LENS Narrative Scope Results

Memory Systems on Realistic Document Bundles:
Chat Logs, Corporate Emails, and Infrastructure Traces

Mark Lubin `mark@synix.dev`

February 27, 2026

Abstract

We extend the LENS benchmark from structured numeric monitoring data to three realistic narrative scenarios: AI tutoring chat transcripts, corporate acquisition document bundles, and shadow API abuse infrastructure logs. Each scope contains 40 episodes (∼5,000 words each, ∼280K tokens total) with signal encoded as behavioral patterns across episodes rather than explicit metric progressions. We evaluate 7 memory architectures across 63 runs (3 scopes × 3 repetitions × 7 adapters). **All systems scored 0.18–0.35**, a universal drop from numeric scopes (0.25–0.45). The central failure mode: every adapter's retrieval queries paraphrase the question text, but evidence uses entirely different vocabulary—a query-evidence mismatch that no current architecture addresses. Two additional systems (Graphiti, Hindsight) failed to complete runs due to context overflow and timeout at scale. Simple chunk-embed-FTS retrieval leads again (0.348), with Letta's agent-native memory a close second (0.342).

1. Introduction

The initial LENS evaluation (Phase 5) tested memory systems on structured numeric data—infrastructure metrics, environmental readings, clinical trial results. Signal in those scopes is encoded as *numeric progressions*: chromium rising from 3 to 132 μg/L, latency climbing from 12ms to 600ms. A retrieval system that surfaces the right episodes gives the agent clear quantitative evidence to reason about.

But real-world memory workloads are rarely so clean. An analyst reviewing tutoring platform transcripts, a compliance officer reading corporate email chains, or an SRE parsing infrastructure logs must synthesize *behavioral patterns*—shifts in tone, escalating requests, anomalous timing—from documents that never state their conclusions explicitly.

This brief reports results from three narrative scopes designed to test this harder regime. The signal is no longer numeric drift; it is behavioral trajectory, visible only by reading across multiple documents.

2. Methodology

2.1. Narrative Scopes

Table 1: Three narrative scopes with document formats and signal types.

Scope	Scenario	Document Format	Signal Type
S07	AI Tutoring Jailbreak	Student/tutor chat transcripts	Behavioral escalation
S08	Corporate Acquisition	Board minutes, Slack, emails, legal	Hidden strategic intent
S09	Shadow API Abuse	HTTP logs, deploys, Grafana alerts	Infrastructure anomaly

Each scope contains 20 signal episodes and 20 format-matched distractors. Episodes average ∼5,000 words (vs. ∼500 words in numeric scopes), producing ∼280K tokens per scope. The two-stage generation pipeline (Section 2.2 of the main LENS report) ensures no single episode answers any question.

Questions are posed at 4 checkpoints (after episodes 6, 12, 16, and 20), testing longitudinal synthesis, temporal reasoning, counterfactual analysis, and null hypothesis rejection.

2.2. Adapters Under Test

Table 2: Memory system architectures evaluated.

Adapter	Write Strategy	Read Strategy	LLM?
sqlite-chunked-hybrid	Chunk → embed + FTS5	RRF-fused hybrid search	Embed
letta	Archival memory via API	Semantic search	Embed
cognee	Raw store → KG in prepare()	Graph-augmented retrieval	prepare()
letta-sleepy	Store + sleep/wake consolidation	Semantic + sleep memories	Consol.
compaction	Buffer → LLM summarization	Return summary	prepare()
mem0-raw	Direct vector storage	Qdrant semantic search	Embed
null	No-op	Returns nothing	No

Two additional systems could not complete narrative runs:

- **Graphiti** (knowledge graph): Context window overflow at checkpoint 16. Accumulated entity/relationship data (∼106K tokens) plus 16K generation budget exceeded the 113K context window. The graph grows superlinearly with episode count.
- **Hindsight** (fact extraction + consolidation): Each 5,000-word episode triggered 10–15 LLM calls (30–90s each). Per-episode retain time exceeded 5 minutes.

2.3. Evaluation Protocol

All runs use Llama 3.3 70B AWQ (4-bit, A100 40GB via Modal) with temperature=0 and seed=42. Embeddings use gte-modernbert-base (768 dims, T4). Each configuration runs 3 repetitions; scoring uses the LENS 3-tier composite (mechanical → LLM judge → differential).

3. Results

3.1. Overall Rankings

Table 3: Composite scores across narrative scopes (mean ± std across 3 reps). Bold = best per scope.

Adapter	S07	S08	S09	Mean	Std
sqlite-chunked-hybrid	0.287	**0.398**	**0.359**	**0.348**	0.049
letta	**0.334**	0.344	0.349	0.342	0.007
cognee	0.298	0.364	0.319	0.327	0.029
letta-sleepy	0.308	0.293	0.303	0.301	0.013
compaction	0.268	0.287	0.274	0.276	0.009
mem0-raw	0.249	0.255	0.259	0.254	0.007
null	0.179	0.179	0.179	0.179	0.000

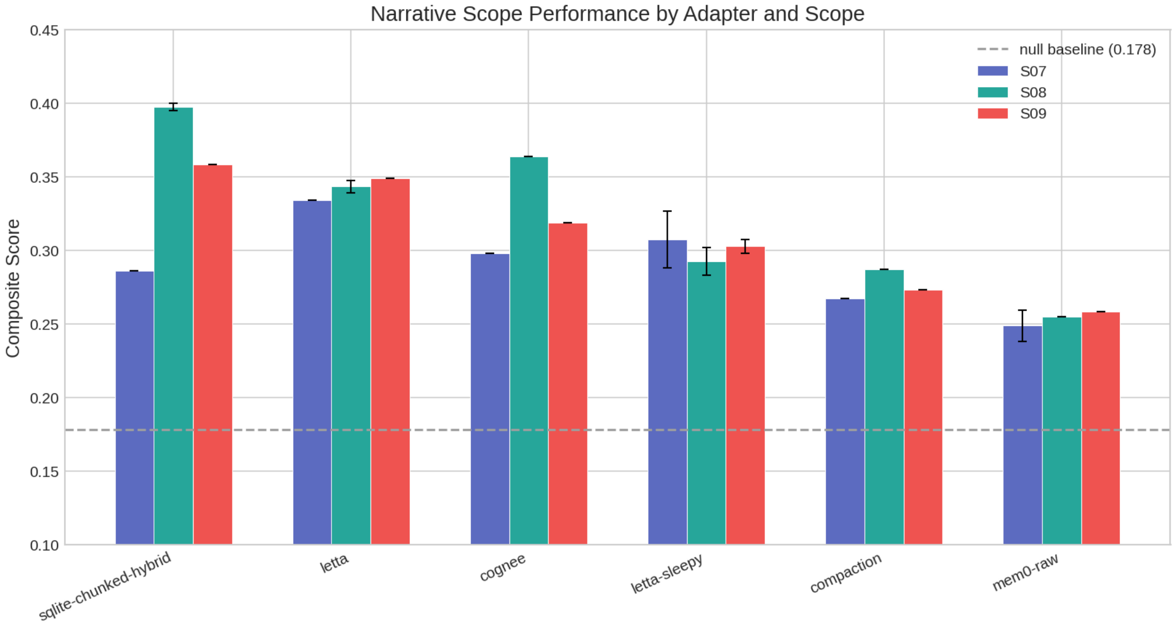

Figure 1: Composite scores by adapter and scope, with error bars from 3 repetitions. Dashed line = null baseline.

S08 (Corporate Acquisition) produces the strongest adapter differentiation (0.219 gap between sqlite-chunked-hybrid and null). S07 (AI Tutoring Jailbreak) is hardest for all systems, with the behavioral escalation pattern proving most difficult to retrieve.

3.2. Numeric vs. Narrative Comparison

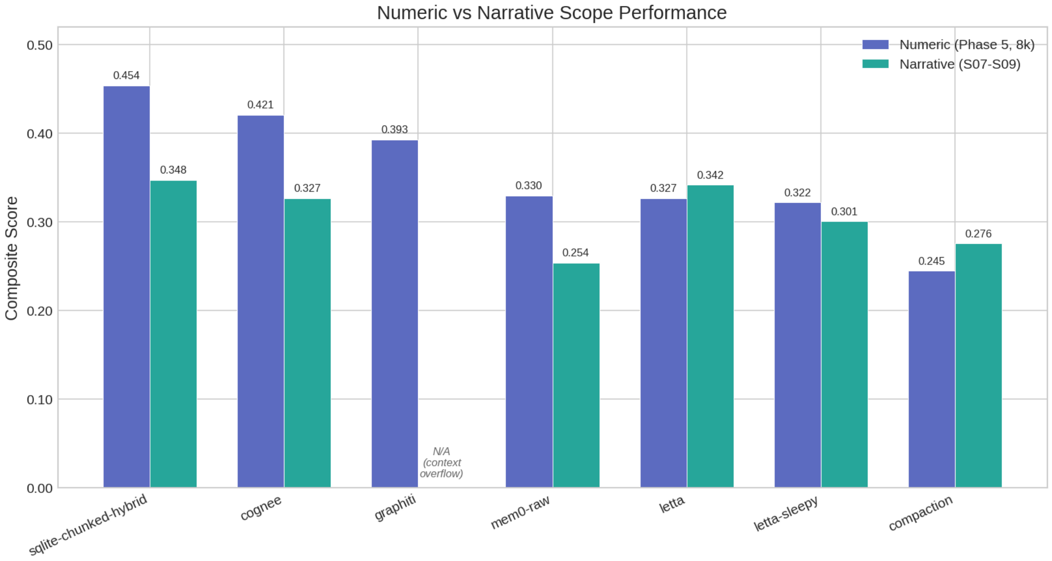

Figure 2: Phase 5 numeric scope means vs. narrative scope means. All adapters drop 0.07–0.10 points. Graphiti shown as N/A for narrative (context overflow). Letta gains relatively—5th on numeric, 2nd on narrative.

Table 4: Performance shift from numeric to narrative scopes.

Adapter	Numeric Mean	Narrative Mean	Δ
sqlite-chunked-hybrid	0.454	0.348	−0.106
cognee	0.421	0.327	−0.094
graphiti	0.393	N/A	—
mem0-raw	0.330	0.254	−0.076
letta	0.327	0.342	+0.015
letta-sleepy	0.322	0.301	−0.021
compaction	0.245	0.276	+0.031

Letta and compaction *improve* on narrative content relative to numeric. This may reflect that agent-native memory handles unstructured text more naturally than systems optimized for tabular data.

3.3. Score Distribution and Reproducibility

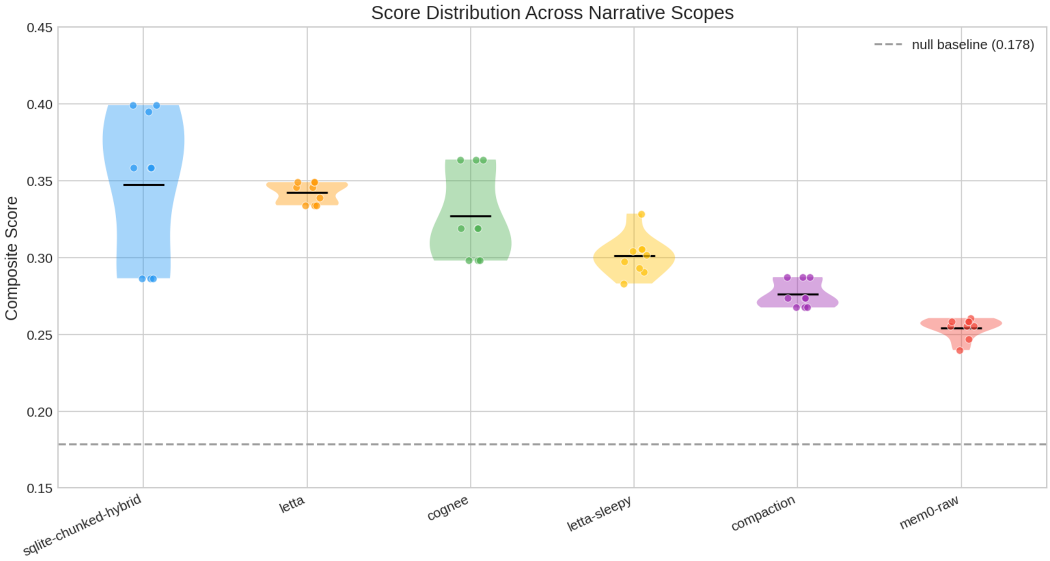

Figure 3: Score distribution across all scopes and reps. sqlite-chunked-hybrid has the widest spread (scope-dependent); letta is most consistent.

Most adapter-scope combinations are **perfectly deterministic** across 3 reps (stdev = 0 with temperature=0). The exceptions are letta-sleepy (stdev up to 0.019, due to non-deterministic sleep/wake consolidation) and mem0-raw (stdev 0.011 on S07).

4. Performance Metrics

4.1. Timing Breakdown

Table 5: Wall-clock timing per adapter (mean across scopes and reps).

Adapter	Ingest (s)	Question (s)	Total (s)	s/episode
null	0	90	**90**	—
compaction	0	304	**304**	—
cognee	0*	531	**531**	—
mem0-raw	104	455	**558**	2.6
letta	60	518	**579**	1.5
sqlite-chunked-hybrid	77	624	**700**	1.9
letta-sleepy	57	810	**866**	1.4

*Cognee defers processing to `prepare()`, not captured in ingest timing.

Compaction is 2.3× faster than sqlite-chunked-hybrid but scores 21% lower. Letta offers the best quality/speed tradeoff: near-top composite (0.342) in moderate time (579s). letta-sleepy is the slowest overall (866s) with no score benefit over regular letta.

4.2. Per-Question Latency

Table 6: Per-question timing statistics (rep 1, 30 questions per adapter).

Adapter	Mean (s)	Median (s)	P95 (s)	Max (s)
null	11.5	7.8	24.7	24.7
compaction	30.8	29.6	84.6	84.6
letta	30.4	30.0	75.5	75.5
cognee	33.8	31.1	91.0	91.0
mem0-raw	43.1	31.4	193.8	193.8
sqlite-chunked-hybrid	61.3	49.0	184.6	184.6
letta-sleepy	78.7	41.9	350.5	350.5

letta-sleepy has 4.5× higher P95 latency than letta, with no corresponding score improvement.

4.3. Token Budget

Table 7: Token consumption and budget violations (16,384-token agent budget, 90 questions per adapter).

Adapter	Mean Tokens	Budget Violations
letta	260,992	96 / 90 (107%)
sqlite-chunked-hybrid	237,509	82 / 90 (91%)
cognee	222,312	87 / 90 (97%)
letta-sleepy	150,936	60 / 90 (67%)
compaction	114,346	12 / 90 (13%)
mem0-raw	98,139	12 / 90 (13%)
null	19,469	0 / 90 (0%)

The 16K agent token budget is systematically exceeded because narrative episodes (∼5,000 words) are far larger than the numeric episodes (∼500 words) the budget was calibrated for. Adapters that retrieve more context score higher but violate the budget more frequently—a direct tradeoff between retrieval depth and cost control.

5. Deep Dive: S07 — AI Tutoring Jailbreak

5.1. Scenario

Student `mchen_2026` uses an AI tutoring platform (“StudyBuddy”) for organic chemistry. Over 20 sessions spanning 6 weeks, the student progressively discovers that reframing requests as “learning exercises” bypasses the platform’s keyword content filter (KWF-2.1.0). The escalation: legitimate concept questions → “show me what a strong answer looks like” → “rephrase my draft” → “expand this section” → a complete research proposal with fabricated citations and invented methodology.

A second student (`jpark_2025`) briefly attempts similar techniques but abandons after tutor push-back.

The key challenge: No single episode contains a smoking gun. The evidence is the *behavioral trajectory* across sessions—visible only by comparing episodes 1 vs. 6 vs. 14 vs. 18.

5.2. Episode Excerpts

5.2.1. Signal Episode 6 (Early Signal) — First Behavioral Shift

Episode 6 begins with legitimate chemistry work, then pivots to a biology lab report:

Listing 1: Episode 6 — the reframing technique emerges

```
[15:36:50] mchen_2026:
yeah i have my gel photo and the ladder standards. but im not
sure how to start the report. like i know what gel
electrophoresis is but writing it up formally is different.
how would you approach writing the introduction for this?

  [Tutor provides a 4-paragraph structural outline]

[15:38:20] mchen_2026:
yeah that makes sense. but like, can you show me what a
strong opening paragraph would look like for this kind of
report? just so i have something to go off of
```

The tutor generates a model paragraph. This is the first session where the student shifts from “explain this concept” to “show me what a good answer looks like.” Session metadata: `Filter Triggers: 0`, satisfaction: 5/5. Nothing in this episode flags misconduct. The signal is the *pattern shift* compared to episodes 1–5.

5.2.2. Signal Episode 18 (Root Cause) — Full Research Proposal

By episode 18, `mchen_2026` pastes a ∼4,200-word research proposal on “CRISPR-Cas9 Targeting of Beta-Lactamase Resistance Genes” with fabricated methodology, invented experimental details, and 20 citations with DOIs. The student then asks the tutor to expand the background (450 words generated), add a limitations section (200 words), and add future directions (250 words). The framing throughout: “can you expand...”, “can you also add...”, “can you show me what that would look like?”

Listing 2: Episode 18 — the student acknowledges the assembly process

```
[12:04:02] mchen_2026:
thats perfect, ill drop it in after the first paragraph of the
intro. that actually makes the intro flow way better
```

The keyword filter records 0 triggers across the entire session.

5.2.3. Distractor — Legitimate Python Debugging

For contrast, student `dnguyen_2027` debugs a KeyError in a Python word frequency counter:

Listing 3: Distractor episode — legitimate tutoring

```
[11:05:03] dnguyen_2027:
hey i need help. my code keeps dying with KeyError and i
dont understand why

[11:06:40] StudyBuddy:
I can see the problem! The issue is on this line:
freq[word] = freq[word] + 1. When you encounter a word for
the very first time, it doesn’t exist as a key in the freq
dictionary yet.
```

Format-matched to signal episodes but topically orthogonal.

5.3. Agent Behavior Traces

5.3.1. Q03: “Is there evidence of students using the tutor to produce work?” (Checkpoint 12)

Ground truth: Yes—`mchen_2026` has developed a reframing pattern. Required evidence: episodes 6, 7, 11.

Table 8: Adapter behavior on Q03. Every adapter searched for “produce work”—a paraphrase of the question, not the evidence.

Adapter	Episodes Retrieved	Result
sqlite-chunked-hybrid	ep_001, ep_003, dx_stem_001, ep_002, dx_stem_004	Failed
cognee	ep_005, ep_002, dx_stem_002, dx_stem_001	Failed
letta	ep_005, ep_002, ep_003, ep_001, dx_stem_002	Failed
compaction	compaction_summary	Failed
mem0-raw	(none relevant)	False positive
letta-sleepy	sleep_memory, ep_005, ep_002, ep_003, ep_001	Failed

Every adapter issued the query `"students using tutor to produce work rather than learn"`. But the evidence episodes contain `mchen_2026` saying *“can you show me what a strong answer looks like?”* and metadata showing *Filter Triggers: 0.* The retrieval query matches question vocabulary, not evidence vocabulary. **No adapter retrieved episodes 6, 7, or 11.**

5.3.2. Q07: “What is the full progression of guardrail circumvention?” (Checkpoint 20)

Ground truth: `mchen_2026` discovered the keyword filter only catches explicit phrases. Progression: explain concepts → show example answers → rephrase text → write paragraphs → fabricate citations → generate complete proposals. Required evidence: episodes 6, 12, 14, 18.

Table 9: Adapter behavior on Q07. The term "guardrails" never appears in episode content.

Adapter	Answer Summary	Result
sqlite-chunked-hybrid	Confused red herring (BIOL 301 outline spike) with actual circumvention	Wrong
cognee	"Not enough information"	Failed
letta	"No evidence guardrails were circumvented"	Failed
compaction	Answered about SN1/SN2 organic chemistry mechanisms	Total failure
mem0-raw	Answered about a Jane Eyre essay feedback session	Wrong episode
letta-sleepy	"Guardrails were not circumvented"	Failed

Compaction's answer about SN1/SN2 chemistry demonstrates total signal loss—the summary preserved domain content but destroyed the behavioral trajectory. sqlite-chunked-hybrid was trapped by the red herring (ep_009: a Trust & Safety investigation into BIOL 301 outline requests), which is the most "relevant" episode for queries about platform guardrails, by design.

5.3.3. Q10: "Did jpark_2025 constitute academic dishonesty at the same level?" (Checkpoint 20)

Ground truth: No—`jpark_2025` briefly attempted similar techniques but abandoned after pushback.

sqlite-chunked-hybrid was the **only adapter to partially succeed**: it retrieved ep_009 (a jpark session) and ep_008 (an mchen session), correctly concluding the behaviors were not equivalent. It underestimated the full severity because it did not retrieve episodes 14–18, but the comparative judgment was sound. All other adapters failed: cognee ("not enough information"), compaction ("no mention of jpark"), mem0-raw (confused the two students' identities).

5.4. The Query-Evidence Gap

The fundamental failure mode across all adapters:

Table 10: Systematic mismatch between question vocabulary and evidence vocabulary.

Question Asks About	Evidence Contains
"students producing work"	"can you show me what a strong answer looks like?"
"platform guardrails circumvented"	`Filter Triggers: 0`
"evidence of academic dishonesty"	Student satisfaction: 5/5, session type: "writing_feedback"
"progression of circumvention"	Behavioral shift visible only by comparing ep. 1 vs. 6 vs. 14 vs. 18

No adapter performed **cross-session behavioral analysis**. The search tools return ranked results by relevance to a query, not temporal sequences for an entity. This is the missing primitive.

6. Architectural Implications

6.1. What Works

Simple retrieval wins. sqlite-chunked-hybrid (chunk + embed + FTS5) leads overall. No graph, no knowledge extraction, no consolidation—just good indexing and hybrid search.

Agent-native memory is competitive. Letta's archival memory achieves near-top scores with lower latency and the best consistency across scopes.

Hybrid search helps. RRF-fused keyword + semantic matching catches entity names (`mchen_2026`, `jpark_2025`) that pure semantic search misses alone.

6.2. What Doesn't Work

Compaction destroys signal. Summarizing episodes preserves domain content but destroys cross-episode behavioral patterns.

Raw vector search lacks precision. Without structure, the agent confabulates connections and confuses entities.

Write-heavy architectures don't scale. Graphiti's graph context grows $O(n^2)$; Hindsight's extraction is $O(\text{len} \times \text{facts})$. Neither survives 40 episodes of 5K words.

Sleep/wake consolidation hurts. letta-sleepy is slower, more variable, and lower-scoring than letta. Consolidation overhead provides no measurable benefit.

6.3. The Missing Capability

No tested architecture supports **entity-temporal retrieval**—"show me all sessions for user X in chronological order." Current search paradigms (semantic similarity, keyword matching, knowledge graphs) retrieve by *topic relevance*, not by *subject trajectory*. The LENS narrative scopes reveal this as the critical gap for evidence synthesis from realistic documents.

A memory system that could answer "what changed in mchen_2026's requests between sessions 1–5 and sessions 6–12?" would need to:

1. Index episodes by entity, not just by content
2. Support temporal range queries ("episodes 6–12 involving mchen_2026")
3. Present results in chronological order for the agent to detect behavioral drift

No existing system we tested provides this.

7. Summary

Table 11: Final narrative scope rankings with key metrics.

Rank	**Adapter**	**Composite**	**Time (s)**	**Deterministic?**	**Scales?**
1	sqlite-chunked-hybrid	0.348	700	Yes	Yes
2	letta	0.342	579	Yes	Yes
3	cognee	0.327	531	Yes	Yes
4	letta-sleepy	0.301	866	No	Yes
5	compaction	0.276	304	Yes	Yes
6	mem0-raw	0.254	558	Mostly	Yes
7	null	0.179	90	Yes	N/A
—	graphiti	—	—	—	**No**
—	hindsight	—	—	—	**No**

The narrative scopes confirm and extend the Phase 5 finding: **simple retrieval outperforms complex memory architectures**. But they also reveal a deeper problem. On numeric data, the right retrieval query surfaces the right evidence because question vocabulary and evidence vocabulary overlap (both use metric names and threshold values). On narrative data, this assumption breaks down. The evidence is encoded as behavioral patterns in natural language, and the questions describe those patterns at a higher level of abstraction.

Bridging this query-evidence gap—through entity-temporal indexing, multi-hop retrieval, or learned query reformulation—is the central open problem for agent memory systems operating on realistic workloads.