across Gemini, Qwen, and DeepSeek, MSR again achieves the strongest Overall and Selection Acc., although individual source regimes favor different methods. The gains therefore reflect better performance on the mixed workload across backbones. Appendix E reports confidence intervals and paired tests. Appendix D.4 compares adapted FLARE and Self-RAG controllers in the SciPhi-Mistral model family with matched source boundaries and retrieval settings.

Table 3: Benchmark validity and judge reliability. (a) Consistency across repeated LLM judgments. (b) Source prediction from questions alone. (c) Agreement between majority LLM judgments and blinded human annotations.

(a) LLM-judge consistency

Rows	N	Unanimous ↑	2–1 split ↓	Pairwise agree. ↑	Fleiss κ ↑
No Memory	900	0.989	0.011	0.993	0.982
RAG	900	0.982	0.018	0.988	0.976
Prompted Gate	900	0.972	0.028	0.981	0.963
MSR	900	0.979	0.021	0.986	0.972
All controlled rows	3600	0.981	0.019	0.987	0.974

(b) Question-only shortcut probe

Input	Sel. Acc. ↑
Question only	0.538
Recent-context	0.420
Prior-memory	0.658

(c) Human–LLM agreement

Comparison	N	Agreement ↑	Cohen's κ ↑
Annotator A vs. LLM majority	160	0.906	0.813
Annotator B vs. LLM majority	160	0.906	0.813
Annotator A vs. Annotator B	160	0.975	0.949

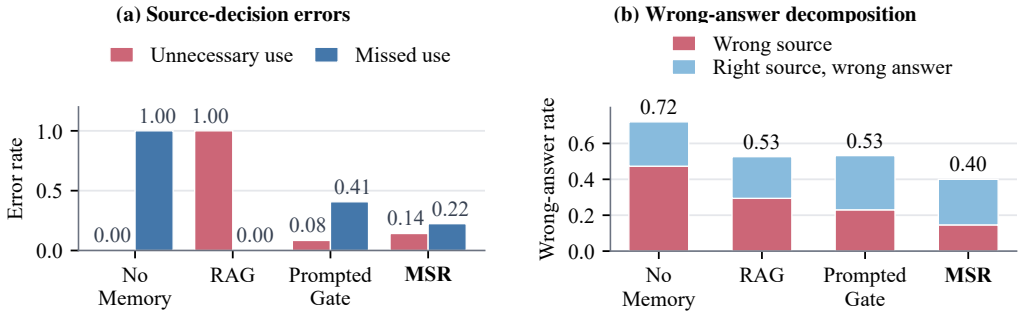

Figure 4: Stage-level diagnostics on GPT-5-mini. (a) Rates of unnecessary and missed prior-memory use. (b) Wrong-answer rates by source-selection correctness.

5.2 Benchmark Validity and Judge Reliability

Because MSBench is new and uses automatic judging, we check source-label leakage, judge stability, and agreement with blinded manual annotation before analyzing failures.

In Table 3, question-only selection reaches 0.538 Selection Acc., indicating limited source-label predictability from wording alone. The three-run judge is nearly always unanimous, and blinded human annotation agrees strongly with majority labels, supporting the aggregate comparisons and diagnostic breakdowns based on these answer-correctness labels.

5.3 Stage-Level Diagnostics

We construct this diagnostic from the gold source label and each method's selected branch: a Recent-context item routed to prior memory counts as unnecessary use, while a Prior-memory item kept in visible context counts as missed use. Figure 4a separates the decision-side effect. MSR reduces missed prior-memory use relative to Prompted Gate (0.407 to 0.224), which explains much of the Prior-memory gain in Table 2. At the same time, MSR is not merely a conservative router: it has higher unnecessary prior-memory use than Prompted Gate, yet still achieves higher Recent-context accuracy than that baseline. Figure 4b summarizes errors: MSR has the lowest total wrong-answer rate, and many residual errors occur after the correct source is selected. The bottleneck is therefore source-specific: Prior-memory items mainly require deciding to recall, while Recent-context items require using selected visible evidence without retrieved-memory or nearby-context drift.

Table 4: MSR component ablations on GPT-5-mini. Gray values denote decreases from Full MSR.

Variant	Rec.	Prior	Overall	Sel.
Full MSR	**0.617**	**0.583**	**0.600**	**0.817**
w/o atom split	0.533 ↓.084	0.483 ↓.100	0.508 ↓.092	0.758 ↓.059
w/o gap sel.	0.533 ↓.084	0.583 ↓.000	0.558 ↓.042	0.817 ↓.000
w/o vis. pack	0.483 ↓.134	0.550 ↓.033	0.517 ↓.083	0.808 ↓.009
w/o exch. exp.	0.617 ↓.000	0.500 ↓.083	0.558 ↓.042	0.808 ↓.009



Figure 5: Weighted error-category proportions among wrong answers, by method and source regime. Within each row, the four error-category proportions sum to 1. Boldface marks MSR. Appendix A.5 reports counts and estimated absolute error rates.

5.4 Phenomenon-Level Error Analysis

We next characterize residual wrong answers by assigning each annotated incorrect response to one dominant failure phenomenon within its source regime. Figure 5 reports their row-normalized composition. Appendix A.5 provides raw annotated counts and estimated absolute rates over all questions in each regime, linking error composition to full-workload error rates.

On Recent-context items, MSR reduces branch and wrong-target errors, leaving visible-support or anchor misses as a larger share of residual errors. On Prior-memory items, it reduces missing-fact errors relative to No Memory and Prompted Gate; remaining errors concentrate on exactness misses and visible-distractor misuse. These shifts distinguish source-selection failures from evidence-use failures within the selected source, motivating the component study below.

5.5 Component Attribution

Table 4 isolates MSR's components under the same GPT-5-mini protocol. Removing gap selection preserves aggregate Selection Acc. but lowers Overall and Recent-context accuracy. Removing the visible support pack mainly hurts Recent-context accuracy, consistent with the anchor misses in Figure 5. Removing support-anchored exchange expansion mainly hurts Prior-memory accuracy, where isolated question-ranked hits lose the local exchange context. Together, these ablations show that MSR benefits from both source selection and source-specific grounding. Appendix D.5 tests reattaching visible context to the prior-memory branch.

5.6 External and Reference Evidence

For LongMemEval-M (Figure 6a), we sample items whose answer session precedes the latest session, preserving the original question, gold answer, and evaluation format. Either session serves as visible context, with earlier turns as prior memory. Panel (b) reports native memory systems' accuracy, tokens, and latency on MSBench under heterogeneous configurations.

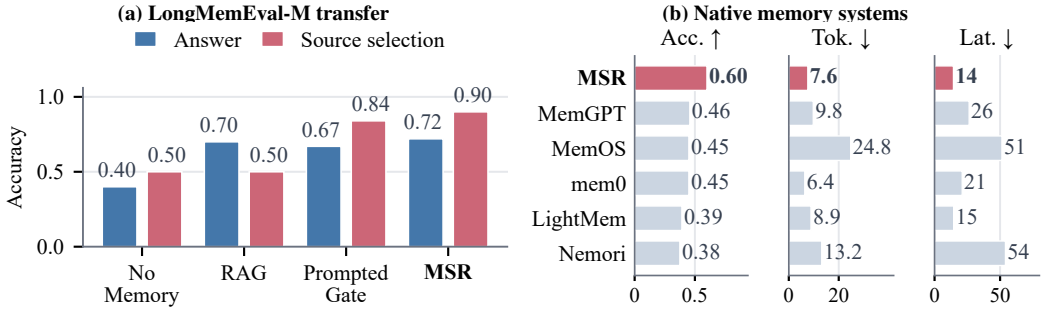

Figure 6: Transfer and native memory-system results. (a) LongMemEval-M answer and source-selection accuracy under the MSBench source boundary. (b) MSBench accuracy, tokens (k), and latency (s); costs are per question.

On LoCoMo [25] and QMSum [40], the four policies cover multi-session dialogue and multi-party meetings under the shared source-choice protocol. MSR leads on Answer and Selection Accuracy on both benchmarks (Table 5(a,b); CIs in Appendix E). Relative to Prompted Gate, LoCoMo shows a larger improvement in source selection than in answer accuracy, while QMSum shows gains in both. Together with LongMemEval-M, these results extend the evaluation beyond the personalized dialogue histories used to construct MSBench.

The controlled cost comparison separates MSR's inference overhead from the heterogeneous memory-system references. MSR uses two backbone calls per question and 7.6k tokens, compared with Prompted Gate's 5.8k. Its mean latency is 14.4 seconds, close to Prompted Gate's 14.5 seconds and 1.1 seconds above RAG. Thus, the accuracy gain comes with higher token use but comparable measured latency to the prompted gate (Appendix D.3).

Table 5: Accuracy on LoCoMo and QMSum (%). Bold values indicate the best result.

(a) LoCoMo

Method	Answer Acc.	Selection Acc.
No Memory	53.7	50.0
RAG	76.9	50.0
Prompted Gate	78.6	77.0
MSR	**80.7**	**85.9**

(b) QMSum

Method	Answer Acc.	Selection Acc.
No Memory	32.7	50.0
RAG	44.0	50.0
Prompted Gate	41.3	71.3
MSR	**55.3**	**88.7**

Extension to abstention. MSR can abstain when neither source supports an answer. On balanced answerable and unanswerable questions with GPT-5-mini, Extended MSR reaches 93.1% action accuracy and 89.7% abstention recall, while answering 96.6% of answerable controls (Figure 7). Binary MSR always answers; answer rate measures response frequency, not correctness. Appendix C.5 details the protocol and mixed-source results.

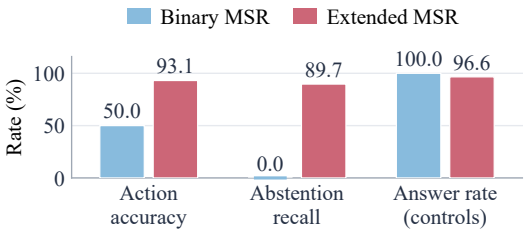

Figure 7: Abstention extension on GPT-5-mini.

6 Conclusion

This paper studies answer-time memory-source choice in long-context dialogue: blind retrieval can overwrite sufficient recent context, while skipping retrieval fails when prior evidence is needed. MSBench makes these complementary failures measurable under source-bounded replay, and MSR addresses them with a reusable source-monitoring state that couples source selection with visible anchoring and support-anchored expansion. Across backbones, ablations, and diagnostics, the results show that dialogue-memory evaluation should test not just memory recall, but whether the chosen source provides the evidence needed for a correct answer at the current dialogue checkpoint.

Acknowledgments and Disclosure of Funding

This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project (2026ZD0127802), in part by the Science Fund for Creative Research Groups of the National Natural Science Foundation of China (62521006), in part by the National Natural Science Foundation of China (62276077, 62406091, U23B2055, 62350710797), in part by Guangdong S&T Program (2024B0101050003), in part by the Guangdong Basic and Applied Basic Research Foundation (2024A1515011205, 2026A1515011718), in part by Guangdong Provincial Key Laboratory of Artificial General Intelligence and Large Models (2026B1212010005), and in part by the Shenzhen Science and Technology Program (KQTD20240729102154066).

To the best of the authors' knowledge, there are no competing financial interests to disclose in relation to this work.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023. URL https://arxiv.org/abs/2310.11511.

[2] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2024. URL https://arxiv.org/abs/2308.14508.

[3] Zihao Cheng, Weixin Wang, Yu Zhao, Ziyang Ren, Jiaxuan Chen, Ruiyang Xu, Shuai Huang, Yang Chen, Guowei Li, Mengshi Wang, Yi Xie, Ren Zhu, Zeren Jiang, Keda Lu, Yihong Li, Xiaoliang Wang, Liwei Liu, and Cam-Tu Nguyen. Lifebench: A benchmark for long-horizon multi-source memory, 2026. URL https://arxiv.org/abs/2603.03781.

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

[5] Bin Deng, Yizhe Feng, Zeming Liu, Qing Wei, Xiangrong Zhu, Shuai Chen, Yuanfang Guo, and Yunhong Wang. RETAIL: Towards real-world travel planning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14870–14902. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.752. URL https://aclanthology.org/2025.emnlp-main.752/.

[6] Shihan Dou, Ming Zhang, Zhangyue Yin, Chenhao Huang, Yujiong Shen, Junzhe Wang, Jiayi Chen, Yuchen Ni, Junjie Ye, Cheng Zhang, Huaibing Xie, Jianglu Hu, Shaolei Wang, Weichao Wang, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Pluto Zhou, Tao Gui, Zuxuan Wu, Xipeng Qiu, Qi Zhang, Xuanjing Huang, Yu-Gang Jiang, Di Wang, and Shunyu Yao. Cl-bench: A benchmark for context learning, 2026. URL https://arxiv.org/abs/2602.03587.

[7] Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation, 2025. URL https://arxiv.org/abs/2510.18866. ICLR 2026.

[8] 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.

[9] Chuanrui Hu, Tong Li, Xingze Gao, Hongda Chen, Yi Bai, Dannong Xu, Tianwei Lin, Xiaohong Li, Yunyun Han, Jian Pei, and Yafeng Deng. Evaluating long-horizon memory for multi-party collaborative dialogues, 2026. URL https://arxiv.org/abs/2602.01313.

[10] Jinchao Hu, Meizhi Zhong, Kehai Chen, and Min Zhang. SearchSkill: Teaching LLMs to use search tools with evolving skill banks. *arXiv preprint arXiv:2605.09038*, 2026. URL https://arxiv.org/abs/2605.09038.

[11] Sen Hu, Zhiyu Zhang, Yuxiang Wei, Xueran Han, Zhenheng Tang, Huacan Wang, and Ronghao Chen. Clonemem: Benchmarking long-term memory for ai clones, 2026. URL https://arxiv.org/abs/2601.07023.

[12] 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.

[13] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity, 2024. URL https://arxiv.org/abs/2403.14403.

[14] Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale, 2025. URL https://arxiv.org/abs/2504.14225.

[15] Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, Radha Poovendran, Gregory Wornell, Lyle Ungar, Dan Roth, Sihao Chen, and Camillo Jose Taylor. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*, 2025. URL https://arxiv.org/abs/2512.06688.

[16] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation, 2023. URL https://arxiv.org/abs/2305.06983.

[17] Cheng Jiayang, Dongyu Ru, Lin Qiu, Yiyang Li, Xuezhi Cao, Yangqiu Song, and Xunliang Cai. Amemgym: Interactive memory benchmarking for assistants in long-horizon conversations, 2026. URL https://arxiv.org/abs/2603.01966.

[18] Jiho Kim, Woosog Chay, Hyeonji Hwang, Daeun Kyung, Hyunseung Chung, Eunbyeol Cho, Yeonsu Kwon, Yohan Jo, and Edward Choi. Dialsim: A dialogue simulator for evaluating long-term multi-party dialogue understanding of conversational agents, 2025. URL https://arxiv.org/abs/2406.13144.

[19] Dong-Ho Lee, Adyasha Maharana, Jay Pujara, Xiang Ren, and Francesco Barbieri. Realtalk: A 21-day real-world dataset for long-term conversation, 2025. URL https://arxiv.org/abs/2502.13270.

[20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

[21] Xingzuo Li, Kehai Chen, Yunfei Long, Xuefeng Bai, Yong Xu, and Min Zhang. Generator-assistant stepwise rollback framework for large language model agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17683–17700. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.892. URL https://aclanthology.org/2025.emnlp-main.892/.

[22] Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. Memos: An operating system for memory-augmented generation (mag) in large language models, 2025. URL https://arxiv.org/abs/2505.22101.

[23] Zhengxi Lu, Fei Tang, Guangyi Liu, Jin Ma, Kaitao Song, Xu Tan, Wenqi Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. UI-Copilot: Advancing long-horizon GUI automation via tool-integrated policy optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19741–19762. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.904. URL https://aclanthology.org/2026.acl-long.904/.

[24] Wenquan Ma, Jiayan Nan, Wenlong Wu, and Yize Chen. What deserves memory: Adaptive memory distillation for llm agents, 2026. URL https://arxiv.org/abs/2508.03341.

[25] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

[26] Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. Nolima: Long-context evaluation beyond literal matching, 2025. URL https://arxiv.org/abs/2502.05167.

[27] Thomas O Nelson and Louis Narens. Metamemory: A theoretical framework and new findings. In *Psychology of learning and motivation*, volume 26, pages 125–173. Elsevier, 1990.

[28] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

[29] Egor Pakhomov, Erik Nijkamp, and Caiming Xiong. Convomem benchmark: Why your first 150 conversations don't need rag. *arXiv preprint arXiv:2511.10523*, 2025. URL https://arxiv.org/abs/2511.10523.

[30] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. On memory construction and retrieval for personalized conversational agents, 2025. URL https://arxiv.org/abs/2502.05589.

[31] Yiting Shen, Kun Li, Wei Zhou, and Songlin Hu. Mem2actbench: A benchmark for evaluating long-term memory utilization in task-oriented autonomous agents, 2026. URL https://arxiv.org/abs/2601.19935.

[32] Yihong Tang, Kehai Chen, Muyun Yang, Zheng-Yu Niu, Jing Li, Tiejun Zhao, and Min Zhang. Thinking in character: Advancing role-playing agents with role-aware reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3926.

[33] Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Zhenhe Wu, ShuangZhi Wu, Zejun Ma, and Zhoujun Li. SCM: Enhancing large language model with self-controlled memory framework. In *Database Systems for Advanced Applications (DASFAA 2025)*, volume 15991 of *Lecture Notes in Computer Science*, pages 188–203. Springer, 2026. doi: 10.1007/978-981-95-4158-4_12.

[34] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory, 2023. URL https://arxiv.org/abs/2306.07174.

[35] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024. URL https://arxiv.org/abs/2410.10813.

[36] Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. 2021. URL https://arxiv.org/abs/2107.07567.

[37] Mufan Xu, Kehai Chen, Jiahao Hu, Xinchao Xu, Muyun Yang, Tiejun Zhao, and Min Zhang. User-aware active knowledge acquisition for emotional support dialogue. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 140750–140774. PMLR, 2026. URL https://proceedings.mlr.press/v306/xu26f.html.

[38] Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, Yuandong Tian, and Jishen Zhao. Ama-bench: Evaluating long-horizon memory for agentic applications, 2026. URL https://arxiv.org/abs/2602.22769.

[39] Jianjie Zheng, Zhichen Liu, Zhanyu Shen, Jingxiang Qu, Guanhua Chen, Yile Wang, Yang Xu, Yang Liu, and Sijie Cheng. Evaluating memory capability in continuous lifelog scenario. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 7063–7089. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.351. URL https://aclanthology.org/2026.findings-acl.351/.

[40] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.472. URL https://aclanthology.org/2021.naacl-main.472/.

[41] 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.

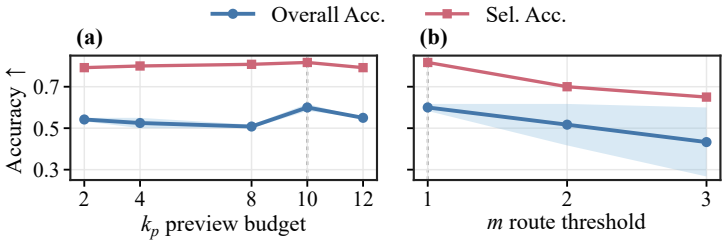

Figure 8: MSR sensitivity to (a) preview budget k_p and (b) route threshold m, varied one at a time. Shaded bands span Recent-context and Prior-memory answer accuracies; dashed lines mark defaults.

A Additional Results

A.1 Hyperparameter Sensitivity

The sensitivity sweep in Figure 8 varies one MSR control parameter at a time while holding the remaining protocol fixed. The shaded band spans Recent-context and Prior-memory answer accuracies, and dashed lines mark defaults. The preview budget k_p must be large enough to expose support gaps. We additionally sweep a generalized route threshold m, the minimum number of actionable gaps needed to enter the prior-memory branch; the main MSR setting uses $m = 1$. Increasing m makes MSR too conservative, reducing Prior-memory support and lowering Overall Acc. This pattern is consistent with the method design: the critical control is reliable source-monitoring evidence, not simply retrieving more items.

A.2 Existing Assets and Terms

MSBench is constructed from PersonaMem v2 dialogue scenes [15]; the public dataset release at `https://huggingface.co/datasets/bowen-upenn/PersonaMem-v2` lists the dataset license as CC BY 4.0. Reference memory-system code used for baselines is credited through the original publications and repository notices: mem0 [4] is Apache-2.0, MemGPT [28] is Apache-2.0, LightMem [7] is MIT, Nemori [24] is MIT, and MemOS [22] is Apache-2.0. Model calls for embeddings, answer generation, routing, question drafting, and judging use commercial API services under the corresponding model-provider terms. These include OpenAI service terms (`https://openai.com/policies/service-terms/`) for GPT-family models, embeddings, and judges; Google Gemini API terms (`https://ai.google.dev/gemini-api/terms`) for Gemini; Alibaba Cloud Model Studio related agreements (`https://www.alibabacloud.com/help/en/model-studio/related-agreements`) for Qwen-family models; and DeepSeek Open Platform terms (`https://cdn.deepseek.com/policies/en-US/deepseek-open-platform-terms-of-service.html`) for DeepSeek-family models. The artifact release preserves third-party notices and license files for redistributed code or derived data.

A.3 Broader Impacts

Better memory-source choice can make personalized dialogue agents more faithful to the user's current state by reducing unnecessary prior-memory use and unsupported recall. The same capability also raises risks if deployed in sensitive settings: incorrect source choices can reinforce outdated personal information, expose irrelevant prior context, or make an assistant appear more certain about a user's history than the evidence supports. MSBench is a replay benchmark and does not deploy agents with users, but downstream systems should pair source-choice control with privacy safeguards, user-facing correction mechanisms, and conservative abstention when neither visible context nor prior memory provides adequate support.

A.4 Limitations

First, as a fully training-free and plug-and-play framework, MSR's upper bound naturally correlates with the instruction-following, evidence attribution, and long-context reasoning capabilities of

the underlying LLM. This dependence is also a strength: MSR does not require model-specific fine-tuning, memory rewriting, or changes to the memory architecture, so future improvements in foundation models can directly unlock stronger source monitoring and source-specific answering without modifying the framework.

Second, MSR currently focuses on a compact source-choice space: answer from visible recent context when it is sufficient, or consult retrievable prior memory when earlier evidence is needed. This abstraction intentionally keeps the method lightweight and interpretable while still supporting visible evidence anchoring and support-anchored exchange expansion. Richer actions, such as partial retrieval, clarification, or abstention, can be layered on top of the same source-monitoring state when applications require finer-grained control. Appendix C.5 evaluates mixed-source answering and abstention; clarification remains a mechanism-level extension.

Third, MSBench uses LLM-assisted construction and judging to scale beyond purely manual benchmark creation. We pair this with human review, shortcut probing, judge self-consistency checks, and a blinded human audit in Section 5.2. Larger future releases can further increase the human-audit budget and domain coverage, while preserving the current benchmark's central advantage: controlled measurement of whether an agent remembers only when memory is actually needed.

Finally, MSBench provides controlled evidence for source choice through 45 complete multi-session histories from PersonaMem-v2. The transfer evaluations on LongMemEval-M, LoCoMo, and QMSum extend this evidence to factual long-context question answering, multi-session dialogue, and multi-party meetings, showing that MSR's gains are not confined to the construction corpus. These evaluations complement the controlled source-choice setting; coverage of broader dialogue domains remains a direction for further validation.

A.5 Error Counts and Absolute Rates

Table 6 reports raw annotated counts and estimated absolute rates, using the weighting of Figure 5 and a denominator of 450 questions per regime. Each baseline contributes 120 annotated errors per regime, reweighted to full-replay error totals; MSR contributes all 23 Recent-context and 25 Prior-memory errors from the six-scene diagnostic, each weighted by 7.5. Total gives the annotated error count and estimated total error rate.

B MSBench Release Details

B.1 MSBench Release Statistics

Table 7 reports the current release by source regime and question family.

B.2 MSBench Review Checklist

Before inclusion, each item is manually reviewed against its replay state. Review verifies that a Prior-memory item is tied to aligned non-visible evidence, that this decisive support is absent from

Table 6: Error counts and estimated absolute error rates by source regime. Each cell reports the annotated count / estimated rate (% of all questions in that regime).

Recent-context	Off-branch	Wrong target	Stale state	Anchor miss	Total
No Memory	35 / 14.4	45 / 18.5	6 / 2.5	34 / 14.0	120 / 49.3
RAG	32 / 15.6	48 / 23.5	10 / 4.9	30 / 14.7	120 / 58.7
Prompted Gate	32 / 12.7	58 / 23.1	9 / 3.6	21 / 8.4	120 / 47.8
MSR	3 / 5.0	4 / 6.7	0 / 0.0	16 / 26.7	23 / 38.3
Prior-memory	Missing fact	Visible distr.	Exact miss	Synth. fail.	Total
No Memory	113 / 89.1	7 / 5.5	0 / 0.0	0 / 0.0	120 / 94.7
RAG	43 / 16.6	1 / 0.4	29 / 11.2	47 / 18.2	120 / 46.4
Prompted Gate	55 / 26.9	10 / 4.9	16 / 7.8	39 / 19.1	120 / 58.7
MSR	1 / 1.7	10 / 16.7	13 / 21.7	1 / 1.7	25 / 41.7

Table 7: MSBench release statistics by source regime and question family.

Split	Items	Single detail	Multi-part	Referenced item	Updated state
Recent-context	450	48	84	27	291
Prior-memory	450	58	112	85	195
Overall	900	106	196	112	486

Table 8: Visible-context audit for Prior-memory questions. Counts and percentages use all 450 questions in this regime.

Relation to the requested prior fact	Count	Share
Unrelated	219	48.7%
Related but insufficient	47	10.4%
Same topic, wrong instance/version	184	40.9%

the visible window C_{t_i}, and that the paired Recent-context item is answerable from C_{t_i} alone while reflecting the latest visible state after updates, overrides, removals, or resets. Review also keeps gold answers concise and evidence-specific so that the visible-versus-retrievable distinction remains sharp.

B.3 Misleading Recent Context Audit

We audited all 450 Prior-memory questions, classifying the visible context by its relation to the requested prior fact. Table 8 reports the three categories.

The two related-context categories together account for 51.3% of Prior-memory questions. In the wrong-instance/version category, the visible window mentions a related topic, entity, or event, but the evidence for the instance requested by the question remains in prior memory. This construction distinguishes topical relevance from support for the requested answer.

C MSR Inference Details

This appendix makes the MSR decision pipeline concrete through the decision interface, inference algorithm, and worked examples for the two answer branches, followed by support-identifier validation and a human audit.

C.1 Decision Interface

The source-monitoring call returns a JSON object with a single top-level field, `atoms`. Each element in `atoms` is one question-conditioned atom record.

Schema

Top-level object. `{"atoms": [atom_record, ...]}`

```
{"atom_id": "A1", "text": "the target language is Ukrainian",
  "visible_evidence_ids": ["visible:1"],
  "preview_evidence_ids": []}
```

- `atom_id`: unique identifier for one atom within the current question.
- `text`: one atomic question. It should preserve exactness, cardinality, ordering, and temporal qualifiers when they matter.
- `visible_evidence_ids`: supporting evidence ids from the visible source. An empty list means the visible context does not ground the atom.
- `preview_evidence_ids`: supporting evidence ids from the preview source. An empty list means the preview does not ground the atom.

Each source contributes at most one supporting evidence id per atom. This id is used to build the source-monitoring state: visible ids anchor the visible support pack, while preview ids identify prior-memory turns that seed support-anchored exchange expansion for the prior-memory branch. Valid ids take the form `visible:i` for visible evidence and `preview:j` for preview evidence.

The complete template below is built by `_build_voi_atom_support_prompt` in the standard `atom_split` mode. At each checkpoint, the placeholders are filled with the current question, a JSON object listing the allowed visible and preview evidence IDs, and the formatted evidence text from each source, with source-specific identifiers attached to the corresponding dialogue turns.

Complete Source-Monitoring Prompt

```
You are an evidence-atom and support-attribution judge for memory-source
selection.
Decompose the question into atomic questions.
For each atom, provide one source-level support indicator per source, not an
exhaustive evidence list.
Each evidence-id array must be either [] or ["one_id"]; never return two or
more ids in either array.
If several evidence items from the same source support an atom, choose the
single most direct support id.
Use [] when that source does not support the atom.
Preserve exactness, cardinality, ordering, and temporal qualifiers when the
question requires them.
Always return at least one atom.  The atoms array must be non-empty; never
return {"atoms":  []}.
Atoms are derived from the question, not from whether evidence is present.
If no source supports an atom, keep the atom and set that source's evidence-id
array to [].
Do not invent ids.  Use only evidence ids that appear in the allowed-id lists
below.
Before returning, verify that every visible_evidence_ids and
preview_evidence_ids array has length 0 or 1.
When a source supports an atom, replace [] with ["one allowed id"] for that
source.
Return only a JSON object with this schema:
{"atoms":  [{"atom_id":  "A1", "text":  "the required fact from the question",
"visible_evidence_ids":  [], "preview_evidence_ids":  []}]}
Question:  {question}
ALLOWED IDS:
{allowed_ids_json}
VISIBLE EVIDENCE:
{visible_text}
PREVIEW EVIDENCE:
{preview_text}
```

C.2 Inference Algorithm

MSR inference at checkpoint t can be summarized as Algorithm 1. Here $e_t^C(a)$ and $e_t^P(a)$ denote the visible and preview support identifiers attached to atom a; either identifier may be empty.

Algorithm 1 MSR inference at checkpoint t: source monitoring, route projection, and branch-specific grounding.

Require: current question q_t, visible context C_t, prior memory $M_{<t}$, budgets k_p, k_r
Ensure: final answer $\hat{y}_t$

Preview stage

Exclude visible turns from the prior-memory retrieval pool
Retrieve a small hidden-only preview P_t with budget k_p

Decision stage

Run the source-monitoring call on (q_t, C_t, P_t)
Extract source-monitoring state $Z_t = \{(a, e_t^C(a), e_t^P(a)) : a \in \mathcal{A}(q_t)\}$
Let $L_t \leftarrow \{a : e_t^C(a) \neq \emptyset\}$ and $F_t \leftarrow \{a : e_t^C(a) = \emptyset,\ e_t^P(a) \neq \emptyset\}$ and $U_t \leftarrow \{a : e_t^C(a) = \emptyset,\ e_t^P(a) = \emptyset\}$
Compute $G_t(q_t) \leftarrow |F_t|/|\mathcal{A}(q_t)|$

Selection-conditioned grounding

if $F_t = \emptyset$ **then**
    Build visible support pack S_t^C from $\{e_t^C(a) : a \in L_t\}$
    Answer from anchored visible evidence with C_t as background, leaving U_t unsupported: $\hat{y}_t \leftarrow \mathrm{Ans}_C(q_t, S_t^C, C_t)$
else
    Retrieve question-ranked hits $Q_t \leftarrow \mathrm{TopK}_{k_r}(q_t, M_{<t})$
    Expand preview support ids into local exchanges $X_t \leftarrow \mathrm{ExpandExch}(\{e_t^P(a) : a \in F_t\}, M_{<t})$
    Set $R_t \leftarrow \mathrm{Merge}^{X \prec Q}(X_t, Q_t)$ by deduplicating, placing support-anchored exchanges first, preserving local exchange order, and preserving question rank for remaining hits
    Answer from retrieved prior-memory evidence: $\hat{y}_t \leftarrow \mathrm{Ans}_M(q_t, R_t)$
end if
return $\hat{y}_t$

C.3 Worked Examples

All examples below are drawn from the real MSBench runs used in our experiments. We preserve the original question text and lightly trim visible or preview evidence snippets for readability. The examples emphasize how the same source-monitoring state drives source selection and branch-specific evidence construction.

Case A: Stay on Anchored Visible Evidence

Question. *"From that translation request, give three elements in one line: target language, small-animal care setting, and requested mood cue."*

Visible context. The final two visible turns already contain the requested triple: the user asks for a Ukrainian translation of a scene about *arranging small shelters and petting shy tails*, and asks for a *gentle, trusting, quietly tender* mood.

Hidden-only preview. The preview may contain older related snippets, but it does not answer any atomic question that is missing from the visible window.

Source-monitoring state.

A_1 target language = Ukrainian
$e_t^C =$ `visible:1`, $e_t^P = \emptyset$, $A_1 \notin F_t$

A_2 small-animal care setting = arranging shelters and petting shy tails
$e_t^C =$ `visible:1`, $e_t^P =$ `preview:2`, $A_2 \notin F_t$

A_3 mood cue = gentle / trusting / quietly tender
$e_t^C =$ `visible:2`, $e_t^P = \emptyset$, $A_3 \notin F_t$

Computed gain. Every required atom is already grounded by the visible context, so no atom belongs to F_t. Therefore $F_t = \emptyset$ and $G_t(q_t) = 0$.

Answer branch. MSR builds the visible support pack S_t^C from `visible:1` and `visible:2`; older related preview evidence is not admitted because it adds no missing support. The final answer is *"Ukrainian – small-animal pet-care setting (arranging little shelters and petting shy tails) – gentle, trusting, quietly tender mood."*

Case B: Retrieve from a Single Preview-Supported Gap

Question. *"What follow-up option did the assistant offer after listing the breakfast ideas?"*

Visible context. The visible recent context window is about an unrelated Hindi-blog translation and a Friday-evening reflection. It contains no breakfast discussion and therefore no answer to the follow-up-option question.

Hidden-only preview. The top preview hit is a prior breakfast-guidance turn that ends with *"Would you like me to also suggest make-ahead options so you can save time in the morning?"*

Source-monitoring state.

A_1 the assistant offered make-ahead breakfast options to save time in the morning
$e_t^C = \emptyset$, $e_t^P = \texttt{preview:1}$, $A_1 \in F_t$

Computed gain. The only atomic question is unsupported in the visible context but grounded by the preview. Hence $F_t = \{A_1\}$ and $G_t(q_t) = 1$.

Answer branch. MSR selects the prior-memory branch. The preview support id `preview:1` seeds the support-anchored exchange set X_t, while the original question also retrieves question-ranked hits Q_t. The final answer is *"They asked if you would like suggestions for make-ahead breakfast options to save time in the morning."*

Case C: Retrieve and Expand Multi-Atom Support

Question. *"For presentation anxiety, two discreet grounding moves were recommended that would not be obvious to the audience."*

Visible context. The visible turns are currently discussing Korean drama recommendations, so they do not ground the presentation-anxiety topic or the requested coping moves.

Hidden-only preview. The highest-ranked preview items recover the earlier advice turn containing a *5-4-3-2-1 sensory check* and a *micro muscle relaxation* such as pressing the toes into the floor, together with the follow-up request for subtle strategies that would not be obvious to the audience.

Source-monitoring state.

A_1 the answer must identify exactly two moves
$e_t^C = \emptyset$, $e_t^P = \texttt{preview:1}$, $A_1 \in F_t$

A_2 one move is a silent 5-4-3-2-1 sensory check
$e_t^C = \emptyset$, $e_t^P = \texttt{preview:1}$, $A_2 \in F_t$

A_3 the other move is a subtle micro muscle relaxation
$e_t^C = \emptyset$, $e_t^P = \texttt{preview:1}$, $A_3 \in F_t$

A_4 both moves should be discreet / not obvious to the audience
$e_t^C = \emptyset$, $e_t^P = \texttt{preview:2}$, $A_4 \in F_t$

Computed gain. All atomic questions are unsupported by the visible context and supported by hidden-only preview evidence, so every atom belongs to F_t and $G_t(q_t) = 1$.

Answer branch. MSR retrieves from prior memory and expands the preview-supported advice turn into its surrounding exchange, preserving both the two techniques and the subtlety constraint. The final answer is *"Try a silent 5-4-3-2-1 sensory check and a micro muscle relaxation, such as subtly pressing your toes into the floor or making a small fist and releasing."*

C.4 Support-Identifier Validation and Human Audit

Support identifiers refer to the fixed evidence units supplied to the monitor. For each atom and source, the monitor selects the single most direct supporting identifier, or leaves the field empty. Support must satisfy the atom's facts and constraints, including the requested instance and temporal

qualifiers; topical overlap alone is insufficient. Deterministic checks enforce unique atom identifiers, membership in the allowed evidence set for each source, and at most one pointer per source.

Two annotators independently assessed whether the selected evidence supported its atom and whether support was present in each source, followed by adjudication of disagreements. Selected-ID precision is the fraction of selected identifiers whose evidence supports the corresponding atom. Support recall measures how often existing support is recognized; support specificity measures how often its absence is correctly recognized. Table 9 reports the results against the adjudicated judgments. Recomputing routes from those judgments yields 99.3% agreement with MSR.

Routing follows the support states rather than numerical confidence scores:

- **Visible support present:** the atom is covered and does not trigger retrieval, whether or not the preview also supports it.
- **Visible support absent, preview support present:** the atom creates an actionable gap and triggers the prior-memory branch.
- **Neither source supports the atom:** it remains unresolved and does not itself trigger retrieval under the binary rule.

A question takes the prior-memory branch if any atom has an actionable gap. When both sources support an atom, visible support has already closed that gap, so no tie-break is needed for that atom. For versioned information in MSBench, the latest valid update defines the current state; older memory cannot override a supported current update.

C.5 Extensions Beyond Binary Source Choice

MSBench holds answerability fixed to isolate source-choice errors. The atom-level state also supports richer actions: retain evidence for requirements already covered by visible context, retrieve support for uncovered requirements, and abstain when the available sources cannot support an answer. An unresolved requirement that the user can supply can instead prompt a targeted clarification. This clarification policy is a mechanism-level extension; the diagnostics below evaluate mixed-source answering and abstention.

We evaluate both extensions with GPT-5-mini. The mixed-source diagnostic holds the source-monitoring state, hidden preview, and retrieved items fixed, changing only the final grounding: Extended MSR retains visible support while incorporating memory support for the missing component. The answerability diagnostic adds an `ANSWER/ABSTAIN` decision on a balanced set of answerable and unanswerable questions. Binary MSR has no abstention action and always answers on this diagnostic.

Visible- and prior-memory-component accuracy score the two source-dependent subanswers separately; both-components-correct requires both to be correct. Action accuracy measures whether the `ANSWER/ABSTAIN` decision matches answerability. Abstention recall is the fraction of unanswerable questions on which the model abstains. Answer rate on answerable controls measures how often the model chooses `ANSWER`, not whether that answer is correct.

Extended MSR improves the both-components-correct rate by 50.0 percentage points (paired 95% CI [32.5, 65.0], exact paired $p = 1.1 \times 10^{-5}$). It also improves answer-or-abstain decisions while answering 96.6% of answerable controls, so the improvement does not arise from indiscriminate abstention. These diagnostics show how the same support state can guide component-specific grounding and answerability decisions beyond the binary source-choice setting.

Table 9: Human audit of atom-level support assignments. All values are percentages.

Audit measure	Visible context	Hidden preview
Selected-ID precision	98.7	99.3
Support recall	98.1	99.3
Support specificity	98.6	99.4

Table 10: Mixed-source answer accuracy with GPT-5-mini (%). Abstention results appear in Figure 7.

Metric	Binary MSR	Extended MSR
Visible-component accuracy	20.0	82.5
Prior-memory-component accuracy	62.5	65.0
Both components correct	2.5	52.5

D Baseline Implementation Details

D.1 Prompted LLM Retrieval Gate

The prompted gate uses the same answer backbone and visible dialogue as the other controlled policies. Its first call either produces the answer directly or requests one `retrieve_memory` call. If retrieval is requested, the returned memory is supplied to the same backbone together with the visible dialogue, and further retrieval is disabled. The gate has no hidden memory preview or atom-level support state.

The following user-message template is assembled by `build_vanilla_control_prompt` in tool mode. The placeholders contain the 16-turn visible dialogue and the original question.

Prompted LLM Retrieval Gate: complete user-message template

```
You are an intelligent assistant handling the current question using SHORT
CONTEXT first.
# CONTEXT
SHORT CONTEXT contains recent conversation turns.
# RULES
- Identify exactly which fact(s) are required by the question.
- Use SHORT CONTEXT first and avoid unsupported claims.
- If SHORT CONTEXT is missing required evidence, use memory evidence.
- If evidence conflicts, prioritize the most recent valid update.
- If evidence remains insufficient, refuse briefly instead of guessing.
- Focus only on the final "Question:" shown at the end of this prompt.
- Final answer text must be exactly one natural sentence.
# OUTPUT MODE (tool)
- If required evidence is missing, you may call retrieve_memory once before
answering.
- At most one memory-evidence action is allowed.
- Keep retrieval query focused and minimal.
- If evidence is sufficient, answer directly without tool call.
- Return exactly one final answer sentence (do not output
RETRIEVE/NO_RETRIEVE).
SHORT CONTEXT:
{short_context}
Question: {question}
```

Retrieval tool schema

```
{
  "type": "function",
  "name": "retrieve_memory",
  "description": "Retrieve relevant memories from the current block.",
  "parameters": {
    "type": "object",
    "properties": {
      "query": {"type": "string"},
      "top_k": {"type": "integer"}
    },
    "required": ["query", "top_k"]
  }
}
```

The executor uses the tool's query to rank only the permitted prior-memory items with `text-embedding-3-small` and cosine similarity. Although the interface exposes `top_k`, the executor fixes the retrieval width at 20 regardless of the model-supplied value. At most one retrieval is executed per question. The model is free to answer without a tool call.

D.2 Input-Length Audit

Table 11 reports the input-token distributions for the four backbones and four source policies. Counts use the same `cl100k_base` tokenizer for the question, visible evidence, and stored retrieved evidence, excluding method instructions. The maximum observed input length is 8,740 tokens, and the truncation rate is 0% across all backbone–method combinations. All four backbones preserve the same visible/memory boundary without losing evidence to provider-specific limits.

Table 11: Input-token distributions in the context audit. P95 denotes the 95th percentile; values are rounded to whole tokens. The truncation rate is 0% for every setting.

Backbone	Method	Mean	P95	Maximum
GPT-5-mini	No Memory	2,107	3,515	3,680
	RAG	4,805	6,520	8,740
	Prompted Gate	3,208	6,275	8,225
	MSR	3,415	5,964	7,807
Gemini 3.1 Flash-Lite	No Memory	2,107	3,515	3,680
	RAG	4,806	6,520	8,740
	Prompted Gate	2,690	5,640	7,891
	MSR	3,410	5,902	7,640
Qwen3 235B-A22B	No Memory	2,107	3,515	3,680
	RAG	4,806	6,520	8,740
	Prompted Gate	2,652	5,868	7,945
	MSR	3,147	5,695	7,640
DeepSeek V4 Pro	No Memory	2,107	3,515	3,680
	RAG	4,806	6,520	8,740
	Prompted Gate	3,382	6,327	8,363
	MSR	3,419	5,985	7,797

D.3 Cost Comparison with Simple Source Policies

Table 12 complements the native memory-system references in Figure 6b with the costs of the simple source policies. MSR uses two backbone calls per question, increasing token use through repeated prompt input. Its reported latency is 14.4 seconds per question, close to Prompted Gate's 14.5 seconds and 1.1 seconds above RAG, while its Answer Accuracy reaches 60.0%, 12.6 percentage points above the strongest simple baseline.

Table 12: Answer accuracy and reported per-question cost on MSBench. Token counts are in thousands and latency is in seconds.

Method	Answer Acc. (%)	Tokens/item (k)	Latency/item (s)
No Memory	28.0	2.9	11.0
RAG	47.4	4.4	13.3
Prompted Gate	46.8	5.8	14.5
MSR	**60.0**	7.6	14.4

D.4 Adaptive Retrieval Comparison

We adapt FLARE [16] and Self-RAG [1] to the MSBench source-choice protocol. Self-RAG uses `SciPhi-Self-RAG-Mistral-7B-32k`; all other methods use its parent model, `SciPhi-Mistral-7B-32k`. All methods share the same 16-turn visible window, non-overlapping prior memory, top-20 retrieval, questions, and judging protocol. Table 13 reports the results.

Table 13: Adaptive retrieval comparison on MSBench with the SciPhi-Mistral model family. All values are percentages.

Method	Answer Acc.	Selection Acc.
No Memory	9.2	50.0
RAG	22.5	50.0
Prompted Gate	16.7	47.5
FLARE	17.5	50.0
Self-RAG	9.2	51.7
MSR	**29.2**	**80.0**

D.5 Visible-Context Reattachment

We test source separation by including the visible context C_t alongside the prior-memory evidence R_t in the prior-memory answer branch. Under otherwise identical conditions, overall answer accuracy falls from 60.0% to 55.0% (Table 14), supporting the role of source-specific grounding after route selection, when answers should use the evidence from the chosen prior-memory source.

Table 14: Effect of adding visible context to the prior-memory answer branch. Values are percentages.

Prior-memory branch input	Overall Acc.
Prior memory only (MSR)	**60.0**
Prior memory + visible context	55.0

E Confidence Intervals and Paired Tests

E.1 Main MSBench Results

For the main results in Table 2, we compute 95% and 99% confidence intervals using 10,000 checkpoint-level paired bootstrap resamples, preserving the Recent-context/Prior-memory question pair at each checkpoint. These intervals summarize uncertainty under resampling of the replay checkpoints. We assess differences between MSR and Prompted Gate using two-sided paired permutation tests that compare the two methods on matched replay checkpoints.

Table 15: Bootstrap confidence intervals for MSBench accuracy (%). Point estimates appear in Table 2. Dashes mark omitted selection intervals for the fixed No Memory and RAG policies (50.0%).

		Answer Accuracy		Selection Accuracy	
Backbone	Method	95% CI	99% CI	95% CI	99% CI
GPT-5-mini	No Memory	[25.8, 30.1]	[24.9, 30.7]	–	–
	RAG	[43.7, 51.1]	[42.5, 52.3]	–	–
	Prompted Gate	[43.7, 49.8]	[42.8, 50.7]	[72.7, 78.1]	[71.7, 79.1]
	MSR	[57.0, 63.0]	[55.7, 64.3]	[79.0, 84.1]	[78.0, 85.0]
Gemini 3.1 Flash-Lite	No Memory	[17.9, 22.1]	[17.3, 23.0]	–	–
	RAG	[34.1, 40.9]	[33.2, 42.1]	–	–
	Prompted Gate	[32.0, 38.0]	[31.1, 38.9]	[64.3, 69.1]	[63.6, 69.7]
	MSR	[40.8, 47.2]	[39.9, 48.4]	[76.5, 81.9]	[75.5, 82.5]
Qwen3 235B-A22B	No Memory	[19.5, 24.1]	[18.6, 24.7]	–	–
	RAG	[35.0, 41.7]	[34.1, 43.2]	–	–
	Prompted Gate	[21.5, 26.9]	[20.8, 27.9]	[52.6, 57.4]	[52.0, 58.0]
	MSR	[38.3, 45.0]	[37.4, 46.3]	[69.8, 75.2]	[68.8, 76.2]
DeepSeek V4 Pro	No Memory	[25.1, 29.6]	[24.5, 30.5]	–	–
	RAG	[40.3, 46.4]	[39.1, 47.6]	–	–
	Prompted Gate	[45.8, 52.5]	[44.6, 53.4]	[79.0, 84.1]	[78.0, 85.0]
	MSR	[52.5, 59.2]	[51.4, 60.2]	[79.8, 84.9]	[78.9, 85.7]

MSR improves Answer Accuracy over Prompted Gate by 6.6–17.5 percentage points across the four backbones. The paired difference intervals exclude zero at both the 95% and 99% levels, and all four two-sided paired tests give $p < 0.001$. For Selection Accuracy, the paired tests identify significant gains for MSR over Prompted Gate on GPT-5-mini, Gemini, and Qwen.

E.2 Transfer Results

Table 16 reports the 95% bootstrap confidence intervals for the LoCoMo and QMSum comparisons in Table 5, covering both answer correctness and source-selection accuracy on each benchmark.

Table 16: 95% bootstrap confidence intervals for transfer accuracy (%). Point estimates appear in Table 5. Dashes mark omitted selection intervals for the fixed No Memory and RAG policies (50.0%).

Benchmark	Method	Answer Accuracy	Selection Accuracy
LoCoMo	No Memory	[45.7, 61.7]	–
	RAG	[68.9, 84.9]	–
	Prompted Gate	[70.5, 85.7]	[70.0, 84.0]
	MSR	[72.6, 87.8]	[79.9, 91.9]
QMSum	No Memory	[26.7, 38.7]	–
	RAG	[33.3, 54.0]	–
	Prompted Gate	[32.7, 50.0]	[66.0, 77.3]
	MSR	[46.0, 64.7]	[84.0, 93.3]

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the memory-source choice problem, the MSBench benchmark, the MSR method, and the controlled empirical claims. The scope is bounded by the source-bounded replay setting and the appendix limitations subsection.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix A.4 discusses backbone dependence, the compact source-choice space, LLM-assisted benchmark construction and judging, and domain coverage. Appendix C.5 evaluates mixed-source answering and abstention extensions; clarification remains a mechanism-level extension.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper introduces a method, benchmark, and empirical analysis, but does not claim formal theoretical results or provide theorem-style proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3–5 specify the benchmark construction, method, baselines, metrics, judging protocol, and main controlled results. Section 5.2 reports benchmark validation and human-audit agreement; the appendix provides the decision schema, complete source-monitoring and prompted-gate templates, inference algorithm, release statistics, worked examples, and hyperparameter sensitivity.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The public project repository is `https://github.com/Jockeylove/MSBench`. The benchmark data, code, prompts, and evaluation scripts are planned for release but are not yet publicly available. The paper and appendix describe the method and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sections 3–5 specify the benchmark release, source regimes, baselines, answer backbones, judging protocol, and metrics. Section 5.2 reports benchmark validation and human-audit agreement; the appendix provides inference details, release statistics, worked examples, and hyperparameter sensitivity. There is no model training or optimizer in the proposed method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Appendix E reports 95% and 99% checkpoint-level paired bootstrap confidence intervals and two-sided paired permutation tests for the main MSBench results, as well as 95% bootstrap confidence intervals for LoCoMo and QMSum. The main-result resampling preserves the Recent-context/Prior-memory pair at each checkpoint.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: The appendix reports per-question token and latency cost profiles, but the manuscript does not fully specify worker type, memory, total execution time, or preliminary compute usage.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work evaluates replayed dialogue benchmark items and LLM-based systems without deploying agents or collecting new human-subject interaction data. It uses licensed public assets and API services under their stated terms, discloses LLM usage, and discusses risks from stale or overconfident memory use.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The appendix Broader Impacts subsection discusses potential benefits for faithful personalized dialogue and potential risks from stale personal information, irrelevant prior context, and overconfident memory use, along with deployment safeguards.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk pretrained model, image generator, or scraped unsafe dataset. MSR is an inference-time coordination method evaluated through a replay benchmark.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix A.2 lists the PersonaMem-v2 license, baseline code licenses, and model/API terms. The transfer benchmarks are credited in Section 5 and the references; their upstream licenses are MIT (LongMemEval), CC BY-NC 4.0 (LoCoMo), and MIT (QMSum). The project repository's Licensing & Data Attribution section records these terms, including LoCoMo's noncommercial restriction.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: MSBench is documented in Section 3 with source regimes, construction rules, quality-control criteria, and evaluation protocol. Section 5.2 reports benchmark validation and human-audit agreement; appendix sections further provide release statistics, the review checklist used for inclusion, inference details, and worked examples.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not report crowdsourcing experiments or research with human subjects. The manual benchmark review described in the paper is an internal quality-control step rather than a participant-facing study.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject experiments, so participant-risk disclosure and IRB approval are not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are central to the method and evaluation: MSR uses LLM calls for source monitoring and answer construction, while a deterministic rule maps the support state to the selected source. MSBench questions are LLM-drafted and manually reviewed, and answer correctness is judged with repeated LLM calls, as described in Sections 3–5 and the appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.