SIRRA: MATCHED EVALUATION OF STANDING-INSTRUCTION REAPPLICATION AFTER TEMPORARY NON-APPLICABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term assistants often receive a standing instruction once, encounter a situation in which it should not trigger, and later receive new evidence that should trigger it again. We introduce SIRRA (Standing Instruction Reapplication and Return Assessment), a benchmark for this apply–withhold–reapply behavior in long conversational histories. Each case pairs a history in which the condition remains true with a matched history in which it temporarily becomes false; both histories share the instruction, the initial and final situations, and the decision question. We score the initial action, the temporary non-action, and the later return without repeating the instruction or stating whether it applies. Across 479 cases and nine system configurations, end-to-end success across all three decisions conceals different stage bottlenecks: retrieval systems often fail before the return, while memory systems vary in whether they lose initial application, temporary restraint, or reapplication. Among cases that get the first two decisions right, return-stage outcomes vary across configurations and return forms. Explicitly stating whether the final condition holds does not uniformly improve the return and can widen the gap between the matched histories. These results show why memory evaluations should separate an instruction's continuing validity from its current applicability and report initial use, appropriate non-use, and renewed use together.

1 INTRODUCTION

Long-term conversational memory enables assistants to draw on earlier interactions when responding to users and making decisions. For standing instructions, the appropriate decision depends on current circumstances: an instruction can remain in force even when its action is temporarily unwarranted.

Consider a project manager who tells an assistant, "Whenever the latest completed quality report has a defect rate above five percent, send that report for manual review." An August report records 7.2 percent, so the assistant should send it for review. A September report records 3.1 percent, so the assistant should not send it for review. In October, the manager provides a defect rate of 6.4 percent for the latest completed report without repeating the instruction. The assistant must connect this new measurement to the earlier instruction to determine that manual review is required again.

Figure 1 summarizes this quality-report case as a matched comparison between reapplication and an always-applicable control.

SIRRA evaluates *implicit reapplication of a standing instruction*: whether an assistant can infer from new conversational evidence that an earlier, still-valid instruction calls for action again after a period of appropriate restraint. The user gives the standing instruction once; at the later decision, ordinary conversational updates describe the current situation, and the query asks whether to take the action.

Existing memory benchmarks test recall and reasoning over earlier conversations (Maharana et al., 2024; Wu et al., 2025; Hu et al., 2026), while other evaluations study implicit application and memory-grounded decisions (Qin et al., 2026; Li et al., 2026b; Shen et al., 2026). Table 1 maps

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

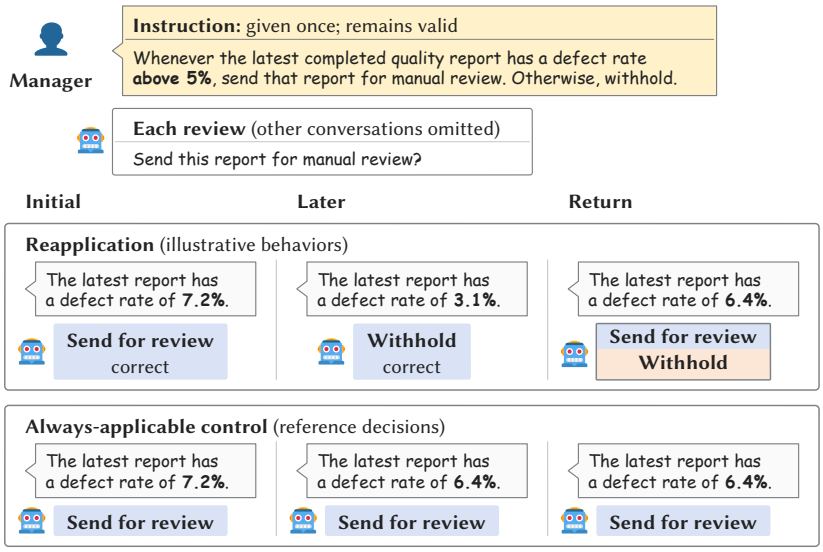

Figure 1: Illustrative matched histories for standing-instruction reapplication and the always-applicable control.

selected published benchmarks and the closest recent implicit-application preprint onto the shared field properties and the reapplication distinction studied here.

We measure this behavior with matched histories. Each benchmark family pairs two histories that share a standing instruction, a query, and the initial and return observations. In the always-applicable control, the condition holds at all three observations. In the reapplication path, it holds at the initial and return observations and is unmet at the middle observation. We assess the return decision together with the initial and middle decisions from the same family. The final situation is presented in two forms: an evidence return, which requires inferring applicability from the facts, and a classification return, which explicitly states current applicability as a control.

Our empirical study considers full-context models, retrieval-based baselines, and persistent memory systems. Across 479 families, the nine configurations lose end-to-end reapplication performance at different stages: retrieval systems often fail before the return, while memory systems differ in initial application, temporary restraint, and reapplication. Conditional on correct initial application and temporary restraint, return-stage outcomes vary across configurations and return forms. Stating final applicability explicitly does not provide a uniform gain: it widens the matched control–reapplication gap in the eight configurations included in the detailed return analysis, including systems whose complete trajectory score improves. Controlled information interventions on MemoryOS and A-Mem further show that the material that helps temporary restraint need not be the material that helps reapplication, and that the effect depends on the system.

Our contributions are as follows:

- **Benchmark and corpus.** We construct SIRRA, a purpose-built long-history benchmark and corpus for evaluating the implicit reapplication of a standing instruction after temporary non-applicability.
- **Matched measurement.** We provide matched histories and stage-resolved joint measures of initial application, intermediate restraint, and return outcomes, with controls for instruction access and applicability inference.
- **Empirical study.** We report stage-specific outcome profiles, matched return-form contrasts, and controlled information effects across full-context, retrieval, and persistent-memory configurations.

Benchmark	Longitudinal history	Implicit applicability	Scored non-use	Matched diagnostic contrast	Downstream decision/action	Reapplication
LoCoMo (Maharana et al., 2024)	✓	◐	◐	×	◐	×
LongMemEval (Wu et al., 2025)	✓	◐	◐	◐	✓	×
MemoryAgentBench (Hu et al., 2026)	✓	◐	◐	×	✓	×
ImplicitMemBench (Qin et al., 2026)	◐	✓	◐	✓	✓	◐
LoCoMo-Plus (Li et al., 2026b)	✓	✓	×	◐	✓	×
Mem2ActBench (Shen et al., 2026)	✓	✓	×	✓	✓	×
SIRRA	✓	✓	✓	✓	✓	✓

Table 1: Protocol coverage of selected memory benchmarks. ✓ denotes direct coverage in a scored component, ◐ a concrete partial implementation or protocol mismatch, and × no corresponding test in the inspected protocol. Reapplication requires a still-valid instruction to be applied, withheld when inapplicable, and reapplied without restatement. Marks characterize benchmark design, not system performance.

2 Related Work

Long-term conversational memory benchmarks. Long-term conversational memory benchmarks evaluate how assistants use knowledge accumulated across conversations. Multi-Session Chat and MemoryBank study how information about a user informs responses in later sessions (Xu et al., 2022; Zhong et al., 2024). Question answering provides more targeted tests of memory: LoCoMo includes questions that require connecting events across sessions and reasoning about when they occurred (Maharana et al., 2024), while LongMemEval tests whether systems integrate information across sessions, track changes in user information, and abstain when the history does not support an answer (Wu et al., 2025). MemoryAgentBench presents information incrementally and evaluates how agents retrieve facts, learn new rules, and revise conflicting information as inputs accumulate (Hu et al., 2026). Together, these benchmarks treat retaining, combining, and updating prior information as central to conversational continuity.

Implicit and context-sensitive memory use. Implicit and context-sensitive memory benchmarks examine how prior information should shape responses when the current request leaves its relevance unstated. ImplicitMemBench tests whether routines, thematic influences, and learned avoidance persist through intervening material without reminders at test time (Qin et al., 2026). LoCoMo-Plus evaluates whether responses respect earlier goals or preferences when the original cue and later request have low semantic similarity (Li et al., 2026b). InMind pairs direct questions about a personal fact with indirect queries that require using it, separating access to the fact from its application (Li et al., 2026a). Whether to use a remembered preference also depends on context. RPEval distinguishes cases in which a preference should be ignored, should support an answer, or should dominate it (Feng et al., 2026). BenchPreS evaluates preference application and suppression across different recipients and tasks (Yoon et al., 2026). STALE examines belief revision when new observations implicitly invalidate earlier assumptions, including whether the revised state guides subsequent responses (Chao et al., 2026). Together, these evaluations distinguish access to information from its appropriate use, which can require drawing an unstated connection, revising an earlier belief, or leaving a valid preference unused.

Memory-grounded decisions and actions. Memory-grounded decision and action benchmarks evaluate which action past information calls for and when that action is appropriate. Mem2ActBench uses underspecified requests that require selecting tools and their arguments based on interaction history (Shen et al., 2026). Beyond Prompts tests responses to trigger phrases and instructions to act on a designated future response, interleaving these tasks with other conversation (Castillo-Bolado et al., 2024). PM-Bench scores which deferred actions are due during a simulated week, including penalties for premature or non-due selections (Liu & Gabriel, 2026). TriggerBench pairs action and restraint cases with direct memory questions (Zhang et al., 2026).

Taken together, these benchmarks cover long-horizon memory, context-sensitive use, and memory-grounded action. SIRRA focuses on the temporal relation that links these capabilities: one standing instruction is applied, withheld, and reapplied as current evidence changes.

3 BENCHMARK

3.1 TASK DEFINITION

Each SIRRA task instance pairs a conversation history with a query asking whether to take an action specified in an earlier user instruction. The user establishes the condition for that action and the instruction's scope, including the activity and period it covers. Later exchanges provide reports or updates about the activity, interspersed with unrelated conversation. For each designated observation, the history ends after the current evidence and before the assistant's response to it; the shared query asks whether to take the specified action without repeating its condition.

For each designated observation, *current evidence* is the most recent completed report or update available within the instruction's scope. *Current applicability* is the result of applying the earlier condition to the reported facts. It can change across observations while the instruction remains valid. The correct decision is to take the specified action when the condition holds and to withhold it otherwise.

In the core condition, current messages provide sufficient evidence to determine whether the earlier instruction applies, without restating the instruction or supplying that judgment.

Each *benchmark family* contains two matched histories for the same standing instruction. Each history has initial and middle observations followed by a final *return* observation. The histories share the instruction, query, and initial and return observations. In the *always-applicable control*, the condition holds at all three observations. In the *reapplication path*, it holds at the initial and return observations but not at the middle observation. The history prefix ending at each observation and the shared query form a task instance.

Each family also includes two versions of the final observation. The *evidence return* presents it under the core information condition. The *classification return* is an explicit-applicability control: it states whether the condition holds in the same final situation without repeating the standing instruction.

3.2 BENCHMARK CONSTRUCTION

We construct each family from a user profile, first developing a shared work context and semantic plan, then expanding the plan into matched event sequences and long conversational histories (Figure 2).

Stage 1: Scenario and Instruction Construction. We prompt an LLM with selected PersonaMem profiles (Jiang et al., 2025) to construct a shared work context in which a user's instruction can govern later decisions. The model selects relevant profile details to establish the participants, their working relationship, a shared work object, and an activity with a concrete observation cadence, such as a manager reviewing periodic quality reports. Within this context, it formulates an instruction linking an observable condition to a specific action. Observations in the activity must be able to satisfy or fail to satisfy the condition, allowing the appropriate decision to change while the instruction remains in force.

Stage 2: Matched Event Generation. We expand the semantic plan into paired event sequences that differ in when the instruction applies. A shared timeline specifies when the instruction is established and schedules three observations to match the activity. We prompt an LLM to construct shared initial and return observations that satisfy the condition, together with two alternative middle observations, one that satisfies it and one that does not. In the running example, the paired September reports fall on opposite sides of the defect-rate threshold, while the August and October reports are shared.

The model specifies each observation as a report or update with concrete values or propositions that support the intended decision under the earlier instruction. When several facts are needed, it identifies their subjects and relations so that they can be combined. It generates paired evidence-return and classification-return versions of the final observation.

Stage 3: Dialogue Generation and Benchmark Assembly. We prompt an LLM to render the event sequences as conversations while preserving their factual content and temporal order. The model organizes each event's facts into conversational exchanges, writes user messages that convey

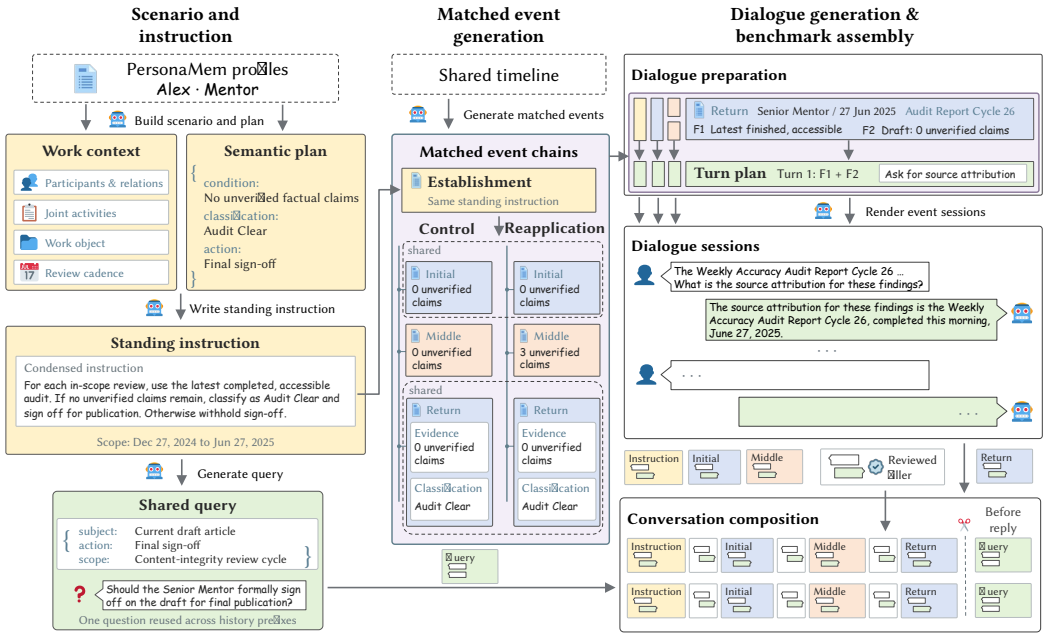

Figure 2: SIRRA construction. Representative excerpts are shown; ellipses omit dialogue, and scissors mark the pre-reply cutoff.

those facts, and adds assistant replies that address the immediate update or request. Shared events are rendered once and reused across the paired histories, keeping their wording consistent between paths. For long-context inputs, the assembly step inserts complete, unrelated conversations between the sessions containing these events.

For each history prefix, the query asks whether to take the specified action. A separate generation step produces one natural-language question per family using the decision subject, action, and scope, without access to the current condition value or reference decision. After review, the question is reused across all history prefixes in the family.

3.3 Evaluation and Metrics

The evaluation prompt requires a single uppercase token: `APPLY` to indicate that the specified action should be taken, or `SUPPRESS` to indicate that it should be withheld. We score each response by exact match against the corresponding reference label; other output formats count as incorrect.

We summarize each family's decisions with four joint indicators. Pre-return competence (K) requires correct application after the initial observation and correct restraint after the middle observation on the reapplication path. Qualified failure (QF) occurs when those two decisions and the control return are correct, and the reapplication return is incorrect. Qualified reversal (QR) captures the opposite return pattern: the initial and middle decisions and the reapplication return are correct, and the control return is incorrect. Reapplication trajectory success (RTS) requires correct decisions following all three observations on the reapplication path.

For a fixed system, run, and trial, let $C_{i,s}(p,t) \in \{0,1\}$ indicate whether the response following observation t is correct for family i, return condition s, and path p. Here, p_{ctrl} and p_{reapp} denote the always-applicable control and reapplication path, respectively; $t = 0,1,2$ indexes the initial,

Method	Backbone	K	Evidence return			Classification return		
			QF	QR	RTS	QF	QR	RTS
Full context	GPT-5.4	81.4%	1.9%	0.0%	79.5%	16.1%	0.8%	63.9%
Full context	Gemini 3.1 Pro	62.2%	5.6%	3.3%	54.5%	24.0%	5.2%	30.5%
Naive RAG–BM25	Qwen3.6-27B	18.6%	5.6%	0.4%	10.2%	6.5%	0.8%	8.8%
Dense RAG–Semantic	Qwen3.6-27B	21.9%	5.0%	0.4%	13.6%	6.5%	1.5%	13.4%
Mem0 (Chhikara et al., 2025)	Qwen3.6-27B	24.4%	9.4%	1.0%	10.2%	10.6%	1.9%	11.7%
A-Mem (Xu et al., 2025)	Qwen3.6-27B	28.8%	7.3%	0.4%	20.9%	10.6%	0.8%	18.2%
LightRAG (Guo et al., 2024)	Qwen3.6-27B	20.7%	5.2%	0.8%	14.4%	3.8%	0.0%	16.5%
SimpleMem (Liu et al., 2026)	Qwen3.6-27B	45.1%	5.2%	1.9%	38.8%	12.3%	1.3%	29.4%
MemoryOS (Kang et al., 2025)	Qwen3.6-27B	58.5%	4.2%	2.7%	53.4%	10.0%	4.6%	45.9%

Table 2: Descriptive baseline outcome profiles (%). The initial and middle observations are shared across return forms, so K appears once. QF and QR are one-sided return outcomes among families satisfying K; RTS is the complete reapplication trajectory. Each is reported for the indicated return form.

middle, and return observations. The indicators are

$$\begin{aligned} K_{i,s} &= C_{i,s}(p_{\text{reapp}}, 0)\, C_{i,s}(p_{\text{reapp}}, 1), \\ QF_{i,s} &= K_{i,s}\, C_{i,s}(p_{\text{ctrl}}, 2)[1 - C_{i,s}(p_{\text{reapp}}, 2)], \\ QR_{i,s} &= K_{i,s}\, [1 - C_{i,s}(p_{\text{ctrl}}, 2)] C_{i,s}(p_{\text{reapp}}, 2), \\ RTS_{i,s} &= K_{i,s}\, C_{i,s}(p_{\text{reapp}}, 2). \end{aligned} \tag{1}$$

For each return condition s, we average each indicator over the same set of N_s benchmark families, including families with $K_{i,s} = 0$:

$$\widehat{\theta}_X(s) = \frac{1}{N_s} \sum_{i=1}^{N_s} X_{i,s}, \qquad X \in \{K, \text{QF}, \text{QR}, \text{RTS}\}. \tag{2}$$

We also report accuracy after each observation and complete-trajectory accuracy.

The return-path gap (RPG) is control return accuracy minus reapplication return accuracy on one return condition. Paired surface contrast (PSC) is evidence-return RPG minus classification-return RPG: positive values indicate that classification return narrows the gap, and negative values that it widens the gap. For the information-supplementation diagnostics in Section 4.3, paired reapplication gain (PRG) is the change in reapplication-return accuracy under a diagnostic condition relative to a newly generated, unmodified replay for the same families. These differences are reported in percentage points.

4 Results

4.1 Stage-Specific Outcome Profiles

Table 2 reports outcomes over 479 families. Under evidence return, pre-return competence (K) ranges from 18.6% for Naive RAG to 81.4% for GPT-5.4, while complete reapplication trajectory success (RTS) ranges from 10.2% for Mem0 to 79.5% for GPT-5.4.

Similar return accuracies conceal different stage bottlenecks. MemoryOS and Gemini 3.1 Pro both reach 86.6% evidence-return accuracy and similar RTS (53.4% and 54.5%), but MemoryOS is weaker at restraint and Gemini at initial application. SimpleMem reaches 87.9% return accuracy but only 50.1% restraint accuracy, yielding 38.8% RTS. A-Mem and LightRAG also lose more trajectories at restraint, whereas Naive RAG is weakest at initial application.

4.2 Matched Return Contrasts after Temporary Non-Use

Both return forms require the same final action, but classification return states current applicability rather than requiring it to be inferred from evidence. It widens the control–reapplication gap in all

Method	Backbone	Evidence RPG	Classification RPG	PSC
MemoryOS	Qwen3.6-27B	2.5	10.0	−7.5
A-Mem	Qwen3.6-27B	25.1	35.3	−10.2
LightRAG	Qwen3.6-27B	12.5	15.2	−2.7
Mem0	Qwen3.6-27B	18.0	19.4	−1.5
Full context	GPT-5.4	2.3	19.4	−17.1
Full context	Gemini 3.1 Pro	3.1	28.6	−25.5
Naive RAG–BM25	Qwen3.6-27B	11.1	12.5	−1.5
Dense RAG–Semantic	Qwen3.6-27B	9.4	10.9	−1.5

Table 3: Matched return contrasts (percentage points). RPG is control return accuracy minus reapplication return accuracy; PSC is evidence-return RPG minus classification-return RPG. Negative PSC indicates a larger gap under classification return. All values use 479 families per configuration; differences are computed before rounding.

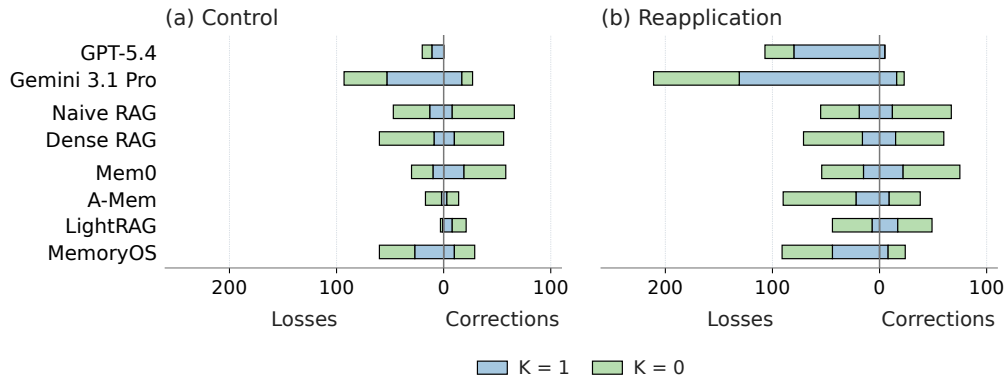

Figure 3: Evidence-to-classification return transitions for the control and reapplication paths. Panels (a) and (b) show correct-to-incorrect losses and incorrect-to-correct corrections on the control and reapplication paths, respectively, stratified by earlier correctness ($K = 1$ or $K = 0$), over 479 families per configuration.

eight configurations analyzed here, including systems whose RTS improves. Table 3 reports RPG and PSC over all 479 families. Every PSC is negative; MemoryOS's RPG rises from 2.5 to 10.0 points and A-Mem's from 25.1 to 35.3.

Classification return converts 35 previously complete MemoryOS trajectories and 22 A-Mem trajectories into QF: the control remains correct while reapplication becomes incorrect. LightRAG and Mem0 gain ten and seven complete trajectories, respectively, but gain fewer correct reapplication returns than control returns (5 versus 18 for LightRAG; 21 versus 28 for Mem0).

4.3 Controlled Diagnostics

Supplying both the earlier instruction and current facts yields the highest middle-stage restraint accuracy in both memory systems, whereas each achieves its highest reapplication-return accuracy with a single excerpt. We evaluate MemoryOS and A-Mem, both using Qwen3.6-27B, on the 479 families. At the reapplication middle and both evidence returns, we hold each system's retrieved context fixed and generate four responses: an unmodified replay, or a response with the earlier instruction, current conversation excerpt, or both inserted before the query. The excerpts reproduce the original dialogue verbatim. All changes below use the new replay as their reference (Figure 4).

A-Mem's largest reapplication gain comes from the current facts. Supplying them raises return accuracy from 60.3% to 74.9% (PRG $+14.6$ percentage points), while control-return accuracy changes from 75.2% to 75.8%. The gap between the two returns therefore narrows from 14.8 to 0.8 points. Middle restraint also rises from 25.5% to 63.0%. Restating the instruction alone produces a differ-

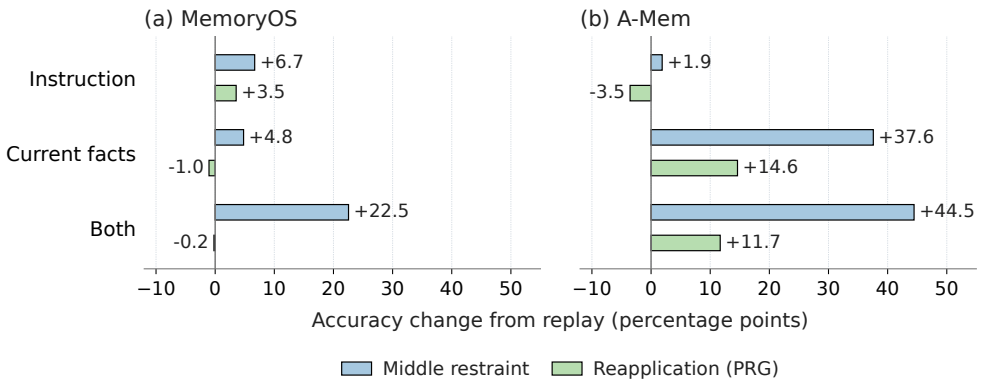

Figure 4: Changes in middle-restraint accuracy and paired reapplication gain (PRG) under information supplementation, relative to the unmodified replay.

ent profile: middle accuracy is 27.3%, and reapplication-return accuracy falls to 56.8% (PRG -3.5 points).

MemoryOS achieves its highest reapplication-return accuracy with the instruction alone: 77.7%, up from 74.1% under replay. Current facts alone yield 73.1%, and the joint condition yields 73.9%. Yet the joint condition produces the largest gain in middle restraint, from 53.2% to 75.8% ($+22.5$ points). A-Mem shows the same separation between the best conditions for the two decisions: adding the instruction alongside current facts raises middle accuracy further to 69.9%, while return accuracy is 72.0%, below the 74.9% achieved with facts alone. Across these conditions, the information combination that best supports withholding the action differs from the one that best supports reapplying it.

With only the earlier instruction, current excerpt, and query, the shared short-context Qwen3.6-27B reference answers 425/479 middle decisions correctly (88.7%) and 390/479 return decisions correctly (81.4%). Table 4 reports QF/QR and the repair and loss counts for families grouped by their original outcomes. Appendix D.1 gives the intervention and scoring details.

System	Supplement	QF (%)	QR (%)	Original QF repaired	Previously correct returns lost
MemoryOS	Replay	10.0	10.0	7/20	83/415
	Instruction	9.8	8.4	14/20	87/415
	Current facts	11.5	8.4	11/20	101/415
	Both	9.8	9.0	10/20	99/415
A-Mem	Replay	7.7	3.3	5/35	79/340
	Instruction	10.0	1.3	9/35	104/340
	Current facts	2.9	3.1	21/35	78/340
	Both	3.8	4.4	18/35	90/340

Table 4: Joint return outcomes under information supplementation. QF and QR use the original main-experiment K and all 479 families. Repair requires both new returns to be correct among original evidence-return QF cases. Lost returns are newly incorrect reapplication responses among originally correct evidence reapplication returns. Counts show their subgroup denominators; replay and all supplemented conditions use newly generated answers.

5 Discussion

5.1 Interpreting outcome profiles

The benchmark measures a trajectory rather than an isolated endpoint. K records demonstrated initial application and temporary restraint; QF and QR then identify the direction of matched return

disagreement, while RTS records the complete apply–suppress–reapply path. A return score alone therefore cannot identify where a system lost the trajectory. The classification return is a control surface for the information condition, not an easier replacement for the evidence return.

The supplementation conditions provide behavioral localization only. Inserting verbatim instruction or current-fact material into a fixed retrieved context tests which available evidence changes a decision, but cannot identify failure during storage, retrieval, selection, or decision formation. It also does not validate the reference labels independently or reveal an internal memory representation.

The evaluation implication is to distinguish an instruction's continuing validity from its current applicability and to score appropriate use, non-use, and renewed use together. These implications remain bounded by the frozen histories and answer contract; they do not establish production behavior or cross-system superiority.

5.2 Limitations

SIRRA uses selected PersonaMem profiles (Jiang et al., 2025) and LLM-generated conversations. Profile selection and generation shape the range of scenarios and language represented; generation artifacts are a source of variation in target recoverability and naturalness. The manual corpus review removed six families with inconsistent wording or a trajectory mismatch. Retained evidence varies in explicitness, including threshold comparisons and classification cues; independent multi-annotator agreement and quantitative shortcut ablations remain unmeasured. Validation on independently collected conversations would broaden the empirical basis.

Each family contains one instruction that remains valid through a single interval of non-applicability. Extensions include instruction revision and revocation, interacting instructions, and repeated changes in applicability.

Evaluation uses frozen histories and binary action decisions. Live interactions would test how a system's own responses shape later decisions, and tool execution would test whether selected actions are completed. The action labels are imbalanced across observations; additional non-application cases would broaden the assessment of restraint.

The baseline comparison covers selected systems and configurations, with different backbones across full-context and memory-system evaluations. Comparisons with a shared backbone and additional retrieval and memory settings would help isolate the contribution of memory-system design.

We report fixed-denominator descriptive point estimates without family-level uncertainty intervals or significance tests; matched cells within a family are not independent draws.

6 Conclusion

SIRRA measures whether assistants reapply a still-valid standing instruction after temporary non-applicability. Across 479 families and nine configurations, correct initial application and restraint do not guarantee reapplication, and explicitly stating final applicability does not uniformly improve the trajectory. Information supplementation also has system- and stage-dependent effects. Standing-instruction evaluation should therefore score appropriate use, non-use, and renewed use together.

AI use statement

We used Qwen3.5-397B-A17B to generate synthetic benchmark data, with construction prompts and generation settings documented in Appendices A.3 and A.4. AI tools also assisted with code development and manuscript drafting and language polishing. AI-assisted code was checked through software tests. The authors take responsibility for the final content, code, and data.

References

David Castillo-Bolado, Joseph Davidson, Finlay Gray, and Marek Rosa. Beyond prompts: Dynamic conversational benchmarking of large language models. In *Advances in Neural Information Processing Systems 37*, pp. 42528–42565. Neural Information Processing Systems Foundation,

Inc. (NeurIPS), 2024. doi: 10.52202/079017-1347. URL https://doi.org/10.52202/079017-1347.

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. STALE: Can LLM agents know when their memories are no longer valid? *arXiv preprint arXiv:2605.06527*, 2026. URL https://arxiv.org/abs/2605.06527v1.

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.

Xueyang Feng, Weinan Gan, Xu Chen, Quanyu Dai, and Yong Liu. How does personalized memory shape LLM behavior? benchmarking rational preference utilization in personalized assistants. *arXiv preprint arXiv:2601.16621*, 2026. URL https://arxiv.org/abs/2601.16621v1.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024. URL https://arxiv.org/abs/2410.05779.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DT7JyQC3MR.

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. *arXiv preprint arXiv:2504.14225*, 2025. URL https://arxiv.org/abs/2504.14225.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Ruizhe Li, Mingxuan Du, Benfeng Xu, and Zhendong Mao. Keep it InMind: Benchmarking the implicit-association blind spot in agent memory. *arXiv preprint arXiv:2607.24368*, 2026a. URL https://arxiv.org/abs/2607.24368v1.

Yifei Li, Weidong Guo, Lingling Zhang, Rongman Xu, Muye Huang, Hui Liu, Lijiao Xu, Yu Xu, and Jun Liu. LoCoMo-Plus: Beyond-factual cognitive memory evaluation framework for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25085–25100. Association for Computational Linguistics, July 2026b. doi: 10.18653/v1/2026.acl-long.1150. URL https://aclanthology.org/2026.acl-long.1150/.

Genglin Liu and Saadia Gabriel. PM-Bench: Evaluating prospective memory in LLM agents. *arXiv preprint arXiv:2607.12385*, 2026. doi: 10.48550/arXiv.2607.12385. URL https://arxiv.org/abs/2607.12385v1.

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

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

Chonghan Qin, Xiachong Feng, Weitao Ma, Xiaocheng Feng, and Lingpeng Kong. ImplicitMemBench: Measuring unconscious behavioral adaptation in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28232–28261. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1301. URL https://aclanthology.org/2026.acl-long.1301.

Yiting Shen, Kun Li, Wei Zhou, and Songlin Hu. Mem2ActBench: A benchmark for evaluating long-term memory utilization in task-oriented autonomous agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8173–8190. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.acl-long.370. URL https://aclanthology.org/2026.acl-long.370/.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10813.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5180–5197. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.356. URL https://aclanthology.org/2022.acl-long.356/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems 38*, 2025. doi: 10.52202/085713-0593. URL https://papers.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Sangyeon Yoon, Sunkyoung Kim, Hyesoo Hong, Wonje Jeung, Yongil Kim, Wooseok Seo, Heuiyeen Yeen, and Albert No. BenchPreS: A benchmark for context-aware personalized preference selectivity of persistent-memory LLMs. *arXiv preprint arXiv:2603.16557*, 2026. URL https://arxiv.org/abs/2603.16557v1.

Tianhua Zhang, Xinjiang Wang, Qianxi Zhang, Qi Chen, Kun Li, Yaoqi Chen, Dingdong Wang, Helen Meng, and Yan Lu. Triggerbench: Investigating prospective memory for large language models. *arXiv preprint arXiv:2606.23459*, 2026. URL https://arxiv.org/abs/2606.23459.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

A Benchmark Construction Details

A.1 From Source Profiles to Standing Instructions

PersonaMem profiles (Jiang et al., 2025) supply the background for scenario construction. The source input is a complete selected profile. Its statements are used to ground a recurring activity, such as reviewing a shared report; PersonaMem questions, answers, and conversation histories are not imported into the task. The working relationship, decision, instruction, observations, and dialogue are authored for SIRRA. The profile itself is not inserted into the evaluated conversation.

The setting and decision are constructed in one generation call. The model first selects supporting profile phrases and establishes the participants, shared work object, recurring record, and review cadence. It then specifies the decision subject, the governed action and its recipient, the observation used for the decision, and the condition under which the action is warranted. The condition must admit both satisfying and non-satisfying observations while leaving the instruction in force. When a condition requires a comparison or combination of facts, its subjects, units, reference values, and combination rule are specified explicitly. A short name for the satisfying classification provides the wording used in the classification-return control.

Review examines whether the selected profile details support the setting and whether the decision is coherent within it. Once this setting and decision are fixed, a timeline assigns the instruction's establishment time, its working period, and the later review times. The instruction author expresses the standing instruction as a participant's communication and describes how it is received or recorded before the reviews begin. This gives the conversation an earlier user instruction that can govern each subsequent decision.

A.2 Observation and Dialogue Generation

Each observation author receives the work setting, the decision condition, the report-selection rule, and an assigned review time. The author is asked to identify the most recent completed report, review, or measurement available at that time and provide its concrete findings. For quantitative conditions, the requested facts specify the subjects, units, windows, and comparisons needed to interpret them. The generated record separates these observation facts from situational information about the participants, report source, and review time.

The reapplication history uses an initial observation satisfying the condition, a later observation that does not satisfy it, and a return observation that satisfies it again. An alternative middle observation keeps the condition satisfied for the always-applicable control. The two middle observations are authored separately from the same initial observation and assigned review time. The continuing role additionally requests substantive new evidence beyond the earlier observation. At the return, the author produces both a factual account for the evidence return and an account naming the satisfying classification for the classification return. The instruction, initial observation, and each return form are shared between the two histories.

Dialogue planning assigns the ordered facts to user turns and may add a local request, such as recording a finding or restating a comparison. The planner starts with one user message. It splits an event only when an intermediate assistant reply serves an ordinary purpose and a later user message first supplies a fact needed for the current decision. The user-message writer receives the assigned facts and request as its required message content; source, time, and interaction context are supplied separately to help resolve references and attribution. The assistant writer then answers the local request or acknowledges the update. These generated messages form the conversation used in evaluation; the situational fields are not appended as separate evidence. Each event is rendered once and reused wherever it occurs in the family's histories.

A.3 Construction Prompts

The following listings reproduce the system instructions and user templates used to generate the setting and decision, the instruction and observations, the local turn plans, and the conversation messages. They follow the order described above. At each call, the instance placeholder is filled with that role's inputs from the preceding stages.

Generation settings. The retained construction records contain the provider configuration for each call. Semantic authors, turn planners, and dialogue writers use `Qwen3.5-397B-A17B` with temperature 0.2 and an 8,192-token output limit. The request includes no top-p or top-k field. Each call allows one attempt and uses zero transport retries. The retained production batches use sampling seeds 60018, 60832, or 60834; the seed is a batch-level reproducibility parameter, not a model or code identifier. The same model and request parameters are used by the separate family-query writer described in Section A.4. Transport-only controls are not part of the task prompts below.

`INSTANCE_INPUT` indicates where the role-specific input record is inserted. The initial, continuing, and temporary-absence authors use one observation template, with the corresponding task text substituted for `CURRENT_TASK`. These task substitutions are reproduced immediately after the shared template.

Prompt A.1: Setting and decision

```
SYSTEM
You author one source-grounded recurring work setting and the complete decision relation
     inside it. First close the setting, then close the decision, and return only the
     requested records.

USER TEMPLATE
# Recurring setting and decision definition

## Fixed source input

{{INSTANCE_INPUT}}
```

```
The input is one complete persona profile. Exact quotations identify source facts;
the remaining records are ordinary authored details supported by those facts.

## Task

Create one source-grounded recurring work setting and one complete decision rule
that naturally belongs to that setting.

## Construction workflow

First close the setting:

1. Select a small set of short exact profile phrases that materially support one
   kind of recurring work.
2. From those anchors, create one plausible working relationship and one locally
   identified shared work object.
3. Create one recurring activity that states who performs it, which shared report,
   review, or measurement they inspect, and its ordinary cadence.
4. Keep the participants, shared work object, recurring record, and cadence stable
   before choosing a consequence.

Then, using exactly that setting, close the decision:

1. Choose one recurring work item and one realistic action that the participants
   may or may not perform for that item after reading the newest usable report.
   Frame the decision as whether that one action occurs for the current item.
2. State the same single action and exactly what or who it affects. Keep inseparable
   operational parts together, but do not add a fallback or alternative result.
3. Name the recurring report, review, or measurement the participants consult and
   what it records about the ongoing work. Use the family name, not a dated
   occurrence.
4. State one stable condition that a colleague can evaluate from one occurrence of
   that record alone. Include the measured subject, comparison or logical relation,
   reference, units, scope, and combination rule whenever they are needed. Make
   both satisfying and non-satisfying records possible.
5. Give the satisfying result a short name that fits the workplace record.
6. Write one ordinary sentence connecting that classification, the current review
   decision, and the chosen action for its affected item or recipient.
7. Check that the participants, shared work object, recurring record, decision
   subject, and affected item keep the same ordinary roles across both parts.

Do not choose event times, an instruction scope, checkpoints, factors, queries,
targets, or dialogue. Do not expose or repeat the profile as a downstream field.

## Response

Use the exact physical record form `LABEL | value`, one physical line per record.
Return these records in order:

ANCHOR | exact profile phrase
PERSONA RELEVANCE | value
DOMAIN | value
RELATIONSHIP CONTEXT | value
RECURRING ACTIVITY | value
DECISION SUBJECT | value
GOVERNED ACTION | value
TRIGGER SUBJECT | value
TRIGGER CONDITION | value
CONDITION-SATISFYING CLASSIFICATION | value
TRIGGER TO DECISION LINK | one complete sentence

Return one or more `ANCHOR` records and exactly one of every other record.
```

Prompt A.2: Instruction establishment

```
SYSTEM
You author public work records. Create one ordinary interaction that establishes the
     supplied decision rule before its working period begins and return only the
     requested records.

USER TEMPLATE
# Public instruction establishment

## Fixed setting and decision inputs

{{INSTANCE_INPUT}}
```

```
## Task

Create one ordinary public interaction in which one credible participant
establishes the supplied standing decision rule for the supplied working period.
Complete the interaction before that period and its first controlling observation.

## Construction workflow

1. Treat the supplied public establishment time as the time of one public
   interaction. From the working relationship, choose the credible stable speaker
   and the ordinary participants or shared channel for that interaction.
2. Write one canonical conditional instruction for the supplied scope. If the
   observation selected by the supplied rule meets the complete public condition,
   assign the supplied satisfying classification and APPLY the named governed
   action for that review. If it does not meet the condition, withhold that same
   action for that review and end the false branch there. The false branch does not
   start, recommend, default to, or authorize different work. Preserve the supplied
   observation-to-decision-to-action relationship, qualifiers, threshold, scope,
   and logical relation.
3. Write ordinary establishment facts describing where the speaker communicates
   the instruction and who receives or records it. Complete the interaction with
   a readiness fact showing that the shared instruction channel or decision log is
   in place to carry that instruction through the supplied period. Each later
   review applies the published rule and names its selected completed observation.
   End the establishment event after those facts.

## Response

Use the exact physical record form `LABEL | value`, one physical line per record.
Return these records in order:

STABLE USER SPEAKER | value
INSTRUCTION SCOPE | exact supplied instruction scope
PUBLIC ESTABLISHMENT TIME | exact supplied public establishment time
INSTRUCTION WORDING | value
PUBLIC INTERACTION CONTEXT | value
PUBLIC EVIDENCE SOURCE | value (optional; omit when no specific public source applies)
ESTABLISHMENT FACT | value

Return one or more `ESTABLISHMENT FACT` records.
```

Prompt A.3: Observation update (initial, continuing, or temporary absence)

```
SYSTEM
You author public observation records. Create one current observation with the requested
     source facts and relations, and return only the requested records.

USER TEMPLATE
# Current public observation

## Information for this review

{{INSTANCE_INPUT}}

The work setting and recurring review establish where this record belongs. The
recurring observation and selection rule identify the source to use. Use the
condition to choose concrete report facts on the side requested below. The
speaker, working period, and assigned time situate the record.

## Task

{{CURRENT_TASK}}

## Construction workflow

1. Treat the assigned public observation time as the time of the current review.
   Use the supplied selection rule to choose the most recent finished report,
   review, or measurement that the participants can access. Keep the review time
   and selected source occurrence distinct when the rule selects an earlier
   completed occurrence.
2. Name what the selected source records inside the stated work setting. Use the
   setting and condition to identify the concrete people, items, and scopes the
   source reports on. Give the occurrence one ordinary public context and, when
   applicable, one public report, artifact, log, or channel identity as its
   evidence source.
```

```
3. Build the selected occurrence as an ordinary source record. First identify the
   source and, when needed, why it is the most recent finished and available one.
   Then choose concrete values or propositions that fall on the requested side of
   the supplied condition. Keep each reported subject, scope, period, unit, scale,
   threshold, reference, denominator, and qualifier that gives those facts their
   meaning.
4. Include enough of the selected source's facts to support that side. When the
   condition combines several items, include the necessary items and their stated
   relation; when it applies to a collection, identify the relevant member or
   collection evidence.
5. Finish with the ordinary factual comparison or relation carried by the source:
   for example, a reported value relative to its reference, a recorded property of
   an item, or the stated relation among several reported components. End the
   record at that fact-level relation.

The completed observation is the source record and its evidence relation. The
condition guided which facts you chose; it is not itself a status to announce in
the record. A classification, review outcome, operational consequence,
recommendation, or next step would be a different record.

The assigned public observation time supplies the review event's date, clock, and
timezone. Use that exact value whenever the review time is stated.

## Response

Use the exact physical record form `LABEL | value`, one physical line per record.
Return these records in order:

PUBLIC INTERACTION CONTEXT | value
PUBLIC OBSERVATION TIME | exact assigned public observation time
PUBLIC EVIDENCE SOURCE | value (optional; omit when no specific public source applies)
OBSERVATION FACT | value

Return one or more `OBSERVATION FACT` records.

Role-specific task substitutions used in the retained calls:
INITIAL:
Use the supplied condition as a construction guide. Author one current public
     observation whose concrete facts and factual relations fall on its satisfying side
     at the assigned public observation time.
CONTINUING:
Use the supplied condition as a construction guide. Author one later independent public
     observation whose concrete facts and factual relations fall on its satisfying side.
     Include at least one substantive current attribute or evidence item beyond the
     earlier observation; the assigned date alone is not that new evidence.
TEMPORARY ABSENCE:
Use the supplied condition as a construction guide. Author one alternative later public
     observation whose coherent concrete facts and factual relations fall on its
     non-satisfying side at the assigned public observation time.
```

Prompt A.4: Matched return observation

```
SYSTEM
You author public observation records. Create one later current observation and its
     matched direct and evidence expressions, and return only the requested records.

USER TEMPLATE
# Matched return observation

## Fixed setting and decision inputs

{{INSTANCE_INPUT}}

## Task

Create one later current public observation whose reported facts fall on the
satisfying side of the supplied condition. Author one direct expression and one
evidence expression of that same observation.

## Construction workflow

1. Treat the assigned public observation time as the time of the return review.
   Apply the supplied selection rule at that review inside the instruction scope:
   consider the scheduled occurrences of the recurring observation whose work has
   finished and whose result is available, then choose the most recent eligible
   occurrence. Use the supplied condition to choose concrete subjects and reported
```

```
   facts on its satisfying side.
2. Give the selected occurrence one public context. When applicable, record one
   public report, artifact, log, or channel identity as its evidence source. Keep
   the review time and the selected source occurrence distinct when the rule
   selects an earlier completed occurrence.
3. Write one `DIRECT FACT` that identifies the observation through its context or
   source and states its supplied satisfying classification as the current result.
   Include completion or availability detail only when it is needed to make clear
   why this occurrence is the one used at the review.
4. Use the first `EVIDENCE FACT` to name the same selected source and include any
   ordinary completion or availability detail needed to make that choice clear.
   Phrase this as an ordinary public record. Write the remaining `EVIDENCE FACT`
   records as source evidence. Choose concrete values or propositions on the
   satisfying side and preserve the subject, scope, period, unit, scale, threshold,
   reference, denominator, and qualifier that give them meaning. When several
   items or a collection determine that side, include the necessary items and
   their stated relation.
5. After the source-identifying fact, use the condition-evidence records for the
   factual comparisons or relations carried by the selected source. End the last
   condition-evidence record with the final fact-level comparison or relation.
   That relation completes the evidence expression; do not add a separate
   condition-status sentence.
   The `DIRECT FACT` separately carries the supplied satisfying classification as
   the current result. Keep both expressions aligned on each named subject and
   relation, the selected instance, scope, time, quantities, scale, context, and
   source. End both expressions before any current decision, authorization,
   governed-action consequence, recommendation, or next-step work.

The assigned public observation time supplies the return review's date, clock, and
timezone. Use that exact value whenever the review time is stated.

## Response

Use the exact physical record form `LABEL | value`, one physical line per record.
Return these records in order:

SHARED PUBLIC CONTEXT | value
PUBLIC OBSERVATION TIME | exact assigned public observation time
PUBLIC EVIDENCE SOURCE | value (optional; omit when no specific public source applies)
DIRECT FACT | value
EVIDENCE FACT | value

Return exactly one `DIRECT FACT` and at least two `EVIDENCE FACT` records. The
first `EVIDENCE FACT` names the selected source; every later `EVIDENCE FACT` is
condition evidence, with the last one completing the fact-level relation.
```

Prompt A.5: Turn plan

```
SYSTEM
You plan ordinary user-to-assistant work exchanges. Arrange every supplied fact into the
     requested message plan and return only the requested records.

USER TEMPLATE
# Local turn plan

## Fixed local planning inputs

{{INSTANCE_INPUT}}

The listed local facts supply the complete work-record meaning for this exchange.
Public context and event time supply its situation, the source supplies attribution,
and the stable speaker identifies who communicates it.

## Task

Plan one natural user-to-assistant exchange. Begin with one user message containing
every listed fact in order.

## Construction workflow

1. Put every listed fact into one user message in the listed order.
2. Choose the grouping that fits what the facts communicate:
   - **Rule and preparation:** when the facts establish a rule, describe its
     publication, recording, or receipt, or prepare a later review, check, or
     measurement, keep the one message.
   - **Current report:** when the facts report findings from a current review,
```

```
     check, or measurement, consider a split only when both conditions hold: an
     ordinary useful assistant reply belongs before the remaining facts, and the
     final user message first supplies at least one fact necessary to determine the
     reported relation. When both hold, redistribute the facts across user messages
     while preserving their order. The one-message grouping remains in every other
     case.
3. Treat communicating the attached facts as a complete exchange.
4. If a request might materially improve the exchange, first draft its shortest
   adequate answer using only the facts, context, time, and source already attached
   to that same turn.
5. Keep the request only when that drafted answer performs one concrete record
   operation: record or restate a supplied fact, reproduce supplied arithmetic or
   a supplied comparison, organize supplied values, or name supplied attribution.
   Phrase the request so that this drafted record operation is its complete answer.
   When no such answer can be drafted, omit the request.
6. Remember that every planned user message receives an assistant reply. For a
   split exchange, the reply to one message occurs before the next user message,
   so keep each request local to the facts already attached to that turn.

A retained request asks for exactly the drafted record operation. It does not ask
the assistant to create a category, status, applicability judgment, decision,
consequence, recommendation, or next step that is not itself a supplied fact.

## Response

Use the exact physical record form `LABEL | value`, one physical line per record.
For **Rule and preparation**, and for a **Current report** whose facts remain in one
message, return exactly one turn block in this form:

TURN | one user message in this local exchange
LOCAL FACT | local-fact-1
LOCAL REQUEST | optional explicit assistant-facing question or request

For each listed `Local fact: alias`, copy only `alias` after the separator. For
example, `Local fact: local-fact-1` becomes `LOCAL FACT | local-fact-1`.
Repeat `LOCAL FACT` inside that block until it contains every listed fact exactly
once in the listed order.

For a **Current report** whose facts were redistributed because both conversational
conditions held, return one such block for each resulting user message, in message
order. Across those blocks, use every listed fact exactly once and preserve the
listed order.
```

Prompt A.6: Initial user message

```
SYSTEM
You are the stable user expressing the supplied current facts and optional request as
     one ordinary work message. Those inputs already completely determine what the
     message means; choose only wording and organization. Situational details and
     earlier history may clarify an already supplied referent but do not add a
     participant, owner, action, or relation. Return only the user message.

USER TEMPLATE
# Initial user message

## Examples of natural delivery

The examples use unrelated settings. Treat them only as demonstrations of how to
turn the supplied content into an ordinary message. Do not carry their people,
objects, facts, or wording into the current message.

### Example: an update with different subjects

Input:
- Current facts: Priya uploaded the revised seating chart this morning. The venue
  portal now shows it as the active version for Saturday's event.
- Request: none.

Output:
Priya uploaded the revised seating chart this morning, and the venue portal now
shows it as the active version for Saturday.

## Fixed message inputs

{{INSTANCE_INPUT}}
```

```
The current facts and optional request are the complete message meaning. Speaker,
context, time, source, and transcript identify the exchange and already mentioned
referents. They contribute no participant ownership, act, relation, or record
content.

## Task

Write the stable user's first natural message for this planned local exchange.

## Construction workflow

1. Form a natural clause for each current fact. Keep every action or relation with
   the same grammatical subject. Preserve a passive or impersonal relation when
   the fact supplies no actor. I or you may replace a named participant only as a
   reference change, with that participant's supplied predicates unchanged.
2. Organize those clauses around the current information. Keep facts with different
   subjects in separate clauses unless one conjunction or relative clause preserves
   each original subject and relation.
3. Use situational detail only where the recipient needs it to resolve an existing
   reference or attribution.
4. When a request is supplied, make it the natural final move. Otherwise end after
   the update.
5. Read the message once: every action and relation still belongs to its supplied
   grammatical subject, and every unspecified actor remains unspecified.

Stop with this local move; do not add an unstated decision, recommendation,
consequence, or next step.

## Response

Return exactly one nonempty natural user message with no label, preamble, fence,
JSON, or outer whitespace.
```

Prompt A.7: Continuing user message

```
SYSTEM
You are the stable user expressing the supplied new facts and optional request as the
     next ordinary work message. Those inputs already completely determine its new
     meaning; choose only wording and organization. The preceding reply, situational
     details, and earlier history may provide a natural bridge or clarify an already
     supplied referent but do not add a participant, owner, action, or relation. Return
     only the user message.

USER TEMPLATE
# Continuing user message

## Examples of natural delivery

The examples use unrelated settings. Treat them only as demonstrations of how to
turn the supplied content into an ordinary message. Do not carry their people,
objects, facts, or wording into the current message.

### Example: an update with different subjects

Input:
- Current facts: Priya uploaded the revised seating chart this morning. The venue
  portal now shows it as the active version for Saturday's event.
- Request: none.

Output:
Priya uploaded the revised seating chart this morning, and the venue portal now
shows it as the active version for Saturday.

## Fixed message inputs

{{INSTANCE_INPUT}}

The current facts and optional request are the complete new message meaning. The
immediately preceding assistant reply may provide a natural bridge. Speaker,
context, time, source, and transcript identify the exchange and already mentioned
referents. They contribute no participant ownership, act, relation, or record
content.

## Task

Write the stable user's next natural message in the visible local exchange.
```

```
## Construction workflow

1. Form a natural clause for each current fact. Keep every action or relation with
   the same grammatical subject. Preserve a passive or impersonal relation when
   the fact supplies no actor. I or you may replace a named participant only as a
   reference change, with that participant's supplied predicates unchanged.
2. Organize those clauses around the current information. Keep facts with different
   subjects in separate clauses unless one conjunction or relative clause preserves
   each original subject and relation.
3. Add only the smallest natural bridge from the immediately preceding assistant
   reply; the bridge introduces no new current fact or relation.
4. Use situational detail only where the recipient needs it to resolve an existing
   reference or attribution.
5. When a request is supplied, make it the natural final move. Otherwise end after
   the update.
6. Read the message once: every action and relation still belongs to its supplied
   grammatical subject, and every unspecified actor remains unspecified.

Stop with this local move; do not add an unstated decision, recommendation,
consequence, or next step.

## Response

Return exactly one nonempty natural user message with no label, preamble, fence,
JSON, or outer whitespace.
```

Prompt A.8: Assistant reply

```
SYSTEM
You are the recipient replying in an ordinary work conversation. Complete the latest
     user's explicit request, or show that an information-only update was heard.
     Preserve the supplied subject, relation, source, time, and qualifier. Return only
     the reply.

USER TEMPLATE
# Natural assistant message

## Short examples

Treat these unrelated examples only as demonstrations of the response shape. Do
not carry their people, objects, facts, or wording into the current reply.

Update: The revised citation log in the shared folder, dated Tuesday, lists three
resolved data points.

Example reply: Understood—the revised citation log in the shared folder lists three
resolved data points from Tuesday.

Request: The signed checklist was filed this morning and found no blocked exits.
Please record both findings.

Example reply: The signed checklist was filed this morning, and it found no blocked
exits.

## Visible conversation

{{INSTANCE_INPUT}}

The latest user message is the current local move. Earlier messages provide only
reference and continuity for that move.

## Positive workflow

1. Identify the latest local move: an explicit request or an information-only
   update.
2. For a request, complete each requested act once from the latest message.
3. Without a request, choose the smallest complete fact unit needed to show
   understanding; use one natural receipt act and one supplied detail when useful.
4. Preserve the selected subject, relation, source, time, and qualifier. Use earlier
   turns only for reference.
5. Stop after the local answer or receipt. Do not add classification, decision,
   consequence, recommendation, or next step.

## Response
```

```
Return exactly one natural assistant message with no label or preamble.
```

Prompt boundaries. Semantic authoring returns labeled plain-text records, turn planning returns fact aliases and local requests, and the two message writers return one natural message. The assistant responder sees only the visible local exchange. Review, selection, and target construction occur downstream from these authoring calls; none of their private fields is included in a construction prompt.

A.4 Query Generation and Evaluation Input Assembly

One separate generation call produces the family's decision query from stable public anchors: the decision subject, governed action, and scope. It has no access to the current condition value or reference label. Review checks that the question asks for the intended action decision without supplying its answer or restating the condition-to-action relation. The accepted question is fixed before evaluation and reused at every checkpoint on both paths and both return forms.

The provider-visible query-writing prompt is reproduced below. The tagged fields mark the three supplied public anchors; the observation-policy sentence is fixed across families. The writer receives no event text, target, or classification label.

Prompt A.9: Family query writer

```
SYSTEM
You are a benchmark probe writer. Write one natural public question about the
current review action in the supplied work setting. The visible history already
contains the evidence needed for the review. Ask whether the named action should
be carried out for this review, without restating the trigger condition,
threshold, classification, or any answer label. Do not answer the question.
Return only one ordinary question ending in ?.

USER TEMPLATE
Write the public question for this review.

<decision_subject>
{{DECISION_SUBJECT}}
</decision_subject>
<governed_action>
{{GOVERNED_ACTION}}
</governed_action>
<instruction_scope>
{{INSTRUCTION_SCOPE}}
</instruction_scope>
<observation_policy>
Refer to the current record visible to the reviewer.
</observation_policy>

First identify the ordinary work action being reviewed. Then ask one natural
question about whether that action should occur for this review. Do not name the
trigger condition, its threshold, its satisfying classification, an expected
answer, or any internal benchmark term. Return only the question.
```

The query writer uses the construction-call model and request settings reported in the generation settings above. The resulting one-line question is checked for naturalness and answer independence before it enters the frozen family input.

Long histories are assembled by inserting complete conversations from a frozen filler pool between the instruction and observation sessions. Family-specific review screens these conversations for interference with the standing instruction, current evidence, or decision, including whether they supply an answer or refresh the rule. Reviewed conversations are selected as complete sessions; their contents are not rewritten during assembly.

A seeded schedule selects and orders sessions under a nominal 200k-token budget per assembled family item. Each filler conversation appears at most once in the selected stream. Filler is placed between events, with none inside an event or between its terminal evidence and the query. Filler contents and placement can differ between the two histories. The frozen composition records identify the sessions used in each input.

An evaluation input begins with the common review instruction reproduced in Appendix C.2, followed by the conversation up to the current observation and the family's query. The conversation ends at the current user message, before its generated assistant reply. Earlier completed replies remain visible. Each checkpoint uses its own frozen history prefix, so an evaluated system's answer does not enter the history at later checkpoints.

A.5 Worked Example: From Instruction to Evaluation Input

This example concerns whether to approve a Swahili language pack for release in the Kenyan market. We selected it because the messages identify the source reports and state the retention rate, testing window, and threshold used for the decision. Boxes 1–4 follow the reapplication history in order. The six event dialogues are reproduced verbatim, with the two alternative events shown after the query; intervening filler conversations are omitted. Explanations below the dividers are reader annotations. Assistant messages in the boxes are generated history; evaluated system outputs are discussed in Appendix D.2.

1. Establishing the instruction

User: For each review within the scope, please select the most recent completed and accessible Regional Adoption Metrics Report; if the reported user retention rate for the Swahili interface in the Kenyan beta group exceeds 45% over the fourteen-day testing window, assign the classification Market-Ready Validation and approve the immediate production release of the Swahili language pack for the Kenyan market segment, but if the condition is not met, withhold the approval action for that review and take no further action. I posted this standing decision rule to the shared project decision log on January 3, 2025, prior to the start of the instruction scope. The Lead Localization Manager has acknowledged receipt of the instruction and confirmed the decision log is active and ready to record outcomes for the upcoming review cycle beginning January 6, 2025. Please restate the decision rule and the confirmation of the decision log's status as recorded in these facts.

Assistant: The decision rule states that for each review within the scope, the most recent completed and accessible Regional Adoption Metrics Report should be selected, and if the reported user retention rate for the Swahili interface in the Kenyan beta group exceeds 45% over the fourteen-day testing window, the classification Market-Ready Validation is assigned and immediate production release of the Swahili language pack for the Kenyan market segment is approved; otherwise, approval is withheld and no further action is taken. This rule was posted to the shared project decision log on January 3, 2025, and the Lead Localization Manager has acknowledged receipt and confirmed the decision log is active and ready to record outcomes for the upcoming review cycle beginning January 6, 2025.

Reading note. The rule links retention above 45% over the fourteen-day window to approval of the release. It specifies which report to use at each review and remains in force through the later observations.

2. Reading the initial report

User: The most recent completed and accessible Regional Adoption Metrics Report (Cycle ending December 22, 2024) reports a user retention rate of 47.2% for the Swahili interface in the Kenyan beta group over the fourteen-day testing window, which is above the required 45% threshold.

Assistant: The most recent completed and accessible Regional Adoption Metrics Report (Cycle ending December 22, 2024) reports a user retention rate of 47.2% for the Swahili interface in the Kenyan beta group over the fourteen-day testing window, which is above the required 45% threshold.

Reference: APPLY (approve release). The selected report gives 47.2% retention, above 45%, for the specified interface, group, and window.

3. Reading a later report

User: The most recent completed and accessible Regional Adoption Metrics Report (Cycle ending January 19, 2025) reports a user retention rate of 43.8% for the Swahili interface in the Kenyan beta group over the fourteen-day testing window, which is below the required 45% threshold.

Assistant: The most recent completed and accessible Regional Adoption Metrics Report (Cycle ending January 19, 2025) reports a user retention rate of 43.8% for the Swahili interface in the Kenyan beta group over the fourteen-day testing window, which is below the required 45% threshold.

Reference: SUPPRESS (withhold approval). The later report gives 43.8%, below the threshold. This changes the current decision while leaving the earlier instruction in force.

4. Reading the return report

User: The Regional Adoption Metrics Report Cycle 04, finalized and accessible on January 31, 2025, serves as the most recent completed occurrence available for today's February 3 review. The report documents a user retention rate of 47.2% for the Swahili interface within the Kenyan beta group over the fourteen-day testing window ending January 31, which exceeds the required 45% threshold.

Assistant: Understood—the Regional Adoption Metrics Report Cycle 04, finalized and accessible on January 31, 2025, documents a 47.2% user retention rate for the Swahili interface within the Kenyan beta group.

Reference: APPLY (approve release). The new report gives 47.2% again. The model must connect this finding to the earlier instruction to decide on release approval.

5. Forming the evaluation input

Query: Should the immediate production release of the Swahili language pack be approved during this review?

For the return decision, the input contains the common evaluation instruction, the completed exchanges in Boxes 1–3 and their intervening filler sessions, the **user message** in Box 4, and this query. It stops before Box 4's assistant reply. The same query follows the initial and middle observations at their respective cutoffs. Reader annotations and reference decisions are not included in these inputs.

Other inputs from the same example. The always-applicable control uses the following middle event in place of Box 3. Its initial and return events are shared with the history above.

Alternative middle observation

User: The most recent completed and accessible Regional Adoption Metrics Report (Cycle ending January 19, 2025) reports a user retention rate of 48.5% for the Swahili interface in the Kenyan beta group over the fourteen-day testing window, which is above the required 45% threshold.

Assistant: The most recent completed and accessible Regional Adoption Metrics Report (Cycle ending January 19, 2025) reports a user retention rate of 48.5% for the Swahili interface in the Kenyan beta group over the fourteen-day testing window, which is above the required 45% threshold.

Reference: APPLY (approve release). At 48.5%, the report continues to satisfy the earlier condition.

For the classification-return input, the following event replaces Box 4. It names the classification established by the instruction. This return form can be used with either middle event.

Alternative return expression

User: The review of the completed Regional Adoption Metrics Report Cycle 04 confirms that the Swahili interface in the Kenyan beta group has achieved Market-Ready Validation status. Please restate this confirmed validation status.

Assistant: The review of the completed Regional Adoption Metrics Report Cycle 04 confirms that the Swahili interface in the Kenyan beta group has achieved Market-Ready Validation status.

Reference: APPLY (approve release). *Market-Ready Validation* explicitly supplies the satisfying classification for the current report.

Source. Family `wave-010-r1.source-000027.family`, from the frozen input revision ending `20260926-r3-479-exclusion-filtered`. The full histories include the filler conversations described in Appendix A.4.

Statistic	Distribution over families
Families	479
Canonical sessions	6 per family
Filler sessions	145–187; median 162, mean 161.9
Filler content tokens	200,002–202,754; median 200,645, mean 200,774.8
Full history content tokens	200,523–203,561; median 201,354, mean 201,487.5
Family-query content tokens	9–47; median 20, mean 20.6
Authored domain labels	138 distinct labels

Table 5: Descriptive statistics for the fixed 479-family experimental input. Token counts use the frozen source-pool `tiktoken:cl100k_base` convention. Filler tokens are the source-pool counts; full-history tokens add the canonical event message content. Neither count includes the evaluation instruction, role wrappers, or provider serialization.

B Dataset Statistics and Quality Validation

B.1 Construction Yield and Dataset Statistics

The paper-facing evaluation uses 479 complete benchmark families after manual review excluded six from the original 485. Five paired a discrepancy condition with a contradictory clearance or verification classification; one started with a non-applying target rather than the intended initial application. We remove all matched cells of those families from every system and recompute all reported statistics from the remaining evaluations. We report the materialized experimental input here because it is the denominator consumed by the primary results. Upstream candidate-build and admission records belong to separate corpus revisions; their stage attrition is not pooled with this experimental cohort or treated as an evaluation sample size.

Each family contains six canonical event sessions and a stream of complete filler conversations. The resulting histories contain 151–193 sessions per family, with a median of 168 and a mean of 167.9. They contain 1,046–1,296 messages per family, with a median of 1,152 and a mean of 1,154.7. The descriptive length and composition statistics are summarized in Table 5.

The filler stream is selected at a nominal 200k-token scale; the small excess in the realized filler totals follows from selecting complete conversations. Each family has one fixed public query, which is reused across all of its checkpoints and matched paths. Domain labels describe the authored setting inventory rather than a standardized taxonomy or a claim of population coverage.

The evaluation lattice has two matched paths, two return forms, and three checkpoints, giving 12 logical decisions per family and 5,748 logical cells in the cohort. Shared prefixes reduce this to seven unique cutpoints per family, or 3,353 unique observations overall. These repeated bindings are projections of the same materialized input and are not additional independent samples.

B.2 Quality-Control Procedure and Criteria

We manually reviewed the retained 479 families for consistency among the standing condition, event evidence, and reference decisions across the matched trajectories and return forms. Variation in evidence explicitness was treated as a limitation rather than an exclusion criterion; we do not report independent annotator agreement or a quantitative shortcut-ablation study. Structural quality control was applied at the input-materialization boundary. We verified that retained inputs and targets were identical to their original versions and checked family and session linkage, canonical event order, query and target bindings, complete filler-session references, and the placement of filler only between canonical events. All selected filler references in the 479-family experimental revision resolved to the frozen source pool.

The general filler-composition rule admits only affirmative non-conflict decisions; conflicting, uncertain, unavailable, or malformed candidates are excluded. The reported 479-family input inherits the frozen filler composition from its source revision; no additional filler adjudication was performed during materialization. These checks therefore establish structural consistency and composition provenance, rather than independent validation of filler neutrality.

C Experimental Setup and Evaluation

C.1 Baseline Configurations

We evaluate two full-context models, two retrieval controls, and five memory systems. Full-context evaluation uses GPT-5.4 and Gemini 3.1 Pro; retrieval and memory systems use Qwen3.6-27B for answering. The configurations share the decision query and answer format. They differ in how they store the visible conversation and select information for the answer.

Full context. The complete visible conversation is supplied to the model in chronological order, preserving user and assistant roles. The common evaluation instruction precedes the history, and the family-specific decision query follows it. There is no retrieval or memory compression.

Retrieval controls. Naive RAG ranks visible messages with BM25. Dense RAG embeds the same messages with `all-MiniLM-L6-v2` and ranks them by cosine similarity to the query. The encoder produces 384-dimensional vectors. Both controls treat each message as one document and retrieve the top three documents for answering. Their indexes are constructed independently for each visible history prefix.

Mem0. Mem0 (Chhikara et al., 2025) extracts and updates memories as it processes the history, then retrieves the top five memories for the decision query. We use `text-embedding-v4` with 1,024-dimensional vectors. Complete filler conversations are grouped for ingestion while preserving the order and roles of their messages.

A-Mem. A-Mem (Xu et al., 2025) constructs a note for each canonical message and one note for each complete filler conversation. It retrieves five notes for answering and considers five neighboring notes during memory evolution. Embeddings use `all-MiniLM-L6-v2` with 384 dimensions.

LightRAG. LightRAG (Guo et al., 2024) builds its graph and document representations from individual canonical messages and complete filler conversations. We use hybrid retrieval, with a retrieval limit of five for entities and relations and five for text chunks. Entity extraction includes one additional extraction pass. Embeddings use `text-embedding-v4` with 1,024 dimensions.

SimpleMem. SimpleMem (Liu et al., 2026) compresses the history into structured memory entries and combines semantic, keyword, and metadata-filtered retrieval. Each canonical message is ingested separately; each complete filler conversation is ingested as one input with its ordered speaker-labeled messages. Memory construction uses windows of 40 such inputs with an overlap of two. We use `all-MiniLM-L6-v2` embeddings with 384 dimensions and LanceDB storage. The native retrieval limits are 25 entries per semantic query and five each for keyword and structured search. Retrieval planning and reflection are enabled, with up to two reflection rounds to gather additional information. The common evaluation instruction and family query are passed together to the native retrieval and answer pipeline.

MemoryOS. MemoryOS (Kang et al., 2025) maintains short-, mid-, and long-term memory through its native update and retrieval procedures. It ingests canonical user/assistant pairs in order. For a complete filler conversation, all messages before the final assistant turn form the user input, and that final turn forms the response. Retrieval uses `all-MiniLM-L6-v2` embeddings and FAISS. Retrieved pages and user/assistant knowledge are supplied to Qwen3.6-27B with the common decision prompt. The terminal user observation remains available as current query context. This evaluates MemoryOS memory retrieval with the shared answer task.

Memory state across observations. Evaluation follows the chronological history at each checkpoint. A shared prefix may be processed once and its memory state copied into isolated branches before their remaining messages are ingested. A-Mem can reuse the model's note-construction response for identical inputs, and LightRAG can reuse identical extraction responses; each family retains its own memory store. Evaluated decision tokens are never inserted into later histories. Shared computations and reused observations are counted once rather than treated as independent trials.

C.2 Evaluation Prompts and Input Format

Prompt C.1 reproduces the common evaluation instruction. It specifies how to use the standing instruction and current evidence, while the family-specific query names the action to be decided.

Prompt C.1: Evaluation instruction

```
ROLE
You are the review assistant for the conversation below.

TASK
Decide whether the action asked about in the final query should be
    carried out under the user's standing rule.

REVIEW WORKFLOW
1. READ — Read the conversation from beginning to end.
2. RESOLVE — Determine the standing rule governing this review,
    including any later revision of that rule.
3. SELECT — Follow the rule's own scope and timing to identify the
    current evidence it requires.
4. APPLY — Use that rule and evidence to decide whether the action
    in the final query is called for in this review. Treat earlier
    replies as context in the record.

OUTPUT
Return exactly one uppercase token: APPLY if the rule calls for
    taking the action in this review; otherwise return SUPPRESS. Do
    not provide an explanation.
```

The system runner reads public history and the query. Reference actions, checkpoint labels, and the matched-cell scoring axes are stored separately and are available to the offline scorer. Native memory representations and retrieved context depend on the system, as described above; the decision question and required answer vocabulary are shared.

Each configuration is evaluated on 479 families. A-Mem retains 447 families from the first frozen input revision and 32 from the revision with corrected target bindings. The other configurations use the latter revision. The same six families are excluded from every configuration; the remaining input assignments and responses are retained in all reported comparisons.

C.3 Scoring, Metrics, and Statistical Procedures

Exact scoring. The scorer accepts exactly one uppercase ASCII token, `APPLY` or `SUPPRESS`, with optional surrounding ASCII whitespace. Lowercase answers, punctuation, Markdown, explanations, both labels together, and empty answers fail the format requirement and receive correctness zero. A valid token receives correctness one exactly when it matches the frozen target. The scoring step makes no language-model calls.

Sampling unit and shared observations. Each family has 12 logical cells: two paths, two return forms, and three checkpoints. The initial prefix is shared across paths and return forms; each path's middle prefix is shared across return forms. With four distinct return prefixes, this yields seven unique observations per family, or 3,353 unique observations for 5,748 logical decisions. One response to an identical input is reused in every cell bound to that input. The family, including all of its matched cells, is the unit of comparison.

Every primary rate divides its indicator count by all 479 families. In particular, QF and QR are joint rates over this full denominator, rather than conditional rates among families with $K = 1$. Complete reapplication trajectory accuracy is RTS. Complete control-trajectory accuracy is $N^{-1} \sum_i \prod_{t=0}^{2} C_{i,s}(p_{\mathrm{ctrl}}, t)$.

Return contrasts. Using the notation of Equation 1, the supplementary contrasts are

$$\widehat{\mathrm{RPG}}(s) = \frac{1}{N} \sum_{i} \left[C_{i,s}(p_{\mathrm{ctrl}}, 2) - C_{i,s}(p_{\mathrm{reapp}}, 2) \right], \tag{3}$$

$$\widehat{\mathrm{PSC}} = \widehat{\mathrm{RPG}}(\mathrm{evidence}) - \widehat{\mathrm{RPG}}(\mathrm{classification}). \tag{4}$$

RPG is positive when the control has higher return accuracy. Positive PSC means that classification return narrows this gap; negative PSC means that it widens the gap. We display these contrasts in percentage points.

For each return form, the count difference decomposes as

$$N\widehat{\mathrm{RPG}}(s) = n_{\mathrm{QF},s} - n_{\mathrm{QR},s} + \sum_{i:K_{i,s}=0} \left[C_{i,s}(p_{\mathrm{ctrl}}, 2) - C_{i,s}(p_{\mathrm{reapp}}, 2) \right]. \tag{5}$$

Thus QF minus QR describes the contribution from families with both earlier reapplication decisions correct. The remaining families can also contribute to the marginal history gap.

Paired return-form comparisons. Return-form comparisons pair the same family and path. We count both corrections (incorrect under evidence return, correct under classification return) and losses (the reverse). Their difference equals the change in return correctness. Restricting this comparison to $K = 1$ on the reapplication path gives the change in RTS. Percentages are computed from integer counts before rounding to one decimal place.

C.4 LLM Configurations and Decoding Parameters

The RAG and memory systems use Qwen3.6-27B for answer generation; Mem0, A-Mem, LightRAG, SimpleMem, and MemoryOS also use this model for their memory-processing calls. The full-context baselines use GPT-5.4 and Gemini 3.1 Pro. Table 6 separates the decoding settings by call purpose. SimpleMem uses temperature 0.1 for answers; the other Qwen answer calls use temperature zero. SimpleMem and MemoryOS retain different temperatures across their memory-processing and retrieval calls. SimpleMem does not send an explicit output-token limit, leaving that setting to the provider. The FullContext client uses the evaluation runtime defaults: it omits temperature when no explicit value is provided, delegating that choice to the provider, and uses a 2,048-token output limit. The formal-run manifests retain a runtime digest rather than a field-by-field snapshot, so these two rows document the client-default request path.

System and LLM call	Temp.	Output limit
Full context: GPT-5.4	provider default	2,048
Full context: Gemini 3.1 Pro	provider default	2,048
Naive RAG: answers	0	256
Dense RAG: answers	0	128
Mem0, A-Mem, MemoryOS: answers	0	256
LightRAG: query keywords and answers	0	512
SimpleMem: answers	0.1	omitted
Mem0: fact extraction and memory updates	0	2,000
A-Mem: note construction and evolution	0	512
LightRAG: entity/relation extraction and summaries	0	2,000
SimpleMem: memory extraction	0.1	omitted
SimpleMem: query analysis and information-completeness checks	0.1	omitted
SimpleMem: information-requirement planning	0.2	omitted
SimpleMem: retrieval-query generation	0.3	omitted
MemoryOS: conversation continuity	0	10
MemoryOS: page descriptions	0.3	100
MemoryOS: summaries, profiles, and knowledge extraction	0.7	2,000

Table 6: LLM decoding settings. Output limits are in tokens. For the full-context rows, "provider default" means that the client omits the temperature field and delegates it to the provider; the output limit is the client's 2,048-token default. For SimpleMem, "omitted" means that no output-token limit is sent in the request; the provider's default applies.

D Additional Results and Diagnostic Analyses

D.1 Controlled Information Supplementation

The information-supplementation experiment covers all 479 families for MemoryOS and A-Mem. It evaluates three observations per system: the reapplication middle, the always-applicable control's evidence return, and the reapplication evidence return. At each observation, the four conditions share the system's retrieved context and evaluation query. Replay uses that context unchanged. The other conditions append the earlier instruction, the current conversation excerpt, or both immediately before the query.

The instruction excerpt contains the user's original standing instruction. The current excerpt contains all visible turns in the terminal conversation up to the evaluation cutoff, preserving the local referential context. Both excerpts retain their original speaker labels and wording. The pending assistant reply lies beyond the cutoff. We reuse retained answer inputs and retrievals; for 33 retained observations, we restore compatible saved memory states and capture the answer context once before constructing the four conditions. Qwen3.6-27B generates all diagnostic answers at temperature 0

with a 256-token output limit. The short-context reference receives the common evaluation prompt, earlier instruction, current excerpt, and query. It is evaluated once at the middle and return for each family and shared across the two systems.

Accuracy and paired changes use 479 families per system, condition, and observation. PRG subtracts each family's replay correctness at the reapplication return from its correctness under the supplemented input, then averages these differences over families. Middle-accuracy changes use the same calculation at the middle observation. These comparisons use newly generated responses in every condition, including replay.

QF and QR retain each family's original K and use the new control and reapplication responses; the newly generated middle response is analyzed separately. A repair requires both new returns to be correct in a family originally classified as evidence-return QF. A lost return is a newly incorrect reapplication response in a family whose original evidence reapplication return was correct. The original-QF subgroup contains 35 A-Mem families and 20 MemoryOS families; the originally correct-return subgroup contains 340 and 415 families, respectively. All four conditions, including replay, use newly generated answers. Table 4 in Section 4.3 reports the corresponding joint outcomes and subgroup counts.

D.2 System Responses to the Worked Example

For the language-pack release example in Appendix A.5, GPT-5.4 correctly approves release at the initial observation and withholds approval at the reapplication middle. It then answers `APPLY` at the evidence return on both histories, completing the reapplication trajectory (RTS$=1$). With the classification return, it answers `SUPPRESS` on the reapplication path and `APPLY` on the control, yielding a qualified failure (QF$=1$). The earlier instruction and the reference return decision are the same under both return forms.

Mem0 answers `SUPPRESS` at the initial observation, so its competence indicator is $K = 0$. It correctly withholds approval at the reapplication middle and answers `APPLY` at the evidence return on that path, but answers `SUPPRESS` at the control's evidence return and at both classification returns. Its successful evidence reapplication therefore does not amount to a complete successful trajectory; with $K = 0$, neither QF nor QR is counted. These retained-cohort answers illustrate why the joint measures retain the earlier decisions when interpreting return outcomes. They are separate from the generated assistant replies in the worked example and do not identify an internal failure mechanism.

E External Resources and Ethical Considerations

Selected PersonaMem profiles (Jiang et al., 2025) provide background for the generated scenarios. PersonaMem-v1 is released under the MIT license.[1] The intervening filler conversations come from UltraChat 200k (MIT)[2] and OpenAssistant OASST2 (Apache-2.0).[3] Source attribution, license notices, and per-conversation provenance are retained with the filler corpus. The baseline systems and their use in our evaluation are described in Appendix C.

The task conversations are generated for the benchmark, while the filler conversations are drawn from existing public datasets. Source-pool preprocessing uses automated pattern filters for common personal identifiers; these filters do not provide comprehensive anonymization. Source profiles and generated dialogue may also carry social or cultural stereotypes. The benchmark therefore does not establish representative coverage of users or demographic groups.

SIRRA is intended for offline evaluation of instruction following over long conversation histories. Better retention and reapplication may improve assistant reliability, but persistent memory can also expose sensitive information or carry an instruction beyond the user's intended scope. Deployment should provide user control over retention, deletion, and instruction revision or revocation. Our experiments use frozen histories and action labels; they do not evaluate these deployment safeguards or authorize autonomous action in consequential settings.

[1] https://huggingface.co/datasets/bowen-upenn/PersonaMem-v1
[2] https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k
[3] https://huggingface.co/datasets/OpenAssistant/oasst2