PAY WHEN ASKED: DEMAND-DRIVEN AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term agents must organize growing archives without knowing which information future queries will need. We view memory as demand-driven computation: retain source evidence while postponing structure until it is useful. Yet query-time construction alone leaves a second cost unresolved: repeatedly rebuilding relations for different questions about the same evidence. We formalize this trade-off by separating global access coverage, local reuse, and answer computation. Our central insight is that sparse global access can coexist with repeated local demand, making selective construction and persistent reuse complementary. Guided by this insight, we introduce **SRH-Graph**, a demand-driven relation-memory method that constructs source-grounded relations on retrieval, reuses them across distinct questions, and invalidates them when sources change; **SRH-Base** provides the lightweight retrieval-only variant. On controlled multi-hop chains, **SRH-Graph** reaches 80.83–100% accuracy where Naive RAG reaches 0–3.33%; reuse reduces repeated construction, while deferral savings vanish near full coverage. On real benchmarks, we find gains on a fixed sparse document stream, but retain Naive RAG's conversational advantage and report higher graph costs. Our results identify when organizing memory on demand is useful, and when simpler retrieval or advance construction remains preferable.

1 INTRODUCTION

Long-term agents must use growing conversation and document archives beyond their immediate context (Packer et al., 2023; Zhong et al., 2024). We ask when organizing those archives is worth its computational cost. Retrieval-augmented generation exposes source evidence without first converting it into structured memory (Lewis et al., 2020; Karpukhin et al., 2020); extraction-based memory and graph indexing add facts, relations, or summaries (Chhikara et al., 2025; Edge et al., 2024a). Constructing such representations for every arrival, however, commits computation before future demand is known.

We view memory as *demand-driven computation*: retain searchable evidence, but build additional structure when a query needs it. LazyGraphRAG and LazyMem already motivate deferred or selective construction (Edge et al., 2024b; Yu et al., 2026). Our question goes beyond when construction starts: *what should persist across distinct questions about the same evidence?* Deferral alone does not specify a reuse policy. Without persistent reuse, answering different questions about one neighborhood can repeatedly incur its construction cost. We therefore study selective construction and reuse jointly, rather than claim query-time construction itself as new.

We formalize this problem by separating *global access coverage*, *local reuse*, and *answer computation*. Our central insight is that low global coverage can coexist with high local reuse (Figure 1): many distinct questions may concern a small part of an archive. Deferral avoids structuring untouched sources; reuse avoids reconstructing visited ones. Separating answer cost also reveals a boundary: savings in construction need not imply comparable lifecycle savings.

From this insight, we develop **SRH-Graph**: *Demand-Driven Relation Memory*. We construct source-grounded relations on retrieval, follow them to bridging evidence, and retain them across questions with source-version invalidation. We also provide **SRH-Base**, a retrieval-only variant for workloads that do not justify relation construction. Users choose between these two versions; we do not introduce an automatic router.

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

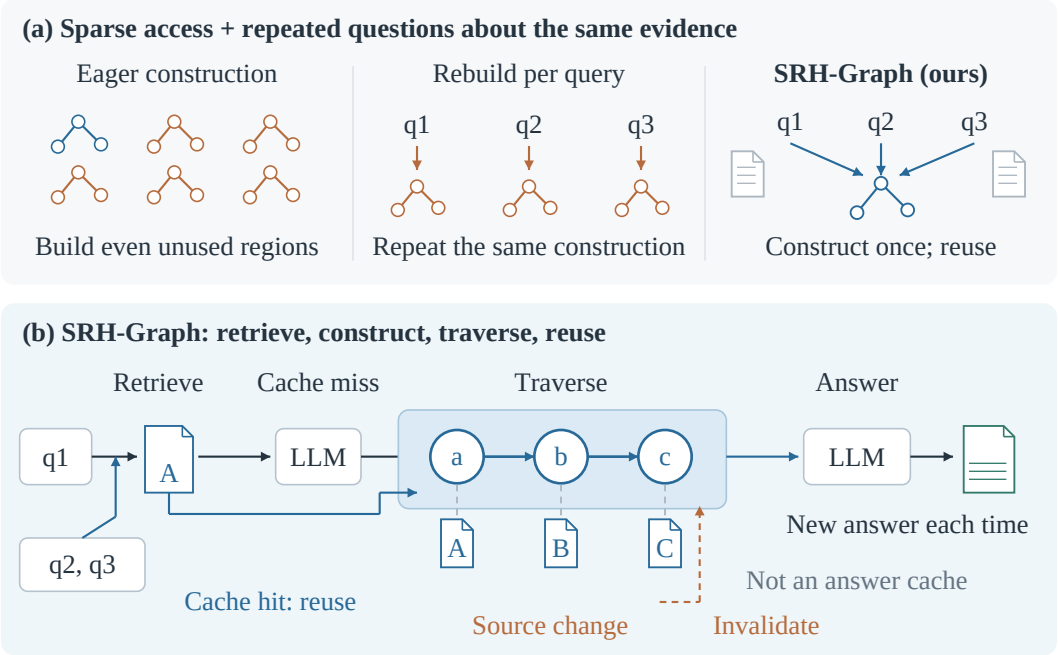

Figure 1: **SRH-Graph** couples selective construction with persistent relation reuse. Gray documents remain unstructured; later queries reuse valid relations but generate new answers. Source changes invalidate affected relations. Schematic, not measured data.

We evaluate the resulting design through matched scheduling, controlled multi-hop streams, and real benchmarks. On the controlled chains, **SRH-Graph** reaches 80.83–100% accuracy versus Naive RAG's 0–3.33%; reuse reduces repeated construction, but deferral savings vanish near full coverage. On real tasks, we find gains on a fixed sparse document stream alongside conversational regressions and higher costs than Naive RAG. Our contributions are the coverage–reuse formulation, a source-versioned relation-memory method, and an evaluation of its operating boundaries. Together, they explain *when to pay for structure*, not why every workload should use a graph.

2 Related Work

Retrieval and agent memory. Beyond the retrieval and retention approaches above, Zep/Graphiti emphasizes temporal knowledge graphs and MemOS unifies memory management (Rasmussen et al., 2025; Li et al., 2025). We compare frozen Mem0, Graphiti, and MemOS implementations on matched cohorts, not every configuration of these frameworks.

Deferred construction and graph retrieval. GraphRAG builds entity and community representations (Edge et al., 2024a); LazyGraphRAG and LazyMem defer construction (Edge et al., 2024b; Yu et al., 2026). We study source-versioned relation reuse; matched comparisons remain incomplete. At our September 26, 2026 check, LazyMem's public repository[1] lacked its trained 4B checkpoint, limiting direct reproduction of that variant. We cite LazyGraphRAG's technical blog; by the same date, we found no complete public research implementation matching its evaluation configuration.

Budget allocation and evaluation. BudgetMem learns routing among runtime memory budget tiers (Zhang et al., 2026); we instead fix each variant and measure its operating point. LoCoMo, LongMemEval, ConvoMem, and RepLiQA test conversational memory and document answering beyond synthetic mechanisms (Maharana et al., 2024; Wu et al., 2025; Pakhomov et al., 2025; Monteiro et al., 2024). Their cohorts, metrics, and historical scoring paths remain distinct.

[1] https://github.com/allacnobug/LazyMem

Table 1: Evaluated variants and controls. A graph ablation is not an implementation of a **SRH-Base/SRH-Graph** runtime switch. The synthetic study disables the answer cache in every arm.

Configuration	Construction timing	Persistent relations	Evidence expansion
SRH-Base	No auxiliary LLM on primary path	None	Frozen base retrieval
SRH-Graph	On admitted-source demand	Source-version keyed	Relation-guided
Matched Eager-Graph	Before questions	Same extraction recipe	Relation-guided
Graph w/o reuse	Rebuilt across queries	No cross-query reuse	Relation-guided
Graph w/o traversal	Query time	Extraction still runs	No edge traversal

3 SRH-GRAPH: DEMAND-DRIVEN RELATION MEMORY

We retain source evidence independently of relation construction so that unconstructed sources remain searchable. We next define our variants, validity rules, and cost model.

3.1 TWO EVALUATED VARIANTS

SRH-Base stores source text and a retrieval index, then selects and packs evidence at query time. Its primary benchmark path performs extractive packing rather than LLM summarization. Benchmark adapters determine the retrieval and context budget; these configurations are frozen within each comparison. The absence of write-side LLM calls does not imply free ingestion: embedding, indexing, storage, and ordinary CPU work remain.

SRH-Graph retains the base evidence context and adds a source-grounded relation cache. It admits the first two source search hits, constructs relations for admitted sources that lack a valid cached extraction, and expands retrieval through referenced entities. The implementation follows outgoing objects for at most three expansion rounds, searches at most two entities per round with two hits per search, and admits at most six sources. These bounds limit the query path; they do not guarantee retrieval of every necessary bridge.

The extractor operates on source content rather than the question or reference answer. It returns at most twelve subject–predicate–object relations with supporting quotations. Each accepted field must occur literally within the corresponding contiguous source quotation; the record binds the source identifier, content hash, and character offsets. Query mentions initialize traversal, and the resulting source quotations are deduplicated and appended only if space remains in the 16,000-character evidence budget. Existing base context is not displaced. This preserves its text, not the answer model's behavior: extra evidence and a longer prompt can still change an answer.

3.2 VALIDITY AND REUSE

The cache key includes corpus, source version, and extraction recipe. Modified or deleted sources invalidate their associated cached relations. Empty or rejected extractions are also cached, so absence of a usable relation does not force a repeated extraction on every query. The raw archive remains searchable: traversal can retrieve a previously untouched source and extend the graph on demand.

This cache stores *relations*, not final answers. Two questions can share a relation while requiring distinct answers and model calls. Conversely, an unchanged final answer does not establish a graph-cache hit. We distinguish relation reuse from exact-match answer caching throughout the experiments. Literal source validation provides traceability, not proof that the relation is semantically entailed or that traversal has found the correct evidence.

Graph w/o traversal still extracts, caches, and quotes relations from initial hits; only edge-guided expansion is disabled. It is not **SRH-Base**. Our two-variant presentation makes no claim of runtime off-switch equivalence.

3.3 COVERAGE AND LOCAL REUSE

Let $\mathcal{V}$ be the source versions available during a query sequence and $\mathcal{U}_Q$ the versions constructed on demand. Under a common per-version construction cost g_v, with answer cost a_q^X for system X, a simple lifecycle LLM-token model is

$$C_{\text{eager}}(Q) = \sum_{v \in \mathcal{V}} g_v + \sum_{q=1}^{Q} a_q^{\text{eager}}, \tag{1}$$

$$C_{\text{graph}}(Q) = \sum_{v \in \mathcal{U}_Q} g_v + \sum_{q=1}^{Q} a_q^{\text{graph}}. \tag{2}$$

Source updates introduce new versions and therefore new construction costs. Subtracting Eq. (2) from Eq. (1) gives

$$C_{\text{eager}} - C_{\text{graph}} = \sum_{v \notin \mathcal{U}_Q} g_v - \sum_{q} (a_q^{\text{graph}} - a_q^{\text{eager}}). \tag{3}$$

The first term is avoided construction; the second is changed answer computation. Actual runs meter construction separately because identical recipes need not produce identical calls.

For intuition, suppose each of m visited neighborhoods among N requires one construction of cost g and receives R questions of answer cost a. Eager, cached demand-driven, and uncached reconstruction cost $Ng + mRa$, $mg + mRa$, and $mRg + mRa$. Sparse coverage favors deferral; local reuse favors caching over reconstruction. Increasing R need not increase percentage savings against eager construction because answers dilute the construction saving. Raw retrieval avoids g and can remain cheaper.

4 EVALUATION DESIGN

We isolate scheduling with fixed downstream artifacts, test traversal and reuse with graph controls, and evaluate complete systems on real benchmarks. Bold method names identify our variants, not the best score.

Component-matched scheduling. We retain the original protocol-frozen sample of 100 LongMemEval-S questions. BGE-M3 retrieves a mean of 1.87 of 47.99 source sessions; query-agnostic Qwen3.5-35B evidence cards and a Qwen3.5-397B answerer operate under a 4,100-token evidence ceiling. Cards are generated once. The eager accounting arm is charged for all cards, while the deferred arm is charged only for the selected cards; both reference the same answer call. Context and output identity are checked by hashes. This is a coupled accounting intervention, not two independent service runs.

Controlled graph workloads. Three fixed synthetic corpora each contain 100 root objects and three-source relation chains. We query 1, 5, 10, 20, or 50% of objects, with eight distinct questions per hotspot, and add a hotspot-migration/source-update stream. Six arms yield 15,624 answers. A separate matched two-arm extension tests 75 and 100% coverage with 8,400 answers. The graph extractor is Qwen3.5-27B; the common answerer is Qwen3.5-397B with temperature zero and thinking disabled. Scoring uses exact JSON values, not LLM judges. All arms have their final-answer cache disabled. Graph-based arms share an extraction and packing recipe, but their separate model calls need not be bit-identical.

Real tasks and comparators. We evaluate LoCoMo (1,986 questions, ten conversations), ConvoMem (657 questions, 50 personas), LongMemEval-S (500 questions), and a fixed sparse RepLiQA stream (40 questions over an archive of 500 documents). LoCoMo and ConvoMem have paired **SRH-Base** outputs in the new evaluation package. LongMemEval has no new **SRH-Base** arm; RepLiQA's **SRH-Base** comparator is historical. External-system outputs are reused only on exactly matched question cohorts, with frozen question/reference bindings. They are historical comparisons, not contemporaneous re-runs. The real answerer is the frozen DeepSeek configuration; synthetic results are not substituted for it.

Retrieval and exposure. The Naive RAG control uses its original 900-character chunks, 150-character overlap, BGE-M3 cosine ranking, and budget filling. Its historical `top_k` parameter is ineffective in this path; we do not describe it as a strict top-k baseline. We do not add hybrid reranking, graph traversal, or answer caching to this control. Graph-derived quotations can increase the actual answer prompt even when a nominal ceiling is shared. Consequently, **SRH-Graph**–**SRH-Base** and **SRH-Graph**–Naive RAG differences measure complete pipelines, not the isolated causal effect of graph edges under equal prompt exposure.

Scoring and uncertainty. The first three real tasks use binary correctness from three primary judges and necessary blind GPT/GLM adjudication. Nine remaining LoCoMo/ConvoMem items were resolved by one actual human reviewer, not by generated labels or a multi-reviewer consensus. Technical answer failures remain in the full denominator with score zero; missing judge responses are routed for resolution rather than converted into negative votes. RepLiQA retains its original set-token F1 metric. We do not average F1 with accuracy.

LoCoMo and ConvoMem intervals use paired conversation/persona cluster bootstrap, respectively; LongMemEval uses question pairing. RepLiQA uses document-cluster resampling conditional on the fixed stream, whose evolving shared cache can introduce additional dependence. Real-task analyses use 20,000 bootstrap replicates. Synthetic inference uses source clusters and 10,000 bootstrap replicates, conditional on the three designed corpora. We report the appropriate sign-flip tests and preserve the frozen multiplicity families. A confidence interval crossing zero is not evidence of non-inferiority. Appendix B records execution, judging, and recovery details.

5 Scheduling, Coverage, and Reuse

We first measure scheduling with fixed answer evidence, then test traversal and reuse on multi-hop streams. Extending coverage reveals where deferral has little construction left to avoid.

5.1 The isolated scheduling effect

The matched first-query experiment costs a mean 126,688 tokens under eager charging and 4,549 under deferred charging, a 96.41% reduction. The mean paired saving is 122,138 tokens, with a 95% interval of [121,749, 122,528]. This saving has a precise interpretation: it avoids charging for unselected cards while holding the answer context and output fixed. It does not establish a quality advantage or measure a deployed query-latency improvement.

A separate 200-question experiment allows the deferred card generator to see the question. It uses 4,651 versus 127,020 first-query tokens, but accuracy is 30.0% versus 32.0%, a difference of -2.0 percentage points (pp), CI $[-5.5, +1.5]$. Its quality-superiority test fails. We retain it to distinguish a large cost reduction from evidence of preserved or improved answer quality. The original meter-based median crossover of 32.30 queries assumes no reuse; it is not an observed cache trajectory.

5.2 When one-pass retrieval misses the bridge

Table 2 exposes a limitation hidden by strong aggregate Naive RAG scores on real tasks: on these controlled multi-hop chains, its exact-value accuracy is only 0–3.33%, versus 80.83–100% for **SRH-Graph** at the same five coverage levels. **SRH-Base** also remains at 0–0.42%, showing that the lightweight variant alone does not solve this workload. At 1% coverage, **SRH-Graph** retrieves every measured cold bridge while constructing 25 of 900 source versions. At 50%, it constructs 598 and recovers 2,160 of 2,386 cold bridges. Relation-guided retrieval can thus reach bridging evidence in an otherwise unconstructed archive.

Disabling edge-guided expansion while retaining extraction and caching reduces accuracy to 0–0.50% (Table 7), supporting the traversal mechanism. Exposure also changes: at 50% coverage, median answer inputs are 5,427 tokens for **SRH-Graph** and 4,780 for Naive RAG. The full gain is not attributable to topology alone. These deliberately relational workloads favor graph retrieval; they do not establish failure of Naive RAG on every multi-hop task.

Quality is not monotonically related to coverage. At 20%, **SRH-Graph** trails matched eager construction by 3.33 pp, with source-cluster CI $[-10.00, +3.33]$. At 75%, the difference is -2.11 pp,

Table 2: Controlled multi-hop exact-value accuracy (%). n: questions per arm across three seeds. Naive RAG and **SRH-Base** rarely resolve the designed chains. The boundary extension tests only the matched graph pair; dashes mean not run. Mechanism ablations are in Table 7.

Coverage	n	Naive RAG	**SRH-Base**	**SRH-Graph**	Matched eager
1%	24	0.00	0.00	100.00	100.00
5%	120	3.33	0.00	100.00	100.00
10%	240	2.08	0.00	90.00	86.67
20%	480	1.67	0.00	86.67	90.00
50%	1,200	1.50	0.42	80.83	78.75
75%	1,800	—	—	81.06	83.17
100%	2,400	—	—	83.08	83.08

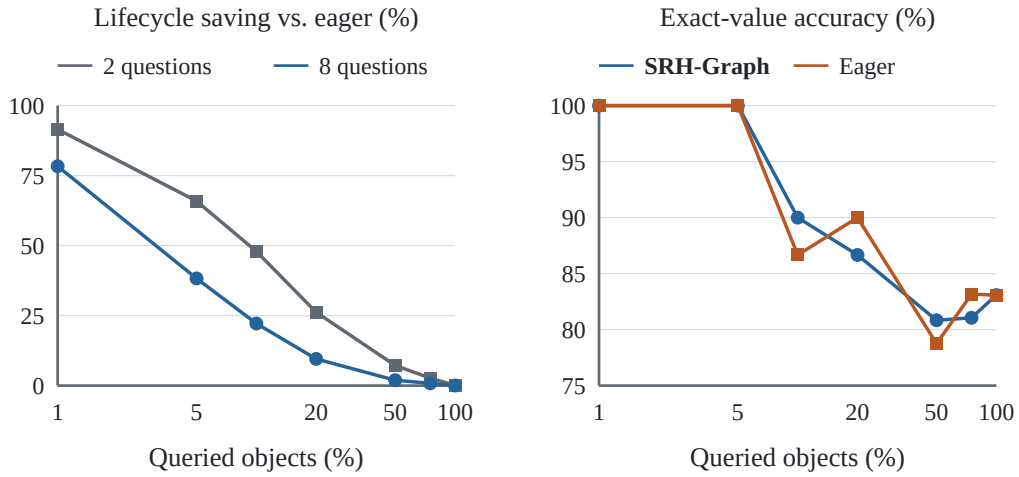

Figure 2: **SRH-Graph** across access coverage. Left: measured lifecycle LLM-token savings against matched eager construction, including answers; lines are visual guides, not fitted thresholds. Right: accuracy of the two graph policies. Coverage is the fraction of target objects queried, not the fraction of source versions constructed.

CI $[-4.39, +0.06]$. At 100%, aggregate accuracy ties, but 16 questions improve and 16 worsen; the paired interval is $[-1.33, +1.33]$. Neither the tie nor non-significant tests establish general equivalence. At 1%, only three source clusters are available: even all 24 **SRH-Graph** answers improving over **SRH-Base** yields a cluster sign-flip $p = .25$.

5.3 The cost surface and its boundary

Figure 2 shows that avoiding unused construction matters most when demand touches a small fraction of objects. At 1% coverage, **SRH-Graph** saves 96.24% of construction tokens but 78.34% of the eight-question lifecycle total. These are different denominators; neither is a remeasurement of the 96.41% card-scheduling result. Lifecycle savings decline to 38.28, 22.18, 9.53, and 1.92% at 5, 10, 20, and 50% coverage. With only two questions per hotspot, construction constitutes more of the total, so the corresponding savings are larger.

Reuse addresses a different comparison. Across the five initial coverage levels, retaining relations saves 39.84–40.73% of eight-question lifecycle tokens relative to rebuilding the graph for each question. These questions have different wordings and require separate answer calls, so the saving is not an exact-answer-cache artifact. **SRH-Graph** still uses 13.60–22.70% more tokens than Naive RAG. Reuse makes structure less expensive than repeated reconstruction; it does not make structure free or guarantee a cost advantage over a system that never constructs it.

Table 3: Final real-task scores with complete denominators. First three rows: accuracy (%); last row: set-token F1 (%). **SRH-Base** is paired in the new LoCoMo/ConvoMem evaluation, historical on sparse RepLiQA, and absent as a new arm on LongMemEval. Naive RAG outputs are historical. Failures are included as zero.

Task	n	**SRH-Graph**	**SRH-Base**	Naive RAG	**SRH-Graph** failures
LoCoMo	1,986	50.10	51.86	53.22	1
ConvoMem	657	77.63	78.08	79.76	1
LongMemEval-S	500	78.00	—	76.80	2
RepLiQA sparse	40	62.32	51.22	47.22	3

The extended boundary test is particularly informative. At 75% coverage, **SRH-Graph** uses 9,542,295 versus eager's 9,620,811 tokens, saving 0.816%. At 100%, it uses 12,632,006 versus 12,636,579, saving only 0.036%. One full-coverage seed is slightly more expensive under deferral, by 150 tokens. There is no measured aggregate crossover from which to infer an exact threshold. Moreover, full target-object coverage constructs 296, 299, and 300 source versions across the three runs, rather than necessarily all 300 in each.

Deferral also moves work onto cold queries. At 75 and 100% coverage, mean first-retrieval times are 20.24 and 17.40 seconds for **SRH-Graph**, versus 0.62 and 0.61 for eager; later retrievals are 0.63/0.56 and 0.57/0.56 seconds, respectively. Eager startup construction is accounted for separately. These measurements support a latency–construction trade-off, not a latency win for the deferred system. The update stream further tests invalidation and changing demand: **SRH-Graph** answers 56/60 source-update questions correctly, compared with 52/60 for eager. Its overall migration-stream advantage is not confirmed by the source-cluster test ($p = .125$).

6 Real-Workload Cost and Quality

Real tasks test transfer beyond the controlled workload. We compare **SRH-Graph** with available **SRH-Base** outputs and historical external configurations, keeping costs and cohort boundaries visible.

6.1 SRH-Graph is not a uniform improvement over SRH-Base

Table 3 and Figure 3 show that additional graph evidence is not harmless by construction. On LoCoMo, **SRH-Graph** answers 995 questions correctly versus **SRH-Base**'s 1,030: 55 improve and 90 worsen. The paired difference is -1.76 pp, with conversation-cluster CI $[-2.71, -0.75]$ and sign-flip $p = .0137$. The original four-task regression family was not fully executed; assigning the two unexecuted tests $p = 1$ gives a conservative four-slot Holm sensitivity of .0547. This does not establish family-wise significance, but it also does not erase the observed regression or demonstrate no loss.

On ConvoMem, **SRH-Graph** scores 510/657 and **SRH-Base** 513/657, with 24 improvements and 27 regressions. The difference is -0.46 pp, CI $[-2.62, +1.70]$, with persona-cluster $p = .787$. We cannot confirm an improvement or non-inferiority. These two conversational tasks favor retaining **SRH-Base** as a separate operating point rather than presenting **SRH-Graph** as an unconditional upgrade.

Sparse RepLiQA gives a different result. **SRH-Graph** reaches 62.32% mean F1 against historical **SRH-Base**'s 51.22% and Naive RAG's 47.22% on the same 40-question stream. The **SRH-Graph**–Naive RAG difference is $+15.09$ pp, document-cluster CI $[+6.51, +24.22]$, with Holm-adjusted $p = .00125$ in the frozen four-external-comparator family. This is evidence for that fixed sparse stream, not broad graph superiority. Its cache evolves through the sequence, so resampling documents does not create independent deployments. Three failed answers remain in the 40-question denominator.

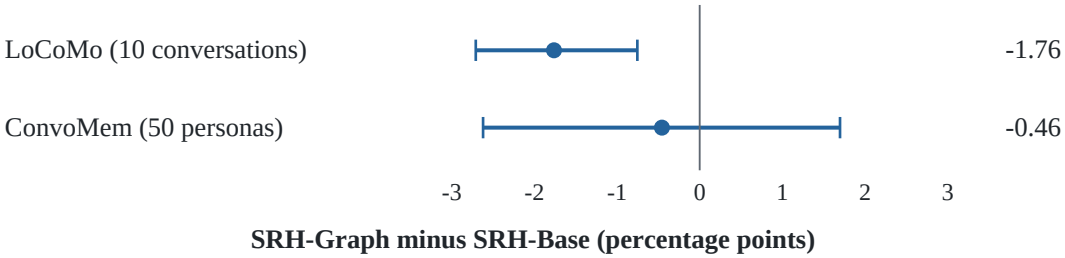

Figure 3: **SRH-Graph**–**SRH-Base** paired accuracy differences and 95% cluster-bootstrap intervals. The analysis retains all questions, including technical failures. Only these two real tasks have completed paired **SRH-Base** arms in the new evaluation package.

Table 4: Task-dependent benefits of relation memory (%). Panel A repeats all five controlled chain conditions from Table 2, with n questions per arm across three seeds. Panel B uses real-task historical comparators. Compare systems within rows, not across panels or cohorts: answer models differ, and RepLiQA uses F1 rather than accuracy. Dashes denote untested comparisons, not zero scores.

Task/cohort	n	**SRH-Graph**	Naive RAG	mem0	Graphiti	MemOS
A. Controlled synthetic multi-hop: exact-value accuracy						
Chains, 1% coverage	24	100.00	0.00	—	—	—
Chains, 5% coverage	120	100.00	3.33	—	—	—
Chains, 10% coverage	240	90.00	2.08	—	—	—
Chains, 20% coverage	480	86.67	1.67	—	—	—
Chains, 50% coverage	1,200	80.83	1.50	—	—	—
B. Real tasks: accuracy, except sparse RepLiQA (F1)						
LoCoMo	1,986	50.10	53.22	31.27	26.74	39.68
ConvoMem	657	77.63	79.76	—	—	—
LongMemEval full	500	78.00	76.80	—	—	—
LongMemEval subset A	50	84.00	—	56.00	44.00	—
LongMemEval subset B	101	78.22	—	—	—	65.35
RepLiQA sparse	40	62.32	47.22	15.71	10.00	12.79

6.2 Where relation memory helps

We bring the two workload regimes together in Table 4. On controlled chains, relation-guided retrieval addresses the missing bridges that defeat one-pass retrieval; on sparse RepLiQA, our complete pipeline also improves F1. Naive RAG's conversational strength does not remove these gains, but limits their scope. We separate synthetic and real panels because their answer models and metrics differ. The historical mem0, Graphiti, and MemOS comparisons use matched cohorts, not current releases; LongMemEval subsets must not be ranked against its full cohort.

We retain the cases where Naive RAG is stronger. On LoCoMo, **SRH-Graph** trails it by 3.12 pp, CI $[-4.66, -1.42]$; on ConvoMem the difference is -2.13 pp, CI $[-5.21, +0.91]$. LongMemEval's $+1.20$ pp difference has CI $[-2.00, +4.40]$; common-Claude scoring reverses that small ordering to -0.20 pp while preserving the larger LoCoMo comparisons' direction. Collection dates, gateways, missing-vote handling, and adjudication paths differ. We therefore make a workload-specific claim, not a general state-of-the-art claim.

6.3 The extra computation must be justified

Figure 4 shows that **SRH-Graph** costs more recorded tokens than historical Naive RAG on every real task. It also costs 2.39 times paired **SRH-Base** on ConvoMem, without confirmed accuracy improvement. Appendix C reports totals and unknown-usage reservations separately.

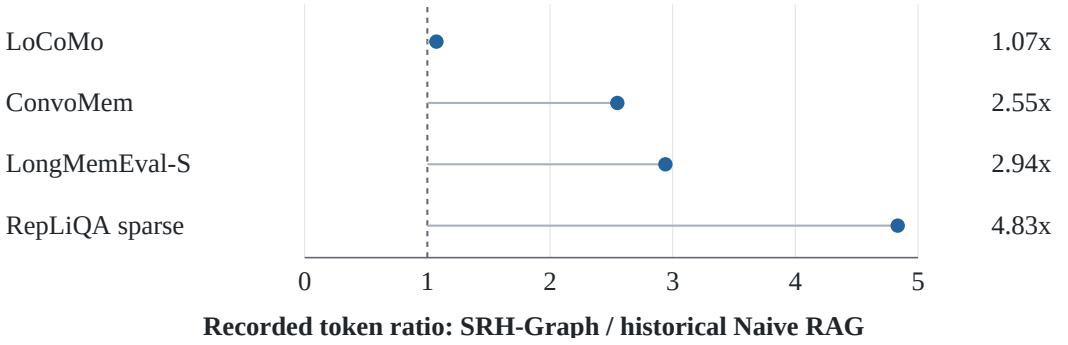

Figure 4: Recorded construction-plus-answer LLM tokens relative to historical Naive RAG on matching cohorts; the dashed line marks equal token use. Judge costs are excluded and reported separately. ConvoMem additionally retains 63,833 tokens of unknown-usage reservation for **SRH-Graph**. Ratios are neither monetary costs nor complete hardware-resource ratios.

Against historical eager pipelines, **SRH-Graph** uses 24.28% fewer recorded tokens than mem0 and 48.14% fewer than Graphiti on LoCoMo, yet 2.42 times MemOS's total. On sparse RepLiQA, it uses 89.39, 98.38, and 65.05% fewer tokens, respectively. These savings depend on implementation and workload; differing model prices, metering completeness, embeddings, and storage preclude a universal monetary-saving claim.

7 Discussion and Limitations

When to use our method. We target repeated questions requiring cross-source evidence that one-pass retrieval misses (Table 4). Otherwise, Naive RAG or **SRH-Base** may suffice. Real-benchmark aggregates do not isolate hop count, so we cannot validate a single-hop/multi-hop partition. Low coverage favors deferral, repeated local demand favors reuse, and tight cold-query latency can favor eager construction: deployment choices, not automatic routing.

What the controls establish. We isolate scheduling through fixed-artifact accounting and test expansion and caching with the two graph ablations. Our symmetric L1 follow-up compares raw, deferred-card, and eager-card pipelines (72 arm-runs; Appendix E). We do not yet isolate why graph evidence hurts some real answers: distraction and increased exposure are possible explanations, not demonstrated mechanisms.

Scope and uncertainty. We test one bounded graph recipe and three designed corpora, not production traffic. The 1% condition has only three clusters; sparse RepLiQA has 40 questions in one evolving stream. External baselines are historical, not same-date re-runs. Matched LazyGraphRAG, LazyMem, and BudgetMem comparisons remain missing. LongMemEval's small advantage depends on scoring, and general non-inferiority is not established.

The original dual-human audit covers 250 risk-enriched legacy items, not graph outputs; nine new single-reviewer resolutions cannot establish inter-rater reliability. Model identifiers are gateway-reported, not independently authenticated. Appendix E separates completed legacy controls from the unfinished four-arm exposure adjudication, which contributes no final efficacy claim here.

8 Conclusion

We introduced **SRH-Graph** to combine demand-driven relation construction with persistent reuse. Our experiments demonstrate access to missing multi-hop evidence and savings from deferral and reuse, with full-coverage and cold-latency boundaries. Real-task gains remain workload-specific, with conversational regressions and costs above raw retrieval. Alongside **SRH-Base**, our method offers a measured choice of *when to pay for structure*, not universal graph superiority.

AI USE STATEMENT

LLMs assisted sentence polishing, paragraph structuring, statistical script validation, artifact auditing, and LATEX typesetting. Separately, the evaluated systems use large language models for answering, relation extraction, and automated judging, as described in the experimental protocol. Model-generated judgments are distinguished from human annotations and are not presented as human labels.

The authors reviewed and approved the protocols, authorized data and API use, audited the saved outputs, and take responsibility for all claims and research artifacts.

ETHICS STATEMENT

Persistent memory can retain sensitive conversation content and propagate incorrect extracted relations. Source provenance enables inspection but does not itself provide privacy, consent, deletion compliance, or factual correctness. Our evaluation uses benchmark and synthetic workloads rather than a deployed user-memory service. Human-review records and API credentials should not be included in public artifacts without appropriate review and redaction. A deployment would require access control, deletion propagation, retention limits, and a policy for handling private or adversarial evidence.

REPRODUCIBILITY STATEMENT

The accompanying draft records source hashes and a claim-to-evidence map without modifying frozen experiment data. Appendices specify cohort boundaries, system configurations, scoring paths, and cost denominators. Numerical figures are regenerated offline from saved statistics and transcribed frozen mechanism tables; no new model call is required to rebuild this manuscript. The analysis package preserves unsuccessful cases and distinguishes human resolutions, automated judgments, and unknown usage.

REFERENCES

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

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

Darren Edge, Ha Trinh, and Jonathan Larson. LazyGraphRAG: Setting a new standard for quality and cost. Microsoft Research technical blog, November 2024b. URL https://www.microsoft.com/en-us/research/blog/lazygraphrag-setting-a-new-standard-for-quality-and-cost/.

Vladimir Karpukhin, Barlas Oğuz, 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. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

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*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhen Tao, Junpeng Ren, Huayi Lai, Hao Wu, Bo Tang, Zhenren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Mingchuan Yang, Tong Xu, Wei Xu, Huajun Chen, Haofeng Wang, Hongkang Yang, Wentao Zhang, Zhi-Qin John Xu, Siheng Chen, and Feiyu Xiong. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724*, 2025. URL https://arxiv.org/abs/2507.03724v1.

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

João Monteiro, Pierre-André Noël, Étienne Marcotte, Sai Rajeswar, Valentina Zantedeschi, David Vázquez, Nicolas Chapados, Christopher Pal, and Perouz Taslakian. Repliqa: A question-answering dataset for benchmarking llms on unseen reference content. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 24242–24276. Curran Associates, Inc., 2024. doi: 10.52202/079017-0764. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2b23626015b6311369e95a70735cbb72-Abstract-Datasets_and_Benchmarks_Track.html. NeurIPS 2024 Datasets and Benchmarks Track. arXiv:2406.11811.

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. URL https://arxiv.org/abs/2310.08560.

Egor Pakhomov, Erik Nijkamp, and Caiming Xiong. ConvoMem benchmark: Why your first 150 conversations don't need RAG, 2025. URL https://arxiv.org/abs/2511.10523.

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

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Jing Yu, Yibo Zhao, Jiaming Zhang, and Xiang Li. LazyMem: Retrieve broadly, construct selectively for efficient long-term agent memory, 2026. URL https://arxiv.org/abs/2607.22690. Preprint, under review.

Haozhen Zhang, Haodong Yue, Tao Feng, Quanyu Long, Jianzhu Bao, Bowen Jin, Weizhi Zhang, Xiao Li, Jiaxuan You, Chengwei Qin, and Wenya Wang. Learning query-aware budget-tier routing for runtime agent memory, 2026. URL https://arxiv.org/abs/2602.06025.

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

A Implementation and Experiment Identity

This appendix distinguishes evaluated code paths from broader architectural ideas. It also records the operational details needed to interpret existing results without changing the systems retrospectively. Exact frozen sources, prompts, and response records remain in the evidence archives referenced by the accompanying source map.

Naming. The historical SRH name expands to Substrate-pluggable Hot Memory, RAG, and Hippocampus-inspired coordination. Here, **SRH-Base** names the evaluated no-graph retrieval/packing configurations; **SRH-Graph**, described as Demand-Driven Relation Memory, names the implementation recorded as `srh_dg_v1`. The descriptor is not a new acronym expansion or a change to the evaluated code. Benchmark adapters need not have identical settings. Dense or sparse-distributed hot-memory substrates, salience decay, and cross-agent coordination are outside the evaluated contribution. The hippocampus analogy is motivation, not a validated cognitive mechanism.

SRH-Base retrieval. The original benchmark adapters use whole-session retrieval or query-guided source-span packing. The span recipe uses approximately 240-token spans with 32-token overlap; turn-centered candidates include neighbors, with a 160-token option for long turns. The recorded scorer combines BM25 with weighted dense, character-ngram, and parent scores (weights .65, .25, and .15 for the latter components). Candidate expansion and archive reranking use multipliers of three and four. Stable ties, redundancy control, and source diversity are implementation rules, not LLM-generated summaries. These details describe the legacy recipe; per-benchmark budgets and adapters remain bound to their saved runs.

Relation extraction recipe. For an admitted source version, the extractor produces source-only relations and supporting quotations, without access to the gold answer. Literal substring and offset checks reject unsupported spans. Corpus identity, source-content hash, and recipe identity bind cached results. Empty extraction results remain cached. A source update invalidates dependent entries; the updated content may be reconstructed if subsequently admitted. Relations help retrieve additional sources, but the final answer is conditioned on source quotations rather than on unsupported symbolic edges alone.

The implementation adds quotes after the base context, subject to the 16,000-character limit. It admits at most six sources and performs at most three rounds of outgoing-object expansion, with at most two entities and two search hits per entity per round. Up to twelve relations are requested per extracted source. Graph w/o traversal (the frozen nodes-only arm) retains source extraction, its cache, and quotations from initial hits, but disables edge-guided expansion; it therefore cannot certify that an off-switch reproduces **SRH-Base**. Constructed-source count, query count, and answer-cache hits are different quantities.

Naive RAG. The primary raw baseline remains the original implementation: 900-character chunks with nominal 150-character line-boundary overlap, BGE-M3 embeddings, normalized cosine ranking, and estimated-token-budget filling. The supplied `top_k` argument is not enforced in that historical path. It has no hybrid lexical score, learned reranker, query-guided span extraction, graph expansion, scope filtering, or answer cache. The enhanced-extractive raw arm used in the older cache experiment is separately named in Appendix E and is not substituted for this baseline.

Answer models and budgets. The original real-task answerer is recorded as `deepseek-v4-pro-260425`. The synthetic graph study uses the deployed Qwen3.5-27B extractor and Qwen3.5-397B answerer, with temperature zero and answerer thinking disabled. The older card-scheduling study instead uses Qwen3.5-35B-A3B extraction and a Qwen3.5-397B-FP8 answerer; its card and answer caps are 384 and 1,024 tokens. These experiments must not be interpreted as using one common extractor.

LongMemEval's effective generation cap is 2,048 tokens from the public CLI path, not the overridden class default of 1,024. Its context includes source dates and retains 13 duplicate session occurrences rather than silently deduplicating the input. The sparse RepLiQA run uses a 256-token generation cap with no empty-answer fallback, a fixed $p = .1$ stream, $R = 1$, and seed 20260728.

Its 500 documents are ingested in the prescribed serial order; the 40 questions are evaluated with an initially empty graph whose state evolves through the stream. It is not a 40-question subset sampled after running a dense graph over all questions.

B Scoring, Recovery, and Statistical Protocol

The final real-task statistics preserve answer-generation failures and distinguish them from failures of the scoring service. Existing answers are never regenerated to improve a score during this manuscript revision. We summarize the decision paths and their limits because they differ from a fully contemporaneous, fully observed panel.

Primary judging and adjudication. LoCoMo and ConvoMem contain 3,972 and 1,314 generated arm–question rows, respectively. Their three technical answer failures leave 5,283 substantive answers and 15,849 primary judge slots. Of these, 15,529 have valid primary votes and 320 are missing. Primary disagreements and incomplete panels enter the frozen blind adjudication workflow; missing votes are not treated as false. The final nine unresolved labels were reviewed by one human using the original mixed-language materials. Three were correct and six incorrect; the H02 translation issue was checked against the original text. Those labels replace only the nine unresolved values; the remaining 5,274 rows are unchanged.

LongMemEval has 498 substantive answers and two technical failures. The 1,494 primary slots contain 1,436 valid votes and 58 missing votes. Eighty-two cases require adjudication: 164 first-round and four second-round votes are valid, with 170 physical requests including other recorded attempts. No human unresolved item remains in this task. RepLiQA uses deterministic set-token F1 and does not use this judge panel. Its binary secondary score, 27/40, is not interchangeable with the main F1 result.

The primary judge families are Claude, Gemini, and Kimi; adjudication uses GPT and GLM. Frozen model identifiers and payload signatures are checked against saved responses. These are gateway-reported identities, not independent verification of the underlying hosted models. Second-round opinions use fixed first-round peer snapshots, so one judge cannot see an already-updated opinion from the other. Recovery retains failed responses, request counts, and unknown usage. Post-hoc engineering amendments are logged rather than presented as part of an unchanged initial preregistration.

ConvoMem execution stages. The completed paired cohort has 612 history-gated cases, 44 native-v3 cases, and one reviewed recovery. **SRH-Graph/SRH-Base** correct counts are 468/473, 41/39, and 1/1 within those groups. These stages reflect execution recovery, not selection by outcome, but they are not 657 fresh independent replications. Without the nine manual resolutions, the identifiable **SRH-Graph–SRH-Base** point-difference bounds are $[-1.86, -1.71]$ pp for LoCoMo and $[-1.07, -0.15]$ pp for ConvoMem. These bounds describe unknown labels, not sampling confidence intervals.

Historical score binding. External comparators retain original question IDs, questions, references, and saved answers. Historical majority labels are overwritten only where corresponding original adjudication records can be bound to those exact items. The shared-Claude sensitivity uses the common judge family to reduce one source of scoring-path variation, but does not remove collection-date or gateway drift. LoCoMo has ten conversation clusters; ConvoMem has 50 personas. LongMemEval's mem0/Graphiti comparison uses 50 cases, its MemOS comparison combines two disjoint stored batches totaling 101, and its Naive RAG comparison uses all 500.

Inference. Real-task paired bootstrap analyses use 20,000 replicates, with seed 20260924 in the recorded **SRH-Graph–SRH-Base** analysis. Exact sign-flip enumeration is used where the cluster count permits; larger cases use the frozen Monte Carlo procedure. For the stopped four-task new-versus-old family, the two unexecuted tests are set to $p = 1$ only in the reported conservative Holm sensitivity. They are not invented experiments. RepLiQA's four-external-comparator Holm family remains distinct from its descriptive historical **SRH-Base** comparison.

The synthetic primary contrasts are **SRH-Graph** versus **SRH-Base** and **SRH-Graph** versus matched eager within each coverage level. Source-cluster bootstrap uses 10,000 replicates; sign flips are exact for small cluster counts and otherwise use 100,000 Monte Carlo draws. Migration clusters have unequal numbers of questions, so the cluster-equal contrast need not equal the difference between query-weighted aggregate accuracies. The reported migration **SRH-Graph**–eager contrast is $+6.67$ pp, CI $[+1.67, +13.33]$, but its sign-flip $p = .125$ does not pass the frozen significance criterion. We retain both statistics rather than substituting a question-level test.

Recovery sensitivity. Sparse RepLiQA initially has 36 substantive outputs and four failures, with mean F1 61.63%. An approved recovery yields 37 substantive outputs and three failures, raising mean F1 to 62.32%. Thirty later attempts recover no additional answer. Both states and all failed attempts are preserved. Reporting only the 37 successful outputs would change the denominator and is not our analysis.

C Additional Resource Accounting

System cost covers source construction and answer generation; research evaluation cost is a separate ledger. Input and completion tokens are added once, with reasoning tokens already included in completion. Unknown-usage reservations are conservative accounting placeholders, not measured consumption.

Table 5: Recorded real-task construction plus answer tokens. Cohorts match within each row. These totals include attributed failed and approved retry attempts for **SRH-Graph**; historical metering completeness differs.

Task	Comparator/cohort	**SRH-Graph** tokens	Comparator tokens
LoCoMo	mem0, 1,986	8,193,069	10,820,303
	Graphiti, 1,986	8,193,069	15,799,717
	MemOS, 1,986	8,193,069	3,386,020
	Naive RAG, 1,986	8,193,069	7,629,147
ConvoMem	paired **SRH-Base**, 657	5,398,881	2,258,468
	Naive RAG, 657	5,398,881	2,118,332
LongMemEval	Naive RAG, 500	6,587,184	2,240,216
	mem0, 50	667,538	23,177,824
	Graphiti, 50	667,538	594,797,967
	MemOS, 101	1,326,234	60,402,036
RepLiQA sparse	historical **SRH-Base**, 40	588,868	108,533
	Naive RAG, 40	588,868	121,807
	mem0, 40	588,868	5,552,504
	Graphiti, 40	588,868	36,246,048
	MemOS, 40	588,868	1,684,749

ConvoMem's **SRH-Graph** reservation is 63,833 tokens (7,131 graph plus 56,702 answer); **SRH-Base** retains 53,991 unknown tokens separately. LongMemEval's physical total is 6,588,099 tokens: 6,587,184 attributable to the formal questions plus 915 from five smoke checks. LoCoMo has an additional shared-answer smoke cost of 141 tokens outside the arm totals. Shared prefix computation is physically performed once where recorded, but scientific system accounting charges the corresponding common work to both logical systems. It is not assigned only to a comparator to manufacture a saving.

Non-LLM resources. The real-task records additionally retain embedding, ingestion, base retrieval, graph expansion, and serialized storage measurements. Graph-expansion time sums are 2,981.28, 18,089.99, 14,584.14, and 1,454.17 seconds for LoCoMo, ConvoMem, LongMemEval, and RepLiQA. These sums include differing concurrency, queueing, and reconstruction conditions, not a common-hardware end-to-end latency benchmark. Embedding time is already included in the relevant ingestion/retrieval components and must not be added again. Peak single-store index bytes and graph snapshot bytes do not measure process RSS.

Table 6: Research scoring costs, excluded from system token ratios. Attempts include unsuccessful and diagnostic requests in the corresponding recorded lineage. Reservations are not confirmed usage.

Stage	Requests	Actual tokens	Unknown reservation
LoCoMo/ConvoMem primary	16,642	5,677,123	1,329,290
LoCoMo/ConvoMem adjudication	1,468	691,535	480,565
LongMemEval primary	1,863	649,820	2,022,606
LongMemEval adjudication	170	95,627	0

The synthetic six-arm collection retains 38,266 physical calls and 92,485,509 known tokens; the coverage extension retains 11,906 calls and 44,535,108 tokens. These physical totals include recorded smoke and other collection calls, whereas per-cell curves describe the formal workload. Both synthetic ledgers have zero unknown-token reservation. They were served by the experimental Qwen deployments, not by the paid DeepSeek answer API. Token totals alone do not measure energy or monetary expense.

D Full Synthetic Cost and Update Results

Separating traversal from reuse. The frozen `lazy_no_reuse` arm is labeled Graph w/o reuse: it clears the relation store between questions, but retains traversal and within-question reuse. The frozen `nodes_only` arm is labeled Graph w/o traversal: it still pays for extraction, caches it, and can append its initial-source quotations, but cannot follow relation edges to additional sources. It is neither a no-extraction baseline nor **SRH-Base**. All arms disable the final-answer cache.

Table 7: Mechanism ablations on the same five stable multi-hop streams as Table 2. Values are exact-value accuracy (%); n is questions per arm across three seeds. No new run or label is introduced by renaming the two controls.

Coverage	n	**SRH-Graph**	Graph w/o reuse	Graph w/o traversal
1%	24	100.00	100.00	0.00
5%	120	100.00	100.00	0.00
10%	240	90.00	87.50	0.00
20%	480	86.67	88.33	0.00
50%	1,200	80.83	78.75	0.50

These controls answer different questions. Near-zero accuracy without traversal tests the ability to recover missing bridges; high accuracy without reuse shows that repeatedly rebuilt relations can still support answers. Reuse is justified primarily by its 39.84–40.73% lifecycle-token saving against reconstruction (Table 8), not by an asserted uniform accuracy gain. At 20% coverage, the reuse arm is less accurate than its no-reuse control. These are mechanism diagnostics, not additional deployed SRH variants.

The synthetic tests separate unused preprocessing from repeated local work. Table 8 retains lifecycle totals rather than reporting only construction savings. The high-coverage extension and update stream add boundary evidence, but are not independent external task validation.

Object coverage and construction coverage are distinct. At 1, 5, 10, 20, and 50% object coverage, **SRH-Graph** constructs 25, 113, 184, 334, and 598 source versions out of 900 across three corpora. Bridging can require sources beyond the initially queried objects. Eager's sources not admitted in the observed stream are unused *in that stream*; this does not establish that they can never be useful. At full object coverage, the three **SRH-Graph** runs construct 296, 299, and 300 sources versus 300 per eager run.

Source-update questions are scored against the new references, while old source and graph snapshots remain available for inspection. The workload includes unrelated text, but it does not independently vary noise intensity or redundancy. We therefore make no quantitative causal claim about denois-

Table 8: Lifecycle tokens after eight distinct questions per hotspot in stable streams; migration follows its frozen turn segmentation. Dashes denote controls not run in the two-arm boundary extension.

Condition	**SRH-Graph**	Matched eager	Graph w/o reuse	Naive RAG
1%	136,110	628,444	226,264	111,757
5%	693,335	1,123,348	1,169,799	565,062
10%	1,334,541	1,714,978	2,222,243	1,126,756
20%	2,679,059	2,961,356	4,502,009	2,265,108
50%	6,435,179	6,561,277	10,781,258	5,664,706
75%	9,542,295	9,620,811	—	—
100%	12,632,006	12,636,579	—	—
Migration/update	3,044,214	3,294,014	5,038,145	2,548,342

Table 9: Hotspot migration and updates: number of correct exact-value answers. Different invalidation counts can reflect different cached-source populations rather than a failure to invalidate.

Arm	Initial /240	New hotspot /240	Updated /60	Invalidations
Naive RAG	4	4	1	0
SRH-Base	0	0	1	0
SRH-Graph	224	216	56	28
Matched eager	208	200	52	30
Graph w/o reuse	208	208	53	3
Graph w/o traversal	0	0	2	11

ing and no human-verified relation-hallucination rate. Literal provenance is an auditable support constraint, not a semantic correctness certificate.

E Legacy Controls and Human Audit

The new study extends rather than replaces the original evidence. We retain the narrow scheduling result, the symmetric-cache follow-up, and the legacy judging audit with their original scopes. Completed legacy controls and unfinished extensions are not merged.

Component-matched cards. The 100-question sample is selected with the recorded salt `iclr2027-controlled-eager-lazy-v1`. Mean all-card cost is 126,054.32 tokens, selected-card cost 3,915.94, and shared answer cost 633.24. Their sums produce 126,687.56 eager and 4,549.18 deferred tokens. The 4,100-token context ceiling is a maximum, not a target padded to capacity; median selected card context is 424 tokens. Equality of final context and answer hashes follows from reusing the same artifacts, and does not show that separately executed extraction replicas would produce identical strings.

Symmetric answer caching. The 72 completed arm-runs cross three seeds, 12 tenants, zero/eight unqueried tenants, zero/six repetitions, three systems, and L1 off/on. The systems are enhanced-extractive raw, deferred query-agnostic cards, and eager prewrite cards. They share the declared L1 semantics when enabled; no post-query writes are tested. This raw arm contains a shared extraction/packing pipeline and is not the manuscript's original Naive RAG implementation. It supplies an historical cache-control result only.

Original dual-human audit. Two raters reviewed 250 substantive legacy items: 155 LoCoMo and 95 LongMemEval, drawn from a 6,958-item pool. The sample contains 149 high-risk, 17 medium-risk, and 84 low-risk items, with risk informed by disagreement screening. Each shuffled worksheet includes the 250 substantive items, 20 sentinels, and 15 duplicates. System identities and automatic decisions are hidden. Rater A labels every substantive item; rater B abstains on one, leaving 249 double-labeled items. Each passes 20/20 sentinels and 15/15 duplicate checks.

Table 10: Symmetric-L1 follow-up, enabled-cache arms. Mean lifecycle tokens across three seeds; query-weighted exact-value accuracy. Repeated wordings are not independent questions. All disabled-cache and per-seed rows remain in the saved analysis.

Unqueried /12	Repeats	System	Tokens	Accuracy	First occurrence
0	0	Enhanced raw	25,249	75.00%	75.00%
		Deferred cards	34,121	83.33%	83.33%
		Eager cards	38,001	86.11%	86.11%
0	6	Enhanced raw	42,864	93.75%	83.33%
		Deferred cards	52,720	94.79%	86.11%
		Eager cards	55,179	96.53%	90.74%
8	0	Enhanced raw	8,411	62.50%	62.50%
		Deferred cards	16,526	75.00%	75.00%
		Eager cards	21,916	83.33%	83.33%
8	6	Enhanced raw	14,272	90.62%	75.00%
		Deferred cards	23,365	93.75%	83.33%
		Eager cards	27,656	95.83%	88.89%

The raters agree on 237/249 substantive items (95.18%); Cohen's $\kappa = .867$, with item-bootstrap CI [.787, .934]. Twelve disagreements and one abstention are not silently assigned consensus labels. The adjudicated automatic chain agrees with 207/237 consensus items (87.34%): 27 errors are automatic-wrong/human-correct and three the reverse. In the randomized low-risk slice, agreement is 81/84 (96.43%). These are conditional audit rates, not design-unbiased population accuracies.

Consensus subsets contain 85 legacy **SRH**, 99 raw RAG, and 53 MemOS items, with eight, twelve, and ten automatic mismatches, respectively. The 30 mismatched answers are short, single-line outputs without Markdown; this describes the sampled errors and does not causally establish a style bias. The audit predates the graph study and cannot validate its new outputs. The later nine-item, single-reviewer resolution described in Appendix B is a distinct process.

Other retained negative results. The original registered granularity selector did not meet its four predictions. Its held-out 400-question window–span contrasts were negative, and the high-fit slice did not establish the proposed equivalence margin. We do not turn that unsuccessful selector into a new automatic **SRH-Base/SRH-Graph** router. Older median prompt-token balancing also did not bound individual exposure differences: a small signed median can coexist with large per-question differences.

The earlier MemOS raw-padding control changes evidence content as well as exposure and retains only the successfully recovered stores; it does not isolate a pure token-budget effect. A later, distinct four-arm DeepSeek exposure collection completed 2,628 answers for 657 questions and the primary judging stage, but the available original 430-split adjudication lineage remains incomplete. The local snapshot records 339 valid first-round verdicts out of 860 required. It is not the completed LoCoMo/ConvoMem graph adjudication summarized above, and its unfinished final result is excluded from this paper's efficacy claims.

F Evidence Inventory and Interpretation Boundaries

The source map accompanying this draft binds the September 26 final real-task analysis, the September 24 mechanism and coverage-boundary reports, and the preserved original manuscript and analyses. The real-task analysis reports checks over 50 numerical assertions and 6,686 file/member hashes. Those integrity checks establish consistency with retained evidence, not the absence of methodological limitations or independent authentication of hosted models.

The full graph core is identified by source SHA-256

```
e5973a9437591c4aec7de3afbbe6434159
dfd42e0d8bed6db3e4cbbd5abdcb78
```

and the coverage-boundary frozen code/input bundle by

```
526518e835d8e3f86a2b92b35746246733
c2d7075b927389343cc36917a6beef
```

The line break is typographical and is not part of either hash. Full machine-readable paths and hashes are in the local source manifest, not in public identifying links.

No new experimental call, answer regeneration, judge invocation, model substitution, or dataset filtering is performed to create this draft. Historical full-benchmark and sparse-subset results are not silently merged. Missing experiments remain missing, and zero-score technical failures remain in their intended denominators. The primary conclusions concern the measured configurations and workloads; they do not establish universal superiority, a learned deployment policy, or biological plausibility.