DRIFTPAIR: COUNTERFACTUAL CARRY EVALUATION REVEALS MEMORY-SYSTEM REVERSALS UNDER CROSS-SESSION DRIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Independent-session scores can favor memory architectures that deteriorate when stale state crosses session boundaries. DriftPair exposes this selection error with a branch-aligned 2×2 protocol that crosses access to pre-session memory with a shared revision realization while matching tasks, policies, models, tools, decoding, budgets, and exogenous randomness. Across 990 tasks in four verifier-backed tool-agent families, carry benefits reuse-oriented policies in stable environments, yet revision produces 4.7- and 5.2-point carry–drift interactions for tiered and rolling summaries, compared with 1.3 and 1.2 points for temporal and verify–evict memory. These interaction gaps change architecture choice: independent sessions rank MemGPT/Letta first, whereas DriftPair ranks it fourth; carrying that choice into the revision regime costs 2.4 session-20 success points, adds 3.3 interference points, and consumes 2,070 additional tokens per episode. A `Qwen2.5-72B-Instruct` replication reproduces the $1 \rightarrow 4$ demotion. By making pre-session state the controlled variable, DriftPair links memory mechanisms to an architecture-selection consequence.

1 INTRODUCTION

An agent with persistent memory enters each session carrying a state produced by earlier tasks. Saved API recipes, schema assumptions, evidence links, and action plans accelerate later work while the environment remains stable. Revision changes the sign of that value: the same state can redirect retrieval toward superseded interfaces, invalid constraints, or obsolete evidence. Choosing a memory architecture therefore requires a conditional quantity—the effect of entering the current task with pre-session state under the revision that actually occurred.

Common evaluation views estimate different pieces of this quantity. Independent sessions measure competence without historical exposure. Unpaired multi-session trajectories measure performance after a history, but mix the contribution of that history with task composition, stochastic tool interaction, and compute. Drifted trajectories reveal adaptation pressure while leaving stable carry benefit and revision-induced carry harm combined. DriftPair connects these views through a branch-aligned 2×2 design. At each scored session, carried and memory-cleared branches begin from the same current environment and share the revision realization, policy, model, tool implementation, budget, and exogenous random stream. The within-instance contrast estimates the signed effect of pre-session state; crossing the contrast with drift identifies the carry–revision interaction.

Agent and tool benchmarks establish long-horizon competence across software, web, embodied, and planning environments (Jimenez et al., 2024; Zhou et al., 2024; Koh et al., 2024; Mialon et al., 2023; He et al., 2024; Liu et al., 2024; Ma et al., 2024; Wang et al., 2023; Xie et al., 2024; Valmeekam et al., 2022; Ye et al., 2024; Shridhar et al., 2021). Memory-focused suites progressively add retention and stateful structure: MemBench and MemoryAgentBench stress incremental multi-turn memory, LoCoMo and LongMemEval probe long-history recall, and MemoryArena evaluates interdependent multi-session tasks and AMA-Bench evaluates memory through questions about agent–environment trajectories (Tan et al., 2025; Hu et al., 2025; Maharana et al., 2024; Wu et al., 2024; He et al., 2026; Zhao et al., 2026). DriftPair makes branch-aligned carry under a shared revision the benchmark-defining intervention and carries its within-instance effect through rank,

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.

Table 1: Evaluation designs and the estimands enabled by their defining axes.

Design family	Interdependent	Controlled revision	Branch-aligned carry	Evaluation object
End-task agent suites	partial	incidental	no	task competence and tool use
Turn-level memory suites	partial	no explicit axis	no	retention and retrieval quality
Session/trajectory memory suites	varies	natural or scripted	no	task- or QA-based memory evaluation
DriftPair	yes	shared realization	yes	carry–drift interaction and selection regret

frontier, and selection-regret decisions. Agent reasoning, tool use, memory architectures, retrieval methods, and factuality evaluators provide complementary context for the storage, revision, and evidence-support mechanisms exercised by this intervention (Qin et al., 2023; Yao et al., 2023; Shinn et al., 2023; Packer et al., 2023; Park et al., 2023; Madaan et al., 2023; Asai et al., 2023; Min et al., 2023; Gao et al., 2023; Izacard et al., 2022; Lewis et al., 2020; Guu et al., 2020; Karpukhin et al., 2020; Santhanam et al., 2021; Wei et al., 2024). Table 1 locates the resulting estimand relative to adjacent evaluation designs.

The identified interaction separates architecture classes. Under stable conditions, all four representative memory policies gain from carry. Under revision, tiered and rolling summaries exhibit 4.7- and 5.2-point interactions, while temporal linking and verify–evict control limit the interaction to 1.3 and 1.2 points. The mechanism has a direct decision cost: on the same 990 tasks and nine systems, independent sessions rank MemGPT/Letta first, while DriftPair ranks it fourth and selects the learned reference. Transferring the independent-session choice into the revision regime loses 2.4 S20 points, adds 3.3 interference points, and consumes 2,070 more tokens per episode. The second base LLM reproduces the three-rank demotion.

This paper makes three contributions:

1. A revision-conditional carry estimand and branch-aligned factorial protocol separate stable carry benefit, drift pressure, and their interaction while preserving policy identity and shared exogenous randomness.
2. A decision analysis converts estimand mismatch into measurable selection regret: the independent-session choice falls three ranks and loses success, interference, and compute efficiency in the revision regime.
3. A mechanism and robustness package connects stale-state reconciliation to interaction magnitude across nine systems, two base LLMs, four onset schedules, and verifier-backed outcomes.

The resulting evaluation standard follows the state variable through its complete scientific chain: intervention, signed effect, interaction mechanism, uncertainty, and architecture-selection consequence.

2 Branch-Aligned Identification Protocol

Paired intervention. For memory policy m, task k, session s, and seed r, DriftPair forks execution immediately before the session. The carried branch exposes memory written during sessions $1{:}s{-}1$; the cleared branch removes only that pre-session state and retains the policy's ordinary read, write, summarize, and evict behavior during session s. Both branches preserve the task instance, current environment state, prompt, tool implementation, base model, decoding, budget, drift realization, policy, and common exogenous random stream. Downstream actions and their tool outputs may diverge as a consequence of the intervention. This alignment makes access to pre-session state the treatment and the branch-score difference a within-instance contrast.

Measurement and parity. Every configuration uses seeds $\{13, 37, 61, 71, 97\}$. Where shown, mean$\pm$SD denotes the arithmetic mean and sample standard deviation of the five seed-level values; these descriptive summaries are not confidence intervals. Latency reports median

and p95, while interference and contamination retain their paired mean estimates. All systems use identical prompts, tool versions, redaction, instrumentation, token and latency caps, and `bge-large-en-v1.5` embeddings where the interface permits replacement. The primary stack uses `Llama-3.1-70B-Instruct`; the replication changes only the base LLM to `Qwen2.5-72B-Instruct`. Runs use four NVIDIA A100 80GB GPUs, CUDA 12.4, PyTorch 2.3.1, and vLLM 0.5.3. Appendix Tables 7 and 9 consolidate notation and policy definitions.

2.1 Standard memory API

Agents interact through four typed calls: `read(query, top_k, mode)`, `write(fragment, type, ttl, provenance)`, `summarize(fragment_ids, budget)`, and `evict(criteria)`. The API fixes argument names so logs are comparable across implementations. All calls are logged with token, latency, bytes, and decision context.

2.2 Instrumentation and logging schema

DriftPair logs memory operations as structured events covering the decision context (episode id, step id, task family, session index, budget state), the operation itself (read/write/summarize/evict plus parameters), the resulting costs (tokens, wall time, index bytes, verifier time), and the outcomes (success/failure, verifier pass/fail, and taxonomy attribution).

2.3 Task families

The four families are tool-verified coding (Jimenez et al., 2024), SQL/data (Li et al., 2023), research synthesis, and policy-shift planning (Shridhar et al., 2021). Each scored task belongs to one family. Sessions are consecutive evaluation rounds with memory carry-over. Drift is injected at a configurable onset session $t^\star$ by modifying tool schemas, inserting contradictions, or flipping constraints. Contradictions are injected as mutually inconsistent fragments (coding: API specification versus tool output; SQL: schema constraints; synthesis: evidence claims; planning: permissions or constraints). Staleness denotes superseded evidence, while contradiction denotes mutually inconsistent evidence without clear supersession.

Drift-onset regimes. DriftPair scores four schedules. *Fixed* sets $t^\star = 10$; *Early* draws $t^\star \sim \mathcal{U}\{3, \ldots, 7\}$; *Late* draws $t^\star \sim \mathcal{U}\{15, \ldots, 20\}$; and *Randomized* draws $t^\star \sim \mathcal{U}\{5, \ldots, 18\}$. Shared seeds align onset draws across policies. Table 13 reports rank and first-front stability across all schedules.

Table 2: DriftPair task families and denominators. Controlled stressors align paired executions, while heterogeneous verifiers prevent a single workflow template from determining the aggregate.

Family	# tasks	Scored sessions	Stressors	Notes
Coding	400	S20; S50 extension	staleness, poisoning	tool-verified signals; repository drift
SQL/Data	250	S20	contradiction, shift	schema evolution; constraint changes
Synthesis	180	S20	contradiction	citation requirements; revisions
Planning	160	S20; S50 extension	shift, budget misuse	mid-horizon policy flips

2.4 Stressors and budgets

Staleness, contradictions, and poisoning attempts are injected at controlled rates once the session-indexed drift onset $t^\star$ fires (Figure 1). Budgets include token caps, latency caps, and memory caps that force eviction. Injection is randomized per episode using the global seed and applied identically across policies. The controlled frequencies support paired attribution and span a broad range of cross-session revision pressure. For ease of adoption, a minimal core evaluation runs the medium tier at sessions 1/5/20 without sweeps; session 50 and stressor-intensity sweeps are optional extensions. Concrete examples make the protocol less abstract. In coding, a session-6 patch recipe can become stale when session-12 tool output introduces a renamed API symbol. In SQL, a column rename plus

a new uniqueness constraint can invalidate an earlier memorized query plan. In synthesis, a claim can be revised by a later source while an older citation remains retrievable. In planning, a location or tool permission can flip mid-horizon so that a once-valid action becomes disallowed.

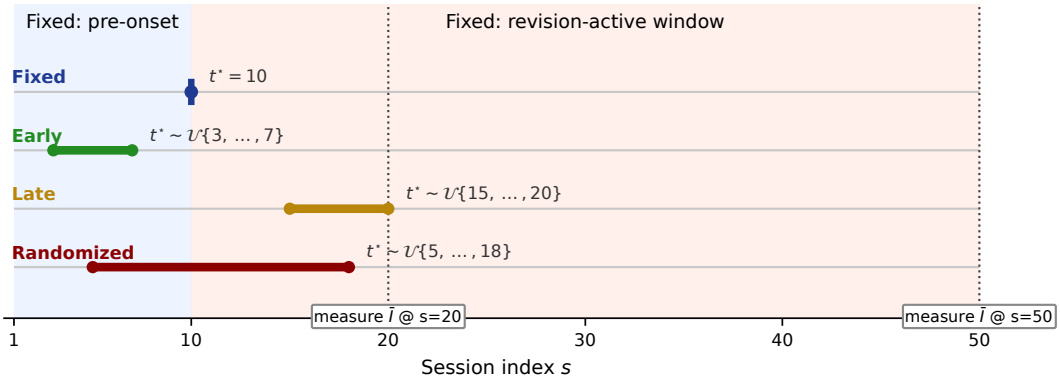

Figure 1: Branch-aligned protocol. Carried and cleared branches share each onset draw; S20 covers all 990 tasks, while the session-50 extension covers 560 Coding and Planning tasks.

Budget tiers. Caps are enforced identically across policies: *small* (10–14k tokens/episode, 15–30s latency, 0.5–1.0GB memory, 0–1 verifier calls), *medium* (16–24k, 30–60s, 1.5–3.0GB, 0–2) for all headline tables, and *large* (28–48k, 60–120s, 4–8GB, 0–4). Tokens per episode is an official ranking axis, preventing quality gains from being separated from model-compute cost.

Stressor mechanics. Staleness (5–30%) inserts newer replacement fragments; contradiction (10–35%) inserts mutually inconsistent fragments with provenance; poisoning (5–25%) inserts tagged adversarial fragments. Family-specific verifiers measure stale retrieval, contradiction handling, exposure, attack success, and recovery time under the configured stressor levels.

Metrics and verifiers. Let $d \in \{0, 1\}$ denote a stable or revised environment and $a \in \{\text{fresh}, \text{carry}\}$ denote access to pre-session memory. The policy- and seed-indexed interference estimand is

$$I_{m,k,s,r}(d) = S^{\text{fresh}}_{m,k,s,r}(d) - S^{\text{carry}}_{m,k,s,r}(d), \quad \bar{I}_{m,s}(d) = \frac{1}{RK_s} \sum_{r=1}^{R} \sum_{k=1}^{K_s} I_{m,k,s,r}(d), \tag{1}$$

where $R = 5$, $K_{20} = 990$, and $K_{50} = 560$ for the Coding/Planning extension. One score $S^{a}_{m,k,s,r}(d)$ is recorded for each task, session, seed, branch, and revision condition. Thus S20 denotes the score at session index $s = 20$, not a 20-session average; per-episode resource accounting is reported separately. Positive interference denotes harmful carry, zero denotes no carry effect, and negative interference denotes beneficial carry; its range is $[-100, 100]$. The factorial interaction

$$J_{m,s} = \bar{I}_{m,s}(1) - \bar{I}_{m,s}(0) \tag{2}$$

measures how revision changes carry's marginal effect. *Stale retrieval rate* divides fragments predating a verifier-passed update by all session-s retrievals. *Injected-fragment exposure* is the fraction of episodes in which a tagged stressor fragment enters the action-attribution trace. Write efficiency is $W_{\text{eff}} = \Delta S / \Delta\text{tokens}_{\text{written}}$ when $\Delta S > 0$ and is undefined otherwise. Code uses pytest (30 s), SQL uses a schema checker and exact result-set diff (10 s), Synthesis uses deterministic claim-to-gold-span citation checks and numerical consistency within 1% tolerance, and Planning uses a deterministic constraint satisfier (5 s).

Reference scaffold. DriftPair separates task policy from memory policy through a deterministic runner, four-call adapter, cost accounting layer, and verifier wrappers with standardized pass/fail semantics.

Table 3: Difference-in-differences at S20. The final column reports $J_{m,20}$ from Equation 2.

System	Clear/ stable	Carry/ stable	Clear/ drift	Carry/ drift	Interaction
MemGPT / Letta	55.2	56.0	53.6	49.7	4.7
LangChain SummaryBuffer	54.1	54.8	52.8	48.3	5.2
Zep v0.21	52.8	53.4	52.6	51.9	1.3
Learned reference	53.0	53.6	52.7	52.1	1.2

3 Scoring and Reporting

Primary report card. Each family supplies a verifier-backed success metric. The official session-20 score is the task-weighted mean $S20_m = (400S_{m,\text{Coding}} + 250S_{m,\text{SQL}} + 180S_{m,\text{Synthesis}} + 160S_{m,\text{Planning}})/990$, computed before rounding. DriftPair reports this score together with $\bar{I}_{m,20}(1)$ and $J_{m,20}$. Session 20 is at or after onset under every schedule and covers all 990 tasks; session 50 provides a longer-horizon view over the 560 Coding and Planning tasks with that extension.

Multi-objective ranking. Policies are Pareto-ranked on (session-20 success↑, $\bar{I}_{20}(1)$ ↓, model tokens/episode↓). Model tokens include prompt and generation tokens under shared accounting. Latency remains constrained and reported; verifier time and index bytes are diagnostics. The family definitions and task denominators make the aggregate explicit.

Decision output. Every policy receives the fixed triple (S20, interference, tokens/episode), exact first-front membership, and a separate display rank by S20. Front inclusion is the multi-objective decision; the displayed selection is the highest-S20 point on that front. Reporting these outputs separately exposes whether evaluator choice changes quality priority or trade-off membership.

Synthesis scoring. The Synthesis family score averages deterministic atomic citation-span matches after Unicode NFKC normalization and numerical-claim consistency within 1% tolerance. It enters the task-weighted S20 aggregate exactly once, through the 180-task family term above.

4 Revision-Conditional Carry Harm and Its Decision Consequences

The evidence follows the identification chain: a factorial experiment isolates the carry–drift interaction, a fixed-task comparison measures its system-selection consequence, and a second base model tests whether the resulting rank reversal persists. All headline experiments use the medium budget, 990 tasks, and five aligned seeds.

4.1 Carry–drift interaction

The 2×2 design crosses access to pre-session memory with revision while holding the task mix and harness fixed. Stable environments make carry beneficial for all four representative systems. Revision then separates stale-summary behavior from temporal and verify–evict memory.

For MemGPT/Letta, interference changes from -0.8 under stability to $+3.9$ under drift, yielding a 4.7-point interaction. LangChain reaches 5.2 points, whereas temporal edges and verify–evict control limit the interaction to 1.3 and 1.2 points. This decomposition identifies revision-induced stale carry as the mechanism tested by the recommendation analysis.

4.2 Fixed-task decision consequence

Table 4 holds the tasks, system universe, prompts, tools, verifiers, and seeds fixed. The independent-session condition resets pre-session state and revision, while DriftPair activates the revision regime and scores the paired carry contrast. The factorial result above supplies the mechanism; this composite

Table 4: Fixed-task recommendation comparison over the same 990 tasks, nine systems, and five seeds. Scores are five-seed means±SD. Independent sessions select MemGPT/Letta; paired revision selects the learned reference.

System	Independent S20	Rank	DriftPair S20	Rank
MemGPT / Letta	55.2±1.1	1	49.7±1.4	4
LangChain SummaryBuffer	54.1±1.2	2	48.3±1.5	6
Learned reference	53.0±1.0	3	52.1±1.3	1
Zep v0.21	52.8±1.1	4	51.9±1.2	2
Hybrid graph+vector	52.2±1.2	5	50.6±1.4	3
Summarization memory	51.9±1.3	6	48.6±1.5	5
Naive retrieval	51.1±1.3	7	45.2±1.8	8
LlamaIndex VectorMemory	50.7±1.4	8	46.4±1.6	7
No memory	44.3±1.5	9	43.8±1.6	9

Table 5: Replication of evaluator-induced recommendation changes across base LLMs.

Base LLM	Independent selection	Paired selection	MemGPT rank
Llama-3.1-70B	MemGPT/Letta	Learned reference	1→4
Qwen2.5-72B	MemGPT/Letta	Zep v0.21	1→4

evaluator comparison measures the resulting decision change. MemGPT/Letta moves from rank 1 to rank 4, while the learned reference moves from rank 3 to rank 1.

Choosing the highest-success system from the independent condition selects MemGPT/Letta. Under the paired evaluation, that choice trails the learned reference by 2.4 S20 points, incurs 3.3 additional interference points, and consumes 2,070 more tokens per episode. Task-level paired outcomes for MemGPT/Letta versus the learned reference are 2,014/1,608/1,328 wins/losses/ties across 4,950 seed-task pairs: MemGPT/Letta wins more individual pairs, but larger losses produce the 2.4-point mean deficit.

4.3 External systems

The four external systems span distinct memory behavior: top-k retrieval, rolling summaries, tiered summary flushing, and a temporal knowledge graph. Their default results appear in Appendix Table 15 under the shared DriftPair harness.

4.4 Base-model replication

The replication changes only the base LLM to `Qwen2.5-72B-Instruct`; embeddings, prompts, tools, decoding, adapters, budgets, tasks, onsets, and seeds remain fixed. Table 5 summarizes the fixed-task recommendation change, while Appendix Table 12 prints every system's report card.

The paired winner changes from the learned reference to Zep, yet MemGPT/Letta undergoes the same three-rank demotion under both LLMs. The preserved object is the interaction-induced decision shift, independent of which revision-aware design occupies rank one.

4.5 Mechanism calibration and supporting checks

The internal controls provide a mechanism scale: naive retrieval falls from 49.8 at S5 to 45.2 at S20, with interference 4.3 and stale retrieval 18.6% on the S50 extension; the learned reference reaches S20 52.1, interference 0.6, and 18,090 tokens per episode (Appendix Table 10). The S20 ordering remains identical across the four onset regimes in Appendix Table 13. Five targeted exploit policies provide an additional fixed-policy comparison (Appendix Table 18).

5 Implications for Agent-Memory Evaluation

DriftPair changes the unit of evidence from an episode score to a state intervention. Benchmark construction must align branches at the session boundary and share the revision realization; otherwise stable carry benefit and revision-induced harm remain in one trajectory-level outcome. Reporting $\bar{I}_{20}(1)$ and $J_{m,20}$ alongside success reveals whether an architecture earns performance through reusable state, reconciliation after revision, or both.

The interaction also supplies an architecture-level diagnostic. Tiered and rolling summaries gain under stability yet incur the largest revision interactions; temporal supersession edges and verify–evict control preserve more of carry's value. This distinction predicts the observed rank shifts and makes leaderboard movement mechanistically interpretable. Recommendation claims can then separate success rank, exact-front membership, and selection regret.

The evaluated domain comprises text-based tool agents, controlled cross-session revisions, shared-budget memory policies, and verifier-backed task outcomes. Within this domain, the factorial, fixed-task, and cross-model analyses support the same interaction pattern and decision consequence. This agreement makes two reporting requirements practical for stateful-agent studies: branch-aligned carry effects and carry–revision interactions. Deterministic claim-to-source and numerical checks place Synthesis on the same primary-score footing as Coding, SQL, and Planning.

6 Conclusion

DriftPair identifies how revision changes the marginal value of pre-session memory. Stable carry benefits every representative policy, yet revision produces 4.7–5.2-point interactions for tiered and rolling summaries and 1.2–1.3 points for temporal and verify–evict designs. Those interaction differences become selection regret: independent sessions choose a system that falls three ranks under paired revision, and the demotion recurs across two base LLMs. Treating carried state as an intervention therefore connects memory mechanism to benchmark score and, ultimately, to the architecture a long-horizon evaluation recommends.

Ethics Statement

DriftPair evaluates text-based tool-agent behavior under controlled revisions to technical APIs, schemas, sources, and permissions. The protocol uses automated verifiers and reports resource costs alongside success and interference so that memory policies are compared under the same task, budget, and revision conditions.

Reproducibility Statement

Sections 2 and 4 specify branch alignment, task families, budgets, models, seeds, estimands, descriptive summaries, and decision rules. The appendix provides policy definitions, per-family and per-model report cards, and fixed-policy calibration checks.

AI Use Statement

Generative AI was used only for manuscript writing, editing, and presentation. It was not used to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims or proofs, refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean or reformat data, conduct qualitative or thematic analysis, or interpret results. The authors reviewed all AI-assisted work and take responsibility for the final content.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv preprint arXiv:2310.11511*, 2023. doi: 10.48550/arXiv.2310.11511. URL https://arxiv.org/abs/2310.11511.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling Large Language Models to Generate Text with Citations. *arXiv preprint arXiv:2305.14627*, 2023. doi: 10.48550/arXiv.2305.14627. URL https://arxiv.org/abs/2305.14627.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv preprint arXiv:2002.08909*, 2020. doi: 10.48550/arXiv.2002.08909. URL https://arxiv.org/abs/2002.08909.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. *arXiv preprint arXiv:2401.13919*, 2024. doi: 10.48550/arXiv.2401.13919. URL https://arxiv.org/abs/2401.13919.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. MemoryArena: Benchmarking Agent Memory in Interdependent Multi-Session Agentic Tasks. *arXiv preprint arXiv:2602.16313*, 2026. doi: 10.48550/arXiv.2602.16313. URL https://arxiv.org/abs/2602.16313.

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot Learning with Retrieval Augmented Language Models. *arXiv preprint arXiv:2208.03299*, 2022. doi: 10.48550/arXiv.2208.03299. URL https://arxiv.org/abs/2208.03299.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can Language Models Resolve Real-world GitHub Issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks. *arXiv preprint arXiv:2401.13649*, 2024. doi: 10.48550/arXiv.2401.13649. URL https://arxiv.org/abs/2401.13649.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2005.11401.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42330–42357, 2023. doi: 10.52202/075280-1835. URL https://doi.org/10.52202/075280-1835.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/e9df36b21ff4ee211a8b71ee8b7e9f57-Paper-Conference.pdf.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. *arXiv preprint arXiv:2401.13178*, 2024. doi: 10.48550/arXiv.2401.13178. URL https://arxiv.org/abs/2401.13178.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023. doi: 10.52202/075280-2019. URL https://doi.org/10.52202/075280-2019.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents. *arXiv preprint arXiv:2402.17753*, 2024. doi: 10.48550/arXiv.2402.17753. URL https://arxiv.org/abs/2402.17753.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A Benchmark for General AI Assistants. *arXiv preprint arXiv:2311.12983*, 2023. doi: 10.48550/arXiv.2311.12983. URL https://arxiv.org/abs/2311.12983.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. *arXiv preprint arXiv:2305.14251*, 2023. doi: 10.48550/arXiv.2305.14251. URL https://arxiv.org/abs/2305.14251.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023. doi: 10.48550/arXiv.2310.08560. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shihao Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bowen Li, Ziwei Tang, Jing Yi, Yuzhang Zhu, Zhenning Dai, Lan Yan, Xin Cong, Yaxi Lu, Weilin Zhao, Yuxiang Huang, Junxi Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Zhiyuan Liu, and Maosong Sun. Tool Learning with Foundation Models. *arXiv preprint arXiv:2304.08354*, 2023. doi: 10.48550/arXiv.2304.08354. URL https://arxiv.org/abs/2304.08354.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. *arXiv preprint arXiv:2112.01488*, 2021. doi: 10.48550/arXiv.2112.01488. URL https://arxiv.org/abs/2112.01488.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377. URL https://doi.org/10.52202/075280-0377.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. MemBench: Towards More Comprehensive Evaluation on the Memory of LLM-based Agents. *arXiv preprint arXiv:2506.21605*, 2025. doi: 10.48550/arXiv.2506.21605. URL https://arxiv.org/abs/2506.21605.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change. *arXiv preprint arXiv:2206.10498*, 2022. doi: 10.48550/arXiv.2206.10498. URL https://arxiv.org/abs/2206.10498.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in Multi-turn Interaction with Tools and Language Feedback. *arXiv preprint arXiv:2309.10691*, 2023. doi: 10.48550/arXiv.2309.10691. URL https://arxiv.org/abs/2309.10691.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802*, 2024. doi: 10.48550/arXiv.2403.18802. URL https://arxiv.org/abs/2403.18802.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. *arXiv preprint arXiv:2410.10813*, 2024. doi: 10.48550/arXiv.2410.10813. URL https://arxiv.org/abs/2410.10813.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. *arXiv preprint arXiv:2404.07972*, 2024. doi: 10.48550/arXiv.2404.07972. URL https://arxiv.org/abs/2404.07972.

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

Chenchen Ye, Ziniu Hu, Yihe Deng, Zijie Huang, Mingyu Derek Ma, Yanqiao Zhu, and Wei Wang. MIRAI: Evaluating LLM Agents for Event Forecasting. *arXiv preprint arXiv:2407.01231*, 2024. doi: 10.48550/arXiv.2407.01231. URL https://arxiv.org/abs/2407.01231.

Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, Yuandong Tian, and Jishen Zhao. AMA-Bench: Evaluating Long-Horizon Memory for Agentic Applications. *arXiv preprint arXiv:2602.22769*, 2026. doi: 10.48550/arXiv.2602.22769. URL https://arxiv.org/abs/2602.22769.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4410c0711e9154a7a2d26f9b3816d1ef-Paper-Conference.pdf.

A Related-work contrast on the identified estimand

Table 6: Estimand-level contrast. Branch alignment under a shared revision realization identifies the signed contribution of pre-session state and its change under drift.

Work family	Interdependent	Controlled revision	Branch-aligned carry	Evaluation object
End-task agent benchmarks	partial	incidental	no	Long-horizon task competence / tool use
MemBench / MemoryAgentBench	partial	no explicit axis	no	Memory use across turns
MemoryArena	yes	task-dependent	no	Interdependent multi-session action
AMA-Bench	trajectories	natural or scripted	no	QA over agent trajectories
DriftPair	yes	shared realization	yes	Revision-conditional carry effect and interaction

Interpretation. End-task, turn-level memory, and interdependent-session evaluations answer complementary questions about competence, retention, and trajectory behavior. DriftPair targets the conditional state effect: per-session branch alignment, a shared revision, equal budgets, policy identity, and common exogenous randomness yield $\bar{I}_{m,s}(d)$, while the factorial contrast yields $J_{m,s}$. Recommendation analysis then measures the cost of selecting architectures with an evaluation that omits this interaction.

B Notation, diagnostics, and baseline definitions

Notation. Table 7 collects the evaluation quantities used throughout the paper.

Table 7: Symbols, ranges, and operational definitions. Session s is the evaluation session; budget B selects the caps in Section 2.

Symbol	Range	Definition
$S^{a}_{m,k,s,r}(d)$	$[0, 100]$	Score for policy m, task k, session s, seed r, state access a, and revision condition d.
$I_{m,k,s,r}(d)$	$[-100, 100]$	Interference: $S^{\text{fresh}} - S^{\text{carry}}$; positive is harmful carry and negative is beneficial carry.
$\bar{I}_{m,s}(d)$	$[-100, 100]$	Mean interference over R seeds and K_s tasks under revision condition d.
$J_{m,s}$	$[-200, 200]$	Carry–drift interaction: $\bar{I}_{m,s}(1) - \bar{I}_{m,s}(0)$.
r_{stale}	$[0, 100]$	Stale retrieval rate (% of retrieved fragments).
c_{contam}	$[0, 1]$	Contamination score (fraction of episodes).
W_{eff}	$\mathbb{R}_+$ or undefined	Write efficiency: $\Delta S / \Delta\text{tokens}_{\text{written}}$ for $\Delta S > 0$; undefined otherwise.
s	$\mathbb{N}$	Session index; $K_{20} = 990$ and $K_{50} = 560$ for the extension.
B	–	Budget tier (small / medium / large).

Failure taxonomy. Table 8 links each failure category to an operational event in the branch-aligned execution trace.

Table 8: Failure taxonomy. Provenance pointers connect each category to the corresponding memory operation.

Failure type	Operational definition
Missing write	Agent fails later after not writing a reusable constraint or tool recipe.
Stale retrieval	A retrieved fragment contradicts newer verifier-passed evidence.
Contradiction ignored	Conflicting fragments are retrieved but the agent does not resolve or verify them.
Poison influence	An injected fragment changes action selection or tool arguments.
Budget misuse	Retrieval or verification budget is spent early and failure occurs later in the session.

Baseline specification. Table 9 distinguishes the memory behavior tested by each internal control under a shared harness.

Table 9: Baseline specification. The no-memory control receives the full current-session budget. All rows share models, prompts, tools, and budget tiers.

Baseline	Specification
No memory	Context-only; no `read`/`write` calls.
Naive retrieval	`bge-large-en-v1.5` + FAISS HNSW; top-32 retrieval; no eviction.
Summarization memory	Progressive LLM summarization; 4k-token summary budget; no raw retrieval.
Hybrid graph+vector	Typed entity–relation store (API symbol / schema / citation / planning-constraint entities with `supersedes` and `contradicts` edges) + vector retrieval; TTL eviction (72 h).
Learned reference (XGBoost controller)	Fixed XGBoost trigger/router selecting `read`/`write`/`summarize`/`evict` actions; diagnostics are excluded from the training target.
Shared harness	`Llama-3.1-70B-Instruct`, identical prompts, tool versions, embedding model, redaction rules, and budget tiers across all five baselines.

C Reference calibration report card

The calibration card separates session-20 comparisons over all four families from the longer Coding/Planning extension. The diagnostic table complements success and interference with retrieval staleness, contamination, and observed latency.

Table 10: Session trends and official axes at medium budget. S5 and S20 use 990 tasks; S50 uses the 560-task Coding/Planning extension. S20 and $\bar{I}_{20}(1)$ are five-seed means±SD where shown.

Policy	S5	S20	S50	$\bar{I}_{20}(1)$	Tokens/ep
No memory	44.2±1.4	43.8±1.6	43.4±1.7	0.0	17,280
Naive retrieval	49.8±1.3	45.2±1.8	41.8±2.1	4.3±0.9	18,540
Summarization memory	51.0±1.2	48.6±1.5	47.4±1.7	2.1±0.6	17,650
Hybrid graph+vector	52.3±1.2	50.6±1.4	49.6±1.5	1.6±0.5	19,120
Learned reference	53.2±1.1	52.1±1.3	50.9±1.4	0.6±0.4	18,090

Table 11: Diagnostics corresponding to Table 10.

Policy	Stale@S50	Contamination	Latency p50 (s)
No memory	0.0%	0.00±0.00	28.4
Naive retrieval	18.6%	0.24±0.05	34.2
Summarization memory	13.2%	0.14±0.04	33.1
Hybrid graph+vector	10.8%	0.11±0.03	37.6
Learned reference	8.4%	0.08±0.03	36.1

D Second-base-model report card

Changing the base model exchanges Zep and the learned reference at the top of the paired ranking while preserving MemGPT/Letta's fourth-place position. The full triples below also show why summarization and no memory remain on the first front despite their lower success scores: their token costs are below those of the two leading systems.

Table 12: `Qwen2.5-72B-Instruct` replication on 990 tasks, five seeds, and fixed onset.

System	S20	$\bar{I}$	Tokens/ep	Independent rank	Paired rank	Front
Zep v0.21	53.0	0.8	18,950	4	1	Y
Learned reference	52.8	0.9	17,860	3	2	Y
Hybrid graph+vector	51.5	1.4	18,810	5	3	N
MemGPT / Letta	50.4	2.2	19,740	1	4	N
Summarization memory	49.6	2.0	17,380	6	5	Y
LangChain SB	49.1	2.5	17,620	2	6	N
LlamaIndex	47.2	3.6	18,410	8	7	N
Naive retrieval	46.7	4.0	18,200	7	8	N
No memory	45.1	0.0	16,930	9	9	Y

Zep or the learned reference occupies the highest-success point on every one of five seeds, and Zep retains first-front membership in 4/5 seed-level reports.

E ONSET-REGIME STABILITY

Schedule robustness. Table 13 prints all four onset regimes on the three official axes.

Table 13: Report cards across the four drift-onset regimes at medium budget and session 20, with tokens/ep (Tok) shown so the Pareto rule *(S20↑, $\bar{I}$ ↓, tokens↓)* is visible under every regime. The printed method rows are five-seed means.

	Fixed			Early			Late			Randomized		
Method	S20	$\bar{I}$	Tok	S20	$\bar{I}$	Tok	S20	$\bar{I}$	Tok	S20	$\bar{I}$	Tok
Naive vector retrieval	45.2	4.3	18.5k	44.1	5.2	18.6k	46.7	3.4	18.4k	45.0	4.5	18.5k
Summarization memory	48.6	2.1	17.7k	48.1	2.6	17.7k	49.8	1.7	17.6k	48.6	2.3	17.7k
Hybrid graph+vector	50.6	1.6	19.1k	50.2	1.9	19.2k	51.3	1.3	19.0k	50.6	1.7	19.1k
Learned reference	52.1	0.6	18.1k	51.7	1.2	18.2k	52.5	0.8	18.0k	52.0	1.1	18.1k

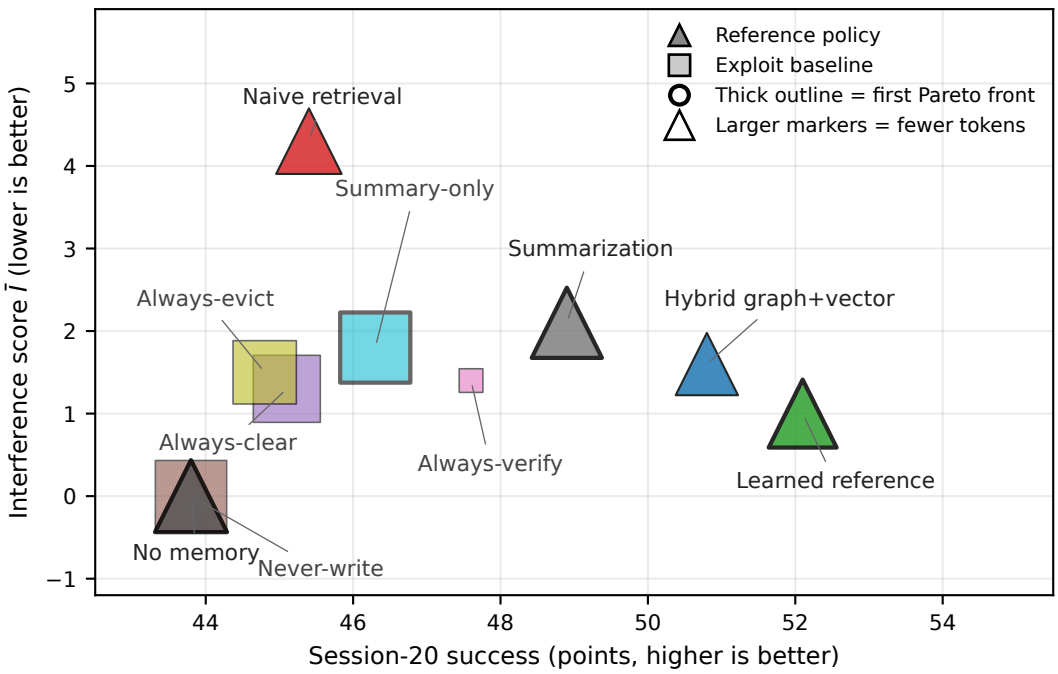

Figure 2: Session-20 Pareto calibration over success, interference, and tokens/episode. The learned reference is the highest-success non-dominated point; targeted exploit policies reveal the tradeoff incurred by optimizing one axis.

F Instrumentation overhead details

The three logging modes expose the cost of progressively richer diagnostics. Full logging retains attribution context at a small latency cost, while counterfactual replay increases both retained log size and the observed overhead.

Table 14: Instrumentation overhead details. Minimal mode supports aggregate scoring; full mode retains failure-audit context with 3.4% p95 latency overhead, below the $\leq$ 10% design target. Counterfactual replay retains additional decision-state context. Per-episode log size and latency quantify the observer-effect cost of each mode.

Mode	Latency overhead p50/p95	Token overhead/ep	Log size/ep (KB)
Minimal	+0.3% / +0.8%	+72 $\pm$ 18	14.6 $\pm$ 4.2
Full (default)	+1.1% / +3.4%	+196 $\pm$ 48	52.3 $\pm$ 16.1
Full + counterfactual replay	+2.3% / +6.2%	+248 $\pm$ 62	64.8 $\pm$ 19.4

G External memory systems under the same harness

Systems and adapter protocol. The default panel evaluates LlamaIndex VectorMemory v0.11.5 (top-k cosine retrieval), LangChain SummaryBuffer v0.3.1 (4k-token rolling summary), MemGPT/Letta v0.5.2 (tiered core, recall, and archival memory with flushing at 85% context pressure), and Zep v0.21 (timestamped temporal graph with 72 h TTL). Thin four-call adapters preserve each policy's memory behavior under shared prompts, models, embeddings, caps, redaction, and logging. The sweep covers five seeds, 990 tasks, paired branches, and four onset schedules.

Table 15: Default external systems under the DriftPair harness (medium budget, fixed onset, four families, five seeds). S20 and $\bar{I}_{20}(1)$ are reported on the same official task-weighted scale used for ranking.

System	S20	$\bar{I}_{20}(1)\downarrow$	Tokens/ep↓	Stale↓	Exposure↓	Pareto
LlamaIndex VectorMemory v0.11.5	46.4±1.6	3.9±0.8	18,760	16.4%	0.21±0.05	no
LangChain SummaryBuffer v0.3.1	48.3±1.5	4.5±0.7	17,940	12.1%	0.13±0.04	no
MemGPT / Letta v0.5.2	49.7±1.4	3.9±0.7	20,160	10.4%	0.12±0.04	no
Zep v0.21 (temporal KG)	51.9±1.2	0.7±0.4	19,320	8.9%	0.09±0.03	no
Learned reference (internal)	52.1±1.3	0.6±0.4	18,090	8.4%	0.08±0.03	yes

H SINGLE-SESSION NO-CARRY COMPARATOR: CONTROLLED RECOMMENDATION SHIFT

Protocol. The independent-session condition clears memory between sessions and disables tool-schema renames, contradiction insertion, and constraint flips. Task distribution, verifiers, caps, prompts, tool versions, seeds, and the nine-system universe match DriftPair. The five internal policies and four external systems use the medium budget and five seeds.

Table 16: Fixed-task independent-session comparison against DriftPair (medium budget, five seeds). MemGPT/Letta moves from first to fourth as stale-summary carry becomes active.

System	Independent S20	Independent rank	DriftPair S20	DriftPair rank	Mechanism
MemGPT / Letta v0.5.2	55.2±1.1	1	49.7±1.4	4	stale summary carry
LangChain SummaryBuffer v0.3.1	54.1±1.2	2	48.3±1.5	6	override dilution
Learned reference (internal)	53.0±1.0	3	52.1±1.3	1	drift-aware verify/evict unused
Zep v0.21 (temporal KG)	52.8±1.1	4	51.9±1.2	2	graph edges unused
Hybrid graph+vector (internal)	52.2±1.2	5	50.6±1.4	3	graph overhead without drift
Summarization memory (internal)	51.9±1.3	6	48.6±1.5	5	context compression still helps
Naive retrieval	51.1±1.3	7	45.2±1.8	8	contradictions disappear
LlamaIndex VectorMemory v0.11.5	50.7±1.4	8	46.4±1.6	7	default top-k mirrors naive retrieval
No memory (context-only)	44.3±1.5	9	43.8±1.6	9	no retrieval budget either way

Interpretation. MemGPT/Letta's tiered memory and summary flushing lead when sessions are independent, while the same carry policy falls behind revision-aware designs once earlier summaries can become stale. Two systems move by at least three ranks, three move by two, three move by one, and no memory remains ninth. The fixed system and task universe isolate a controlled recommendation comparison; the factorial analysis in Table 3 identifies the carry–drift mechanism.

I BENCHMARK INTEGRITY / ANTI-GAMING EVIDENCE MATRIX

Table 17: Evidence matrix for the retained identification and decision claims. Each row links a controlled comparison to its manuscript evidence.

Object / threat	Trust boundary	Property / comparator	Benign cost	Failure / evidence
Carried state / stale or contradictory memory enters a revised session	Matched branches fix task, environment, prompts, tools, model, budget, drift, and policy	Marginal carry effect; 2×2 carry-by-drift and independent-session control	One additional scored branch	Stale-summary interactions reach 4.7–5.2 points; Table 3
Architecture selection / an evaluator omits carried-state revision	The same task universe, system universe, and seeds are compared under independent and paired conditions	Selection regret in S20, interference, and tokens/episode	A second evaluator condition	MemGPT/Letta moves from rank 1 to 4; Table 4
Base-model dependence / one LLM induces the reversal	Every non-LLM component remains fixed	Independent and paired selections under Llama-3.1-70B and Qwen2.5-72B	One second-model sweep	The MemGPT/Letta demotion is retained; Table 5

Evidence interpretation. The matrix makes the retained evidence path explicit: a branch-aligned state intervention identifies the carry–revision interaction, a fixed-task comparison exposes the selection consequence, and a base-model replication tests whether the demotion persists.

J TARGETED EXPLOIT POLICIES

The targeted policies isolate extreme memory-operation choices. Their status below compares each measured triple with the official nine-system panel: the no-memory control dominates never-write through lower token cost at equal success and interference. The learned reference dominates the clear, verify, and evict policies on all three axes, while summary-only retains a token–interference trade-off.

Table 18: Exploit-policy audit against the official nine-system panel at session 20 with fixed onset and medium budget. Summary-only is non-dominated; never-write loses 8.3 S20 points, always-verify adds 5.7k tokens, and the learned reference is the highest-success first-front point.

Exploit policy	S20	$\bar{I}\downarrow$	Stale↓	Exposure↓	Tokens/ep↓	Pareto status
Never-write (no memory carry-over)	43.8±1.6	0.0	0.0%	0.00	17,420	dominated by no memory
Always-clear-after-drift (purge at $t^\star$)	45.1±1.7	1.3	2.1%	0.04	18,260	dominated by learned ref.
Always-verify (re-verify every retrieval)	47.6±1.5	1.4	6.2%	0.05	23,810	dominated by learned ref.
Always-evict (aggressive TTL 12 h)	44.8±1.7	1.5	7.8%	0.06	18,940	dominated by learned ref.
Summary-only (no raw retrieval)	46.3±1.6	1.8	9.4%	0.09	17,650	yes (non-dominated)
Learned reference policy	52.1±1.3	0.6	8.4%	0.08	18,090	yes (non-dominated)