Skill2Mem: Probing Skill-to-Memory Transfer in LLM Agent Memory

Anonymous ACL submission

Abstract

Long-term memory is becoming a core substrate for LLM agents, but most memory systems still rely on hand-written procedures for deciding what to store, update, retrieve, or ignore. Meanwhile, public agent ecosystems distribute reusable skills that encode execution-time procedures. We study whether such skills can be transferred, without training, into memory systems as compact retrieval-time priors. Skill2Mem adapts public skills into memory-facing priors and injects a small selected set into the retrieval planner of a strong memory baseline under a fixed budget. This controlled transfer interface separates direct execution utility from memory alignment. Across LongMemEval and consistency checks, Skill2Mem reveals an alignment-sensitive transfer boundary: curated public priors can induce structured negative transfer, especially on memory-heavy queries, while selector-strength, bank-size, gating, and k/budget controls show that the pattern is not reducible to weak selection or to the generic cost of adding text. The contribution is a reusable diagnostic protocol for identifying when public procedures become misaligned memory priors for LLM agent memory.

1 Introduction

Long-term memory has become a central design axis for LLM agents. Benchmarks such as LoCoMo and LongMemEval show that sustained, multi-session interactions require more than larger context windows; systems must decide *what* to store, *how* to consolidate it, and *when* to retrieve or abstain (Maharana et al., 2024; Wu et al., 2024). Recent systems improve these steps with structured storage, hierarchical memory, reflection, or operating-system metaphors (Chhikara et al., 2025; Kang et al., 2025; Tan et al., 2025; Rasmussen et al., 2025; Liu et al., 2026). Yet a common bottleneck remains: most memory pipelines still hard-code a small set of human-written procedures for memory extraction, update, and retrieval.

At the same time, public agent ecosystems distribute *skills* as reusable procedural resources. SkillsBench shows that curated skills can substantially improve agent pass rates, while self-generated skills often fail to help; notably, focused skills with 2–3 modules outperform comprehensive documentation (Li et al., 2026). These findings motivate a transfer hypothesis: public skill collections may encode procedural priors that are useful beyond their original execution setting.

This paper studies whether such priors transfer to memory. A reasonable but untested hypothesis is that skills useful for downstream execution may also inform upstream memory procedures. We formalize this question as *skill-to-memory transfer*: adapt a public agent skill into a compact prior over memory operations, retrieve a small set of relevant priors at test time, and use them to condition a memory pipeline. This question follows recent work that treats procedural memory as a distinct design target for agents (Fang et al., 2025; Zhang et al., 2026). Existing procedural-memory systems typically learn skills through reinforcement learning and self-evolution (Zhang et al., 2026; Yang et al., 2026, 2025), or target execution generalization (PolySkill, 2025). Skill2Mem isolates a train-free regime of *test-time transfer* without learning a controller or updating model parameters.

We instantiate this question through **Skill2Mem**, a controlled study design in which each public skill is adapted into a compact mem-

ory prior with four fields—trigger, write focus, update rule, and retrieval cues—and a selector injects a small top-k set into a memory system's *retrieval planner*. Direct skill use preserves task-specific execution context. Skill2Mem instead keeps a short memory-facing cue that must help the planner decide which evidence to retrieve. The setup therefore tests interface alignment under compression.

Our base system is SimpleMem (Liu et al., 2026), a recent train-free system that combines semantic structured compression, online semantic synthesis, and intent-aware retrieval planning. We use MemSkill (Zhang et al., 2026) as a conceptual anchor because it explicitly frames memory operations as skills, but we do not inherit its training-heavy controller-designer pipeline.

The study reveals a useful transfer boundary rather than a leaderboard-style gain. In the TF–IDF selector run, curated public priors produce the clearest LongMemEval degradation, reducing judge accuracy by 2.7 points on the shared 300-question subset. In an embedding-selector diagnostic run on the same subset, curated, size-matched noisy, and MemSkill priors tie the no-skill baseline under judge accuracy but remain below generic and random controls, and curated remains below baseline under EM. Confidence gating reduces exposure without restoring the baseline, while smaller k partially mitigates the default setting. Skill2Mem therefore turns a broad transfer hypothesis into a precise diagnostic claim: public skills can be useful procedural resources, but a compressed memory-prior interface requires alignment with retrieval-time memory operations.

Contributions.

- We define *skill-to-memory transfer* for LLM agent memory: adapting public execution skills into train-free, test-time priors over memory operations.
- We introduce Skill2Mem as a diagnostic protocol built on a strong memory baseline (SimpleMem), with controls over selector strength, bank composition, confidence gating, and k/budget sensitivity.
- We identify an alignment-sensitive transfer boundary: execution utility does not by itself guarantee memory alignment, and the resulting mismatch is visible through selector concentration, asymmetric flips, and case-level retrieval behavior.

2 Background and Related Work

2.1 Memory systems for LLM agents

Long-term memory benchmarks have highlighted persistent weaknesses of both long-context models and naive retrieval pipelines. LoCoMo introduced very long multi-session dialogues with long-range temporal and causal dependencies (Maharana et al., 2024). LongMemEval decomposes long-term interactive memory into information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention (Wu et al., 2024). MemoryAgentBench broadens the evaluation space toward accurate retrieval, test-time learning, long-range understanding, and selective forgetting in incremental multi-turn settings (Hu et al., 2025).

System designs vary widely. Mem0 dynamically extracts, consolidates, and retrieves salient information, including a graph variant for relational structure (Chhikara et al., 2025). Zep/Graphiti emphasizes a temporal knowledge-graph substrate for dynamic enterprise memory (Rasmussen et al., 2025). Reflective Memory Management combines prospective multi-granularity summarization with retrospective retrieval refinement (Tan et al., 2025). Memory OS of AI Agent introduces a hierarchical storage and update architecture inspired by operating systems (Kang et al., 2025), while MemOS extends the OS metaphor into a broader memory-centric system framework spanning plaintext, activation, and parameter memories (Li et al., 2025). Among train-free systems, SimpleMem is especially relevant because it achieves a strong performance-efficiency frontier via semantic compression, online synthesis, and intent-aware retrieval planning (Liu et al., 2026).

2.2 Skills, procedural memory, and self-evolving agents

The procedural side of agent memory is receiving increasing attention. Memp studies lifelong procedural memory built from trajectories and distinguishes fine-grained instructions from higher-level script-like abstractions (Fang et al., 2025). MemSkill goes one step further by treating extraction, consolidation, and pruning routines as learnable and evolvable memory skills, se-

lected by a controller and refined by a designer (Zhang et al., 2026). AutoSkill similarly proposes experience-driven lifelong learning through self-evolving skill repositories and standardized skill representations (Yang et al., 2026). H2R uses hierarchical hindsight reflection to separate planning-level and execution-level memory in multi-task agents (Ye et al., 2025), and MUSE organizes experience in a hierarchical memory module for self-evolving long-horizon productivity agents (Yang et al., 2025).

These works are conceptually close to ours but methodologically different. Most optimize or evolve skills through training, reflection loops, or reinforcement learning. Skill2Mem instead isolates a train-free transfer setting: reusing existing external skill resources as *test-time priors* for memory operations without updating the model or learning a controller.

We also note that reinforcement-learning-centric agent frameworks (e.g., OpenClaw-RL) represent a complementary, training-heavy regime for accumulating procedural competence (Wang et al., 2026); Skill2Mem instead focuses on test-time transfer of already-public skills.

2.3 Agentic skills as reusable external priors

Public agent skills provide an external source of procedural knowledge. SkillsBench provides the most direct empirical motivation for our work: across 84 evaluated tasks in 11 domains, curated skills improve average pass rate by 16.2 percentage points, self-generated skills do not help on average, and focused 2–3-skill configurations outperform larger collections and comprehensive documentation (Li et al., 2026). PolySkill adds another key insight: separating a skill's abstract goal from its concrete realization can improve generalization and reuse across websites and tasks (PolySkill, 2025).

These findings motivate a narrower question: when execution-oriented procedural knowledge is compressed into a short memory cue, does any of its utility survive as guidance for memory operations? Our work evaluates that interface directly rather than assuming that execution-time gains transfer unchanged.

2.4 Survey perspectives

Recent surveys reinforce the timeliness of this question. Work on memory for LLM-based agents argues for understanding memory through representation types and atomic operations such as consolidation, updating, indexing, forgetting, retrieval, and compression (Du et al., 2025). "AI Meets Brain" connects cognitive-neuroscience perspectives to LLM agents and explicitly points to skill acquisition as a future direction for memory systems (Liang et al., 2025). Our paper operationalizes this survey-level intuition into a concrete empirical study. Appendix J provides a more detailed positioning table that contrasts train-free memory systems, procedural-memory skill learning, and public skill ecosystems.

3 Problem Setting and Research Questions

We study a memory agent that receives a stream of interactions $H = (u_1, \ldots, u_T)$ and later answers a query q using an external memory store. Existing systems typically rely on a small hand-designed set of write/update/retrieve heuristics. We ask whether public procedural skills can serve as external *memory priors*: compact, test-time guidance that helps the memory system choose what evidence to retrieve and how to reason over it.

Let $\mathcal{B}$ denote a skill bank consisting of public agent skills or memory-native skills. Each skill is transformed into a compact memory prior $p = A(s)$ containing four operational fields: **trigger** (when it is relevant), **write focus** (what to preserve), **update rule** (how to revise under conflict/updates), and **retrieval cues** (how to search or filter memory). Given a query context q, Skill2Mem retrieves top-k priors under a fixed token budget and conditions the memory pipeline on them. In this study, we restrict the intervention to retrieval planning (retrieve-only), leaving write-side memory construction unchanged.

This scope matters. Direct skill use can rely on task-specific execution context, whereas Skill2Mem compresses a skill into a short procedural prior that must help a memory planner decide what evidence to retrieve. This compression creates a controlled interface for testing transfer, but it can also discard context and over-generalize broad skills into misleading memory guidance. We therefore cast the paper around three research questions:

RQ1: Under a fixed token budget, how do curated public skills compare with `base`, `generic`, and `random` controls as retrieval-time memory priors?

RQ2: Does the evidence localize any degradation to a *memory-alignment gap*, rather than to the mere addition of extra text or a weak selector?

RQ3: Which memory abilities are most sensitive to misaligned external priors in the retrieve-only interface?

4 Method

Skill2Mem is a controlled diagnostic interface for testing whether public skills transfer into memory as *test-time priors*. It consists of an adapter, a selector, and a retrieval-time intervention, while the underlying memory store and write-side processing remain fixed. This separation lets the experiment attribute changes in behavior to the injected prior interface rather than to a different memory construction pipeline. Figure 1 summarizes the study design.

4.1 Adapting skills into memory priors

Let $\mathcal{B}$ be a skill bank and $s \in \mathcal{B}$ a skill. Skill2Mem maps each s to a *memory prior* $p = A(s)$ with four operational fields: **trigger** (when relevant), **retrieval cues** (what to query/filter), **update rule** (how to revise under conflict or updates), and **write focus** (what should have been preserved at write time). The adapter is a normalization step. Its role is to convert heterogeneous skills into a comparable memory-facing representation so that the study tests *transfer* rather than raw document length or formatting. Appendix C specifies the representation details.

Direct invocation versus memory-prior injection. A public skill can be useful when used directly because it carries task-specific procedural context. Skill2Mem removes much of that context and keeps compact memory-facing guidance. For example, a skill about completing a multi-step task may be compressed into a prior about retrieving prerequisite state and recent evidence. Such a prior may be sensible for task-execution memory, but misleading for a personal long-term-memory question about dates or preferences. The study evaluates whether the compressed prior is aligned with memory operations.

Budget control. To separate "procedural alignment" from "adding more text", we enforce a fixed total prior-token budget B (384 tokens). For a top-k set of priors, we allocate at most $\lfloor B/k \rfloor$ tokens per prior and truncate the final concatenated block to B tokens. This removes length advantages from longer source skills (details in Appendix C).

4.2 Selecting priors under a fixed budget

Given a query q, we retrieve top-k adapted priors by matching over the adapted prior text. We intentionally keep k small to limit conditioning overhead and to mirror SkillsBench's finding that a few focused skills can outperform comprehensive documentation (Li et al., 2026). The main selector is TF–IDF, chosen as a transparent default; a local embedding selector is used as a stronger diagnostic control to separate selector weakness from memory-prior misalignment. The reported selector settings are specified in Appendix A.

Formal view. Let $\mathcal{P}_{\mathcal{B}}$ denote the adapted prior set from bank $\mathcal{B}$, and let $\mathrm{sim}(q, p)$ be the selector similarity. Skill2Mem selects $P_k(q) = \mathrm{TopK}_{p \in \mathcal{P}_{\mathcal{B}}} \mathrm{sim}(q, p)$ and injects a formatted block $I(P_k(q), B)$ into the retrieval planner, where $I(\cdot, B)$ enforces the fixed token budget B. We view this as supplying a test-time procedural prior to the planner without changing the underlying model or write-side memory.

4.3 Intervention scope: retrieval planning only

We inject priors into retrieval planning because this stage determines *which* memory slices are searched and *how* evidence is aggregated. This is a natural point of intervention for SimpleMem, which explicitly includes intent-aware retrieval planning (Liu et al., 2026). We condition the planner with the selected prior set and leave write-side processing unchanged. This restricted intervention isolates retrieval-time priors from changes to memory construction.

5 Experimental Setup

The experimental design focuses on the comparisons needed to interpret skill-to-memory transfer; Appendix A and Appendix I provide the full protocol and reproducibility details.

Benchmarks. We use LongMemEval as the primary benchmark because it targets the memory abilities most exposed to retrieval-time priors: information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention (Wu et al., 2024; Feng et al., 2024). We

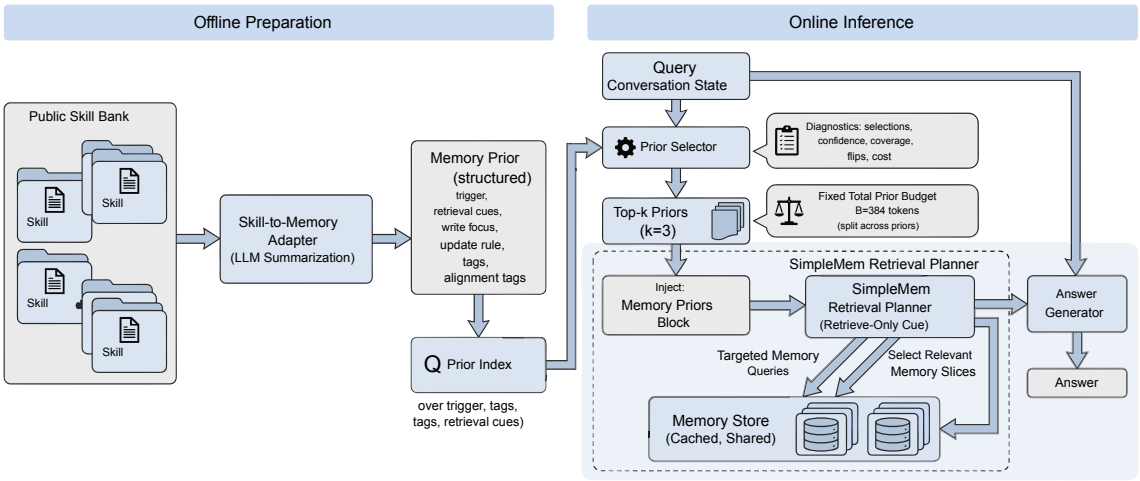

Figure 1: Skill2Mem study design. Public skills are adapted into compact memory priors (trigger, write focus, update rule, retrieval cues). At test time, a selector retrieves top-k priors under a fixed token budget and conditions SimpleMem's retrieval planner (retrieve-only), while the underlying memory construction remains unchanged.

use LoCoMo as a secondary consistency check because it stresses very long conversational memory with event and temporal structure in a different interaction format (Maharana et al., 2024).

Baseline and controls. The literature baseline is SimpleMem (Liu et al., 2026), used without external priors. Controlled variants keep the same retrieve-only intervention and change only the injected prior block: a budget-matched generic guideline, random priors from the same bank, curated public priors adapted from a curated public skill collection, noisy public priors adapted from a broader community collection, and memory-native priors adapted from MemSkill (Zhang et al., 2026). Bank composition and licensing are summarized in Appendix B.

Selector and bank controls. To test whether the observed transfer boundary is tied to a weak selector, the protocol treats selector strength as an explicit experimental factor. The additional diagnostic evidence uses local embedding retrieval as the stronger selector. To address bank-size and domain confounds, we include a fixed-seed `noisy_size18` control sampled from the broader noisy bank to match the curated bank size. These controls are not new memory algorithms; they are diagnostic probes for separating selection weakness, bank/domain mismatch, and the retrieve-only memory-prior interface.

Evaluation. All conditions keep write-side memory construction fixed and vary only the retrieval-time prior block. For LongMemEval, we report judge accuracy as the primary metric and exact match (EM) as a stricter secondary metric. For LoCoMo, we report the official category-aware QA score. The appendix reports paired uncertainty estimates, flip analyses, and efficiency statistics.

6 Results

6.1 Main results on LongMemEval

Table 1 addresses the first alternative explanation: if public procedures transfer cleanly into memory priors, curated skills should outperform a no-skill planner and simple text controls under the same retrieval budget. The shared 300-question LongMemEval subset shows a different boundary. Curated public priors reduce judge accuracy from 0.760 to 0.733, a 2.67-point drop relative to `base`, while random and noisy controls remain effectively tied with the baseline.

This table localizes the failure of naive transfer. Curated public priors do not outperform `base`, `generic`, or `random`; at the same time, to-

Condition	Judge	Δ (pp)	95% CI	EM	Pos/Neg flips
SimpleMem (Liu et al., 2026)	0.760	+0.00	—	0.247	—
Generic guideline	0.747	−1.33	[−4.00, +1.33]	0.233	6/10
Random priors	0.763	+0.33	[−2.33, +3.00]	0.240	8/7
Curated public priors	0.733	−2.67	[−5.33, +0.00]	0.243	5/13
Noisy public priors	0.763	+0.33	[−1.67, +2.33]	0.260	5/4
MemSkill priors (Zhang et al., 2026)	0.740	−2.00	[−4.33, +0.33]	0.243	4/10

Table 1: TF–IDF LongMemEval results on the shared 300-question evaluation subset. Judge deltas and 95% paired-bootstrap intervals are computed relative to `base`; flips count incorrect→correct / correct→incorrect transitions. Detailed setup and efficiency statistics are reported in the appendix.

ken budget alone does not explain the pattern, because `generic` trails `base` whereas `random` and `noisy` remain effectively tied with it. EM is flatter than judge accuracy, so the core signal is not uniform metric collapse but a specific pattern of judge-side degradation, asymmetric flips, and unstable retrieval behavior under curated priors.

These results are most useful as a diagnosis rather than as a standalone performance comparison. The interval estimates remain wide at this scale, but the qualitative pattern is consistent: curated shows the largest average drop, the most negative flip balance, and the strongest concentration in a few execution-oriented priors. Section 7 develops this pattern into a mechanism-level diagnosis.

6.2 LoCoMo as a consistency check

LoCoMo bounds the generality of the LongMemEval pattern. On the 200-question evaluation subset, all retrieve-only variants remain within 0.005 of the SimpleMem baseline (0.338), including curated public priors at 0.338. We therefore use LoCoMo as a consistency check rather than as the primary mechanism claim: it does not provide stable positive-transfer evidence at this scale, and it also does not reproduce the LongMemEval degradation. The full table is reported in Appendix H.

6.3 Selector-strength controls

A central alternative explanation is that the LongMemEval degradation is merely a consequence of weak selection. We therefore separate the TF–IDF result above from an embedding-selector diagnostic run, which evaluates the same shared 300-question LongMemEval subset with a local embedding selector. Table 2 reports this diagnostic run. The backbone model differs from the TF–IDF run, so the table is used for within-run stress testing rather than direct numerical comparison across selectors.

The stronger-selector stress test narrows the selector-only explanation. Curated, size-matched noisy priors, and MemSkill all tie `base` under judge accuracy but remain below `generic` and `random`; under EM, curated and `noisy_size18` are below `base`. This reduces the plausibility of a pure TF–IDF-only explanation while also showing that selector strength changes the magnitude and metric profile of the effect.

6.4 Mitigation and sensitivity

We next test whether the observed pattern can be mitigated without changing the memory system. Confidence-gated injection uses the curated top-1 embedding score and injects priors only above a threshold. The thresholds are selected from the always-inject score distribution to target approximately 75%, 50%, and 25% coverage.

These mitigation and sensitivity results identify actionable operating points rather than a universal fix. Confidence gating mainly buys lower exposure and lower token cost; it does not reliably erase the default curated loss. By contrast, reducing k to one prior changes the outcome, suggesting that failure is not caused by public priors alone but by the interaction among selector confidence, number of injected priors, and the retrieve-only planner interface.

7 Analysis

The diagnostic signal is concentrated, not diffuse. Figure 2 gives the main quantitative diagnostic from the TF–IDF run; Tables 2 and 3 provide the embedding-selector follow-up. The picture is parameter-sensitive rather than uniformly harmful. Under embedding retrieval, curated priors have full coverage and lower EM than `base`; confidence gating cuts coverage to 75%, 50%, and 25%, but EM remains 0.240–0.247 rather than recovering the no-skill baseline (0.257). Concen-

Condition	Judge	EM	Mean tokens	Top-1 score	Top-1 conc.
Base	0.700	0.257	4777	–	–
Generic guideline	0.747	0.253	5690	–	–
Random priors	0.727	0.257	5545	–	0.087
Curated public priors	0.700	0.247	5544	0.127	0.217
`noisy_size18`	0.700	0.243	5568	0.150	0.260
MemSkill priors	0.700	0.250	5582	0.159	0.293

Table 2: Embedding selector-strength diagnostic on the shared 300-question LongMemEval subset using local embedding retrieval (`sentence-transformers/all-MiniLM-L6-v2`) and `gpt-5.4`. All non-base rows use retrieve-only injection with k=3 and B=384. Top-1 concentration is the share of queries assigned to the most frequent top-1 prior.

Policy	Coverage	EM	Mean prior tok.
Always inject	1.00	0.247	384
Gate q25	0.75	0.240	288
Gate q50	0.50	0.247	192
Gate q75	0.25	0.240	97

Table 3: Confidence-gated curated-prior injection on the embedding-selector run. Gating reduces token exposure; the no-skill baseline EM is 0.257.

Variant	Judge	EM	Mean tokens
k=1, B=384	0.733	0.260	5590
k=3, B=384	0.700	0.247	5544
k=5, B=384	0.693	0.237	5573
k=3, B=192	0.703	0.260	5214
k=3, B=768	0.707	0.257	6351

Table 4: Small curated-prior sensitivity matrix on the same embedding-selector setup. The default k=3, B=384 is not the only operating point: reducing to one selected prior improves judge accuracy, while increasing to five priors worsens it.

tration is the complementary failure mode: the most frequent curated top-1 prior covers 21.7% of queries, while `noisy_size18` and MemSkill are even more concentrated (26.0% and 29.3%). Higher selector confidence and concentration therefore do not imply better memory alignment.

A representative failure case. A representative LongMemEval case makes the mechanism concrete. The query asks for a temporal comparison; without priors, the planner retrieves enough support, while curated public priors steer retrieval toward broad execution cues and lead to abstention. The degradation is visible at the selector and planner level, not only in aggregate scores.

The drop concentrates in memory-heavy question types. The TF–IDF breakdown gives a specific answer to RQ3. The curated drop is concentrated in multi-session and temporal reasoning rather than spread uniformly (Appendix E). Curated falls from 0.671 to 0.605 on multi-session questions and from 0.630 to 0.593 on temporal reasoning, while knowledge-update questions remain stable or improve (0.848 to 0.870). This points to retrieval-planning mismatch: cross-session aggregation and time normalization are the abilities most exposed to bad procedural priors at retrieval time.

Evidence for an alignment gap. Within the current retrieve-only interface, the evidence favors an *alignment-gap* explanation over a simple "more text hurts" explanation. Same-budget generic text can outperform base, random priors can be competitive, and curated priors improve when k is reduced to one. The evidence is conditional: the default curated setting (k=3, B=384) underperforms base/random on EM and fails to beat generic/random under judge accuracy; size-matched noisy priors do not solve the problem; and confidence gating reduces exposure without restoring the baseline. Execution-oriented public skills therefore require selector, bank, and interface alignment before they can be treated as memory procedures.

8 Discussion

Scope of the contribution. Skill2Mem tests whether public execution skills transfer into retrieval-time memory priors. In the studied setting—SimpleMem, retrieve-only injection, fixed-budget prior blocks, and controlled subsets—the results support a conditional diagnostic claim rather than a performance claim. Naive public-skill-to-memory transfer is brittle and alignment-sensitive: stronger local retrieval changes the effect without making curated public priors a stable advantage. The pattern is inspectable through selected skills, question types, and planner behavior.

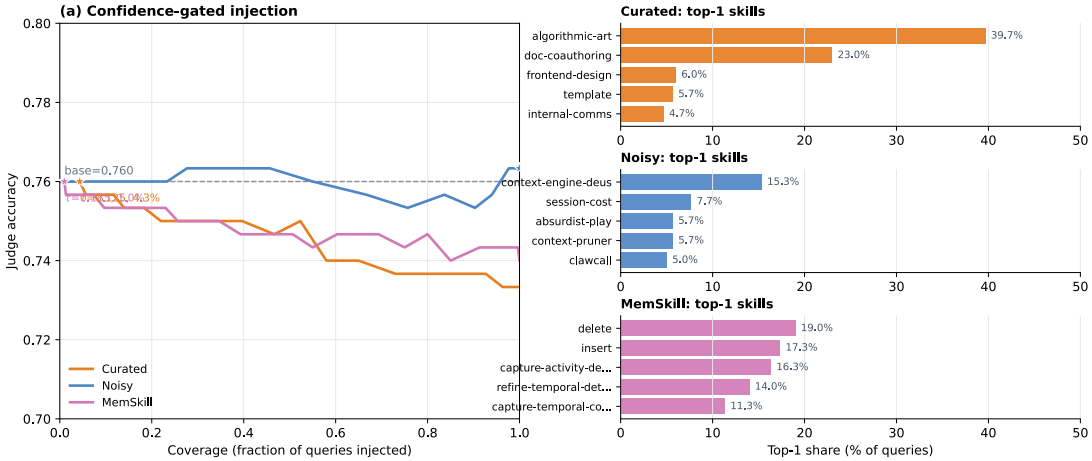

Figure 2: TF–IDF quantitative diagnosis on LongMemEval-300. Left: confidence-gated injection shows the coverage–accuracy tradeoff. Right: frequent top-1 prior categories. High selector concentration and confidence do not guarantee memory alignment; embedding-selector results are in Tables 2 and 3.

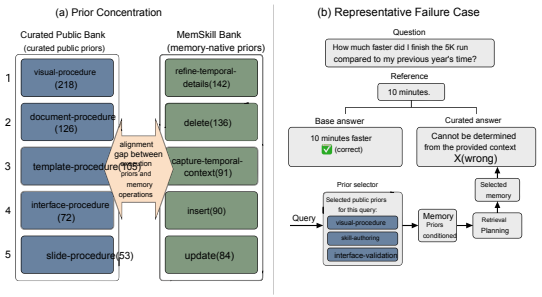

Figure 3: Representative alignment-gap mechanism. Public execution priors redirect a temporal-comparison query away from the needed memory operation, turning an answerable case into abstention.

Claims outside scope. The evidence concerns compressed public-skill priors, not direct skill use or learned memory controllers. The `noisy_size18` control reduces the bank-size confound but is not a domain-matched memory-prior bank. Write-time normalization, update policies, learned controllers, and broader public-skill curation remain outside the tested intervention.

Implications for future systems. External priors remain attractive because they are explicit, controllable, and train-free, but they require *memory-aware alignment*. Stronger selectors, confidence gating, size-matched controls, and small k/budget ablations help separate selector weakness, bank composition, and interface mismatch. Skill2Mem is thus a controlled setting for studying transfer, not evidence that larger public-skill banks should be injected by default.

9 Conclusion

Skill2Mem establishes skill-to-memory transfer as a diagnostic axis for LLM agent memory. The central finding is not that public skills are a poor resource; it is that execution-time utility must be aligned with memory operations before public procedures can serve as retrieval-time priors. In the studied retrieve-only, fixed-budget interface, curated public priors change planner behavior but can mismatch the temporal, cross-session, and abstention-sensitive operations required by memory-heavy queries.

The value of Skill2Mem is diagnostic rather than leaderboard-oriented. Its explicit priors make the transfer boundary inspectable through selected skills, selector concentration, correctness flips, confidence-gating coverage, and k/budget sensitivity. The results point toward domain-matched memory banks, stronger selector controls, and write/update interventions as next steps for reliable memory skills.

10 Limitations

Skill2Mem is intended as a diagnostic boundary study, and its current scope points to several direct extensions. First, the main evidence uses fixed evaluation subsets; full benchmark runs, multi-seed estimates, and open-model replications are the natural way to turn the present diagnosis into a stronger benchmark claim. Second, the TF–IDF and embedding diagnostics differ in backbone settings, so a fully controlled selector grid over rerankers, model backbones, and seeds is needed before attributing all variation to the selector alone. Third, the public banks are not domain-matched to memory operations: `noisy_size18` reduces the simple size confound, but a stronger follow-up should build or balance a public memory-oriented bank by operation tags such as temporal reasoning, conflict update, preference, and abstention. Finally, this paper studies retrieve-only prior injection into a retrieval planner; write-time extraction, update-time consolidation, and learned controllers are natural next intervention points for testing whether better interface alignment turns public procedures into useful memory skills. Public skill sources also require source-aware filtering and redistribution checks before any derived priors are released or deployed.

11 Ethical Considerations

External skills can encode biases, unsafe instructions, or domain-specific assumptions. Reusing them as memory priors therefore requires at least three safeguards: source-aware filtering, exclusion of clearly unsafe or high-risk domains, and auditing of failure modes where a prior induces incorrect overwrites, unsafe retrieval suggestions, or privacy-sensitive storage patterns. More broadly, importing public agent resources into academic memory benchmarks raises questions about transparency and redistributability. Derived priors and per-example diagnostics should therefore be released only when source constraints permit; otherwise, aggregate statistics are the appropriate public evidence.

References

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, Bo Zhang, and Liang He. 2026. AutoSkill: Experience-Driven Lifelong Learning via Skill Self-Evolution. arXiv preprint arXiv:2603.01145.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025. Rethinking Memory in LLM based Agents: Representations, Operations, and Emerging Topics. arXiv preprint arXiv:2505.00675.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2025. Memp: Exploring Agent Procedural Memory. arXiv preprint arXiv:2508.06433.

Shicheng Ye, Chao Yu, Kaiqiang Ke, Chengdong Xu, and Yinqi Wei. 2025. H2R: Hierarchical Hindsight Reflection for Multi-Task LLM Agents. arXiv preprint arXiv:2509.12810.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. arXiv preprint arXiv:2507.05257.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI Agent. arXiv preprint arXiv:2506.06326.

Zhiyu Li, Chenyang Xi, Chunyu Li, et al. 2025. MemOS: A Memory OS for AI System. arXiv preprint arXiv:2507.03724.

Jiafeng Liang, Hao Li, Chang Li, et al. 2025. AI Meets Brain: Memory Systems from Cognitive Neuroscience to Autonomous Agents. arXiv preprint arXiv:2512.23343.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026. MemSkill: Learning and Evolving Memory Skills for Self-Evolving Agents. arXiv preprint arXiv:2602.02474.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating Very Long-Term Conversational Memory of LLM Agents. arXiv preprint arXiv:2402.17753.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. 2026. OpenClaw-RL: Train Any Agent Simply by Talking. arXiv preprint arXiv:2603.10165.

Simon Yu, Gang Li, Weiyan Shi, and Peng Qi. 2025. PolySkill: Learning Generalizable Skills Through Polymorphic Abstraction. arXiv preprint arXiv:2510.15863.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv preprint arXiv:2501.13956.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. 2026. SimpleMem: Efficient Lifelong Memory for LLM Agents. arXiv preprint arXiv:2601.02553.

Xiangyi Li, Wenbo Chen, Yimin Liu, et al. 2026. SkillsBench: Benchmarking How Well Agent Skills Work Across Diverse Tasks. arXiv preprint arXiv:2602.12670.

Zhen Tan et al. 2025. In Prospect and Retrospect: Reflective Memory Management for Long-term Personalized Dialogue Agents. arXiv preprint arXiv:2503.08026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. arXiv preprint arXiv:2410.10813.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't Hallucinate, Abstain: Identifying LLM Knowledge Gaps via Multi-LLM Collaboration. arXiv preprint arXiv:2402.00367.

Cheng Yang, Xuemeng Yang, Licheng Wen, et al. 2025. Learning on the Job: An Experience-Driven Self-Evolving Agent for Long-Horizon Tasks. arXiv preprint arXiv:2510.08002.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv preprint arXiv:2504.19413.

A Experimental protocol and baselines

We report controlled subset evaluations rather than full-benchmark sweeps. All LongMemEval conditions are evaluated on the same 300-question subset, and all LoCoMo conditions are evaluated on the same 200-example subset. Sampling, random controls, and paired comparisons use fixed seeds so that every condition is scored on identical items.

System settings. Unless otherwise stated, the TF–IDF subset runs use retrieval-only intervention, a fixed prior budget $B = 384$, $k = 3$ injected priors, and deterministic decoding. The underlying memory system is SimpleMem (Liu et al., 2026). Table 5 consolidates the models, banks, and evaluation controls for this main subset run. The embedding-selector controls keep the same LongMemEval subset and intervention surface, but switch the selector/model settings described in Table 6; those controls are used for diagnostic stress testing rather than direct model-controlled selector comparison.

Evaluation measures. For LongMemEval, we report exact match and judge accuracy under the yes/no protocol of Wu et al. (2024). For LoCoMo, we report the official category-aware QA score. Efficiency measures (tokens and latency) are kept out of the main paper and reported only in the appendix.

B Skill bank statistics

Table 7 summarizes the three skill banks used in our LongMemEval evaluation subset study. Token counts are computed over the source skill text using a standard tokenization scheme.

Skill length distribution. Figure 4 visualizes the token-length distributions of source skills in each bank. Public skills are often thousands of tokens long, while our prior injection is capped at a fixed budget B=384 tokens; this gap motivates our budgeted truncation scheme and ensures comparability across banks.

C Prior representation and budget control

Prior representation. Each skill is adapted into a compact structured prior containing trigger, retrieval, write, and update fields. We require this representation to be valid before reuse across runs. The adapter is instructed to prefer memory operations such as temporal normalization, conflict handling, session boundary tracking, source attribution, and abstention.

Proposition (budget fairness). For any query, Skill2Mem injects at most B prior tokens, and therefore does not advantage conditions with longer source skills. *Proof.* Each retrieved prior is truncated to at most $\lfloor B/k \rfloor$ tokens; the concatenated prior block is then truncated again to B tokens.

Generic guideline baseline. The `generic` condition uses a budget-matched retrieval guideline instead of skill-specific priors. The guideline emphasizes atomic information needs, targeted retrieval, temporal normalization, conflict resolution, and abstention. It is truncated to the same total budget (B=384) to isolate the effect of *procedural content* rather than length.

D Uncertainty, flips, and selector confidence

Paired bootstrap uncertainty (judge accuracy). Table 8A reports paired bootstrap uncertainty for per-condition judge-accuracy deltas relative to `base`, computed over the 300 shared questions

Item	Choice	Notes / reference
Base system	SimpleMem	Train-free memory pipeline with intent-aware retrieval planning (Liu et al., 2026)
Primary benchmark	LongMemEval	Interactive long-term memory benchmark with five ability categories (Wu et al., 2024)
Secondary benchmark	LoCoMo	Very long conversational memory benchmark (Maharana et al., 2024)
Agent model	`gpt-5.4-high`	Used for generation in the TF–IDF subset runs; fixed across conditions
Adapter model	`gpt-5.2-low`	Used for one-shot prior adaptation in the reported setting
Judge model	`gpt-5.2-low`	Used for LongMemEval yes/no judging
Subset recipe	`seed=42`, deterministic shared subsets	Shared 300-question LongMemEval subset; shared 200-example LoCoMo subset
Curated public bank	curated public skill collection	Curated public bank used in the study
Noisy public bank	broader public skill collection	Larger public-skill control bank
Memory-native bank	MemSkill priors	Research-native memory-operation source (Zhang et al., 2026)
Controls	Generic guideline; random priors	Same-budget non-skill and random-selection controls introduced in this paper
Decoder / selector	temperature 0; TF–IDF top-k matching	Shared across conditions; retrieval-side only
Diagnostic evidence	per-example selections, selector confidence, efficiency summaries, case annotations	Used only for diagnosis and appendix analyses

Table 5: Experimental protocol for the main TF–IDF subset evaluations. Section I summarizes the reproducibility information associated with this setting.

Item	Additional setting	Purpose
Shared subset	LongMemEval 300, seed 42	Keeps question IDs comparable to the main LongMemEval run
Agent model	`gpt-5.4`	Used for the embedding-selector controls; not compared numerically against the TF–IDF `gpt-5.4-high` run
Selector	local embedding retrieval	Uses a local MiniLM sentence-transformer; no OpenAI embedding calls
Primary diagnostic	embedding selector	Produces base/generic/random/curated/MemSkill condition summaries
Bank-size control	`noisy_size18`	Fixed-seed 18-prior sample from the noisy public bank
Mitigation	confidence-gated injection	Thresholds selected from curated top-1 embedding-score quantiles
Sensitivity	curated k/B matrix	Tests k=1, 3, 5 and B=192, 384, 768 around the default

Table 6: Embedding-selector diagnostic protocol. These runs isolate selector strength, bank size, gating, and default-parameter sensitivity while preserving the retrieve-only Skill2Mem interface.

Bank	N	Mean tok	Median	Min–Max	Source type
Curated public	18	2354	2031	28–7322	public execution skills
Noisy public	100	1516	1019	16–12875	broader public skills
MemSkill (research-native)	16	140	150	81–182	memory-native skills

Table 7: Skill bank statistics for the LongMemEval evaluation subset.

(20,000 bootstrap resamples; seed 0). We report percentile 95% intervals.

Correctness flips vs `base` (judge labels). Table 8B reports correctness flips relative to `base` on the 300 shared LongMemEval questions. "Positive" flips are incorrect-to-correct; "negative" flips are correct-to-incorrect.

Selector confidence statistics. Table 8C reports selector confidence statistics for conditions that retrieve priors using TF–IDF similarity. We report

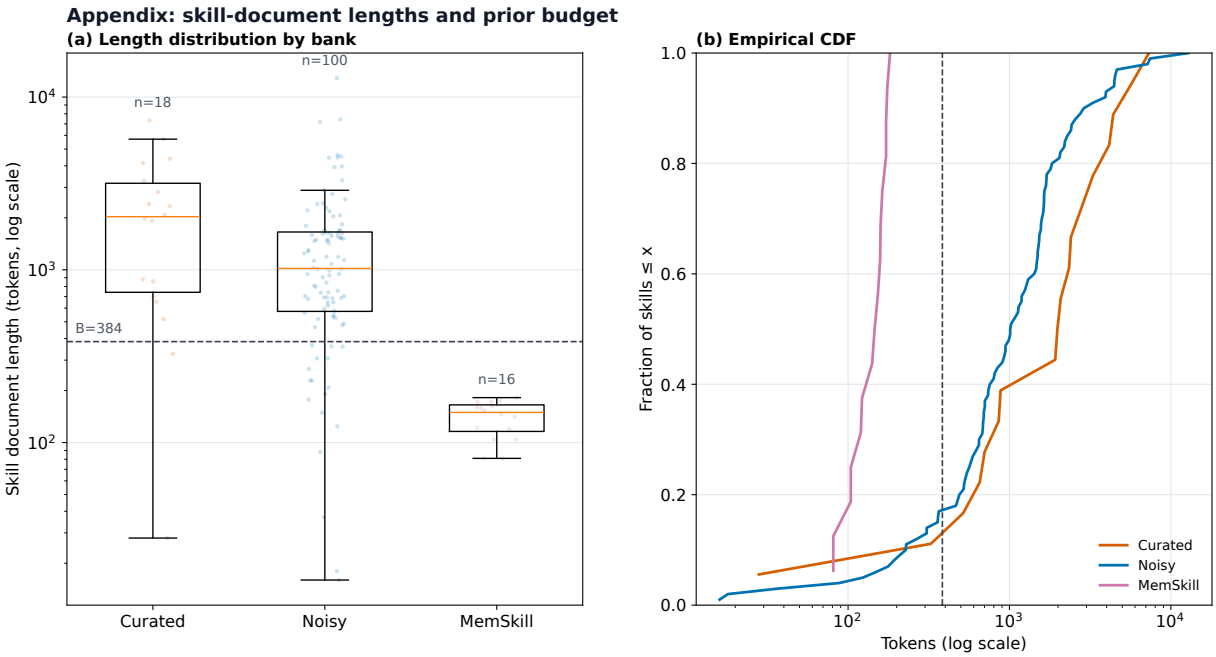

Figure 4: Source-skill lengths by bank (log scale) and empirical CDFs. The dashed line marks the fixed prior-token budget B=384 used in all conditions.

the top-1 similarity score per query.

E LongMemEval breakdown by question type

Table 9 reports judge-accuracy breakdowns for the shared 300-question LongMemEval evaluation subset.

F Selection analysis and case studies

The selector-level quantitative diagnostics and representative case visualization are kept in main Figures 2 and 3. The appendix adds supporting counts.

Most frequent prior categories. Table 10 shows the most frequently selected prior categories in the shared 300-question LongMemEval evaluation subset. Counts are over all selected slots (300 questions $\times$ k=3).

Prior categories associated with negative flips. To localize failure modes, we count which selected prior categories appear on *correct*$\rightarrow$*incorrect* flips relative to `base` (judge labels). In the curated public bank, a small number of broad execution categories dominate negative flips, consistent with an alignment gap between public execution priors and memory procedures. In MemSkill, negative flips involve memory-native operators, suggesting a different failure mode (prior-policy interference).

Representative curated-bank cases. Table 12 summarizes four representative cases drawn from the improve/degrade case sets. We keep the appendix focused on a few interpretable examples here and defer the full case inventory to the supplementary materials.

G LoCoMo and efficiency

Table 13 reports mean token usage by stage on LongMemEval. The fixed prior-token budget primarily increases retrieval-planning stages, while answer-generation tokens remain comparable.

H LoCoMo results

Table 14 reports the LoCoMo subset evaluation (200 QA; seed 42; retrieval-only; k = 3; total prior budget 384). Unlike LongMemEval, LoCoMo scores are tightly clustered among conditions at this scale; efficiency overhead is still visible.

I Reproducibility summary

The supplementary materials accompanying this paper contain the information needed to reproduce the reported comparisons and examine their behavior: model settings, seeds, prior-budget settings,

(A) Paired bootstrap Δ vs base

Condition	Δ (pp)	95% CI
Generic	-1.33	$[-4.00, +1.33]$
Random	$+0.33$	$[-2.33, +3.00]$
Curated	-2.67	$[-5.33, +0.00]$
Noisy	$+0.33$	$[-1.67, +2.33]$
MemSkill	-2.00	$[-4.33, +0.33]$

(B) Correctness flips vs base

Cond.	Pos	Neg	Net
Generic	6	10	-4
Random	8	7	$+1$
Curated	5	13	-8
Noisy	5	4	$+1$
MemSkill	4	10	-6

(C) Selector confidence (top-1 sim.)

Bank	Mean	Med	P90	% $<$ 0.05
Curated	0.070	0.067	0.114	27.0
Noisy	0.067	0.063	0.099	24.3
MemSkill	0.095	0.092	0.144	8.7

Table 8: LongMemEval evaluation-subset summary statistics for (A) paired bootstrap uncertainty of judge-accuracy deltas, (B) correctness flips relative to `base`, and (C) selector confidence (top-1 TF–IDF similarity).

Type	N	Base	Gen.	Rand.	Cur.	Noisy	Mem
knowledge-update	46	0.848	0.826	0.891	0.870	0.891	0.870
multi-session	76	0.671	0.671	0.645	0.605	0.645	0.618
single-session-assistant	40	1.000	1.000	1.000	1.000	1.000	1.000
single-session-preference	15	0.467	0.400	0.467	0.400	0.533	0.333
single-session-user	42	0.952	0.952	0.976	0.952	0.952	0.952
temporal-reasoning	81	0.630	0.605	0.630	0.593	0.630	0.617

Table 9: LongMemEval breakdown by question type on the shared 300-question evaluation subset (judge accuracy). The agent and judge models are fixed across conditions.

Curated (public)	
`visual-procedure`	218
`document-procedure`	126
`template-procedure`	105
`interface-procedure`	72
`slide-procedure`	53
`canvas-procedure`	44
`interface-validation`	34
`service-procedure`	33
`output-procedure`	33
`communication-procedure`	31
MemSkill (memory-native)	
`refine-temporal-details`	142
`delete`	136
`capture-temporal-context`	91
`insert`	90
`update`	84
`capture-activity-details`	84
`track-object-movements`	39
`capture-entity-nuances`	37
`track-object-location`	36
`handle-entity-relationships`	29

Table 10: Most frequently selected prior categories on the LongMemEval evaluation subset (top-10 by count; 900 total selected slots).

Curated negative flips (N=13)	
`visual-procedure`	12
`template-procedure`	6
`document-procedure`	3
`interface-validation`	3
`document-format-procedure`	2
`canvas-procedure`	2
`communication-procedure`	2
`slide-procedure`	2
`output-procedure`	1
`tool-use-procedure`	1
MemSkill negative flips (N=10)	
`capture-activity-details`	5
`delete`	5
`refine-temporal-details`	5
`insert`	4
`update`	4
`capture-entity-nuances`	1
`noop`	1
`track-object-movements`	1
`handle-entity-relationships`	1
`track-object-location`	1

Table 11: Top prior categories on correct→incorrect flips vs `base` (top-10 by count; LongMemEval evaluation subset; judge).

bank definitions, per-example selections, predictions, judge outcomes, uncertainty estimates, type breakdowns, and improve/degrade case lists.

Study setting. For the main TF–IDF LongMemEval study, the reported setting fixes the agent model to `gpt-5.4-high`, the adapter and judge model to `gpt-5.2-low`, the shared seed to 42, the injected-prior count to k=3, and the total prior budget to B=384. These materials are sufficient to reproduce the condition-level tables and identify which priors were selected for each query.

Embedding-control evidence. The embedding-selector diagnostic uses the same LongMemEval shared 300 subset and includes condition summaries, selector summaries, bank tags, question-alignment tables, and per-condition breakdowns. This evidence supports within-run selector, gating, size-control, and k/B analyses; it is not treated as a direct numerical replacement for the TF–IDF

D	QID	Type	Query / reference	Base → curated	Selected priors
–	`61f8c8f8`	multi-session	5K improvement vs. previous year; ref: 10 minutes	"10 minutes faster" → "Cannot be determined"	`visual-procedure`, `skill-authoring`, `interface-validation`
–	`a9f6b44c`	multi-session	bikes serviced/planned in March; ref: 2	"2 bikes" → "1 bike"	`tool-use-procedure`, `visual-procedure`, `template-procedure`
+	`7401057b`	knowledge-update	Hilton free-night stays; ref: two stays	"1 free night" → "Two free-night stays"	`output-procedure`, `visual-procedure`, `template-procedure`
+	`edced276`	multi-session	total travel days in Hawaii + NYC; ref: 15 days	"Cannot determine total" → "15 days total"	`spreadsheet-procedure`, `document-procedure`, `visual-procedure`

Table 12: Representative curated-bank case snapshots from the LongMemEval evaluation subset. Full improve/degrade case lists are included in the supplementary materials.

Condition	Plan info req.	Plan targeted queries	Analyze query	Answer gen.
Base	714	878	388	3841
Generic	1131	1237	395	3793
Random	1082	1224	391	3733
Curated	1090	1235	396	3771
Noisy	1104	1227	393	3781
MemSkill	1098	1218	390	3737

Table 13: Mean tokens by stage on the LongMemEval evaluation subset. The agent model is fixed across conditions.

Cond.	Score	Tok.	Sec.
Base	0.338	3,343	32.88
Generic	0.333	4,096	41.29
Random	0.337	4,070	54.38
Curated	0.338	4,068	53.92
MemSkill	0.337	4,078	40.30

Table 14: LoCoMo results on the shared 200-question subset (retrieval-only; $k = 3$; B=384).

`gpt-5.4-high` tables.

The embedding-control materials include the condition-level summaries used for the main selector, gating, and k/B tables; per-question rows used for selector concentration, bank tags, and breakdowns; and confidence-gating evidence used for the coverage–accuracy tradeoff.

Secondary observations. Taken together, the appendix tables reinforce two observations that are useful but not central enough for the main paper. First, the extra compute is concentrated in retrieval-planning stages rather than answer generation, which is consistent with the narrow intervention studied here. Second, LoCoMo remains essentially flat across conditions at this scale, suggesting that the negative-transfer signal is strongest when the benchmark explicitly stresses multi-session reasoning, temporal normalization, and update-sensitive retrieval.

J Positioning overview

Table 15 provides a positioning view that contrasts train-free memory systems, procedural-memory skill learning, and public skill ecosystems. This comparison makes the paper's scope precise. Unlike MemSkill or AutoSkill, we do not learn or evolve memory skills. Unlike SkillsBench, we do not study execution-time skill utility. Unlike SimpleMem, Mem0, or Memory OS, we do not propose a new memory architecture. The contribution instead sits at the intersection of these lines: a train-free memory base, explicit public skill resources, and a diagnostic evaluation of whether procedural transfer succeeds at test time.

Work	Year	Core idea	Train-free?	Main benchmarks	Relevance to this paper
LoCoMo	2024	Very long multi-session conversational benchmark with temporal/event structure	–	LoCoMo	Primary benchmark for long-range memory and event reasoning
LongMemEval	2024/25	Interactive memory benchmark with five abilities and a 3-stage design lens	–	LongMemEval	Primary benchmark for update/abstention and interactive memory
Mem0	2025	Dynamic extraction, consolidation, retrieval; graph variant	Yes	LoCoMo	Strong practical baseline in train-free memory design
RMM	2025	Prospective and retrospective reflection for dialogue memory	No/Hybrid	LongMemEval	Relevant retrieval-refinement comparator; more complex than our target
Memory OS	2025	Hierarchical storage/update architecture inspired by OS	Mostly yes	LoCoMo	Shows value of structured memory operations without skill transfer
Memp	2025/26	Learnable lifelong procedural memory from trajectories	No/Hybrid	ALFWorld, TravelPlanner	Closest procedural-memory framing; useful conceptual neighbor
H2R	2025	Hierarchical hindsight reflection for planning/execution memory	No	ALFWorld, PDDLGame	Supports stage decomposition and hierarchical memory intuition
PolySkill	2025/26	Decouples skill goals from task realization for generalization	No	Mind2Web	Motivates abstraction when adapting public skills
MemSkill	2026	Memory operations as learnable/evolvable skills	No	LoCoMo, LongMemEval	Main conceptual anchor; source of memory-native priors
AutoSkill	2026	Experience-driven lifelong learning via self-evolving skill repository	No/Hybrid	Multi-domain	Shows skill standardization and cross-agent reuse potential
SkillsBench	2026	Evaluates how curated skills help agents across domains	–	84 tasks, 11 domains	Main motivation that skill curation and granularity matter
SimpleMem	2026	Train-free semantic compression, synthesis, retrieval planning	Yes	LoCoMo	Base memory system for Skill2Mem

Table 15: Positioning of the proposed study. We deliberately target the narrow intersection of *train-free memory systems* and *external skill priors*.