Lightweight LLM Agent Memory with Small Language Models

LightMem: bounded online selection, asynchronous consolidation

Paper by Jiaquan Zhang et al.
Independent paper-reading presentation

arXiv:2604.07798v3 · revised 22 April 2026
Read on 27 September 2026 · about 25 minutes + backup

The expensive part of memory need not run every turn

Retrieval-driven memory	Repeated large-model operations
Small online overhead, but weak queries and noisy candidates can hurt answer quality.	Flexible memory decisions, but planning, filtering and writing add online latency.

- Use small language models for constrained, frequent decisions.
- Move heavier abstraction and integration to an offline LLM.
- Improve the retrieval and writing pipeline together.

Paper, Secs. 1–2, PDF pp. 1–3. This is a system-design contribution.

The three stores separate information by role

STM: immediate context C_t

Recent turns in the prompt. Neither persisted nor externally retrieved.

MTM: user-specific episodes M_u^{MTM}

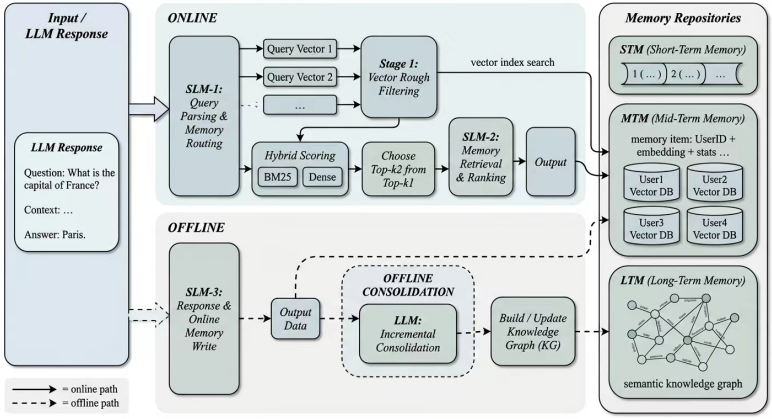
Summaries + embeddings + time/access metadata + user ID. Capacity B.

LTM: shared semantic knowledge M^{LTM}

De-identified abstractions in a graph; no raw personal episodes or user IDs.

Paper, Sec. 3.2, PDF p. 3; Appendix 8.1.1, PDF p. 11. LTM is not a personal transcript archive.

LightMem separates retrieval, writing and consolidation



Original Fig. 2, PDF p. 4, Zhang et al. (CC BY-NC-SA 4.0). The text defines SLM-3 as online writing, despite its placement in the diagram's OFFLINE area. BM25 details are not specified in the method.

SLM-1 turns an ambiguous request into a retrieval plan

$$\mathcal{Q}_t = \left\langle \{q_t^{(i)}\}_{i=1}^{n}, \phi_t, K \right\rangle$$

- Infer intent from user input x_t and recent context C_t.
- Generate hypothetical queries (HQs); route to MTM or LTM.
- Set metadata filters ϕ_t and allocate a final budget K.

Paper's example: "Recommend a dinner spot."

MTM: dietary preferences and constraints.
LTM: general restaurant knowledge. No claim of live restaurant data.

Paper, Sec. 3.3, Eq. (1), PDF p. 3; Table 1, PDF p. 5; prompt in Sec. 10.1, PDF p. 13.

A bounded candidate pool makes semantic selection practical

Stage 1: coarse retrieval

Vector search with metadata filters.

$$K_1^{(i)} = \frac{2K}{n}, \quad C = \bigcup_i C^{(i)}$$

Target: roughly $2K$ candidates.

Stage 2: SLM-2 selection

Check semantic support for the HQs.

$$R_t \subseteq C, \qquad |R_t| \leq K$$

Select entries; do not rewrite them.

Algorithm 1 rounds quotas up and truncates excess candidates. Duplicate or missing hits can leave fewer than $2K$ distinct entries.

Paper, Sec. 3.4, Eqs. (3)–(5), PDF pp. 3–4; Algorithm 1, p. 11; Sec. 10.2, pp. 13–14.

SLM-3 converts a completed interaction into reusable episodes

$$(x_t, y_t, C_t) \longrightarrow \text{compact summary} \longrightarrow M_u^{\mathrm{MTM}}$$

- Write after the response y_t has been generated.
- Merge redundant records; resolve conflicts using time and evidence.
- Evict stale or low-utility entries under a capacity bound:

$$\left|M_u^{\mathrm{MTM}}\right| \leq B, \qquad B = 10^4 \text{ entries.}$$

Reading takeaway

A fixed number of records bounds storage growth; it does not guarantee that the right facts survive.

Paper, Sec. 3.5, Eq. (6), PDF p. 4; Algorithm 2, p. 12. Utility and conflict thresholds are not specified.

The offline LLM abstracts episodes into a shared graph

1. Select new, reactivated or capacity-flagged MTM records.
2. Abstract and de-identify; find nearby semantic anchors in LTM.
3. Insert, link, merge or update knowledge; decay weak evidence.

Reported maintenance

Every 10–15 turns
∼3.5 s per offline batch
∼1 new node per 4 turns

Cost interpretation

Asynchronous execution removes a dependency from the response path. It still consumes compute and requires resource scheduling.

Paper, Sec. 3.6, PDF pp. 4–5; Algorithm 3, p. 12; Table 10, p. 16. Specific offline model not reported.

Query expansion can spend the budget on the wrong intent

A recall request

"What did I decide about the project timeline?"

Useful HQs → MTM

Identify the project.
Retrieve the user's actual decision.

Distracting HQ → LTM

Retrieve general planning principles.
Uses part of the same Top-K budget.

The answer may remain correct but include unnecessary advice. A selector cannot recover evidence absent from its candidate pool.

Failure example: Paper, Sec. 9.2, PDF pp. 12–13. Last sentence is the reader's pipeline interpretation.

Evaluation covers quality, latency and memory failures

- **LoCoMo**: five question categories over long conversations.
- **DialSim**: long-running, multi-party dialogue; lexical and semantic scores.
- **Baselines**: LoCoMo, ReadAgent, MemoryBank, MemGPT, A-MEM.

Reported implementation

Quantized Llama-3.2-1B (default); Qwen2.5-1.5B variant.
384-d MiniLM embeddings; 2,000 LoRA samples for SLM-2.
One RTX 4090 (24 GB); remote GPT-4o-mini for latency tests.

Paper, Sec. 4.1, PDF p. 5. Protocol caveat: a fixed GPT-4o-mini generator is stated, but Table 2 uses multiple backbone labels without fully clarifying which components vary.

LightMem improves F1 across the LoCoMo categories

GPT-4o-mini block	A-MEM F1	LightMem F1	Gain
Single-hop	44.65	45.81	+1.16
Multi-hop	27.02	28.85	+1.83
Temporal	45.85	46.28	+0.43
Open-domain	12.14	13.52	+1.38
Adversarial	50.03	54.57	+4.54

- All gains are absolute F1 points, not relative percentages.
- Full-history LoCoMo still scores **69.23** on adversarial questions.

Paper, Table 2, PDF p. 6. Temporal F1 is 46.28 here; the statistical appendix reports 46.20 separately.

The headline gain is consistent with a simple table average

Table 2 model label	Mean F1 gain	A-MEM tokens	LightMem tokens
GPT-4o-mini	+1.868	2,520	1,150
GPT-4o	+2.084	1,216	685
Qwen2.5-1.5B	+2.142	1,300	728
Qwen2.5-3B	+3.116	1,137	611
Llama-3.2-1B	+2.792	1,376	754
Llama-3.2-3B	+2.722	1,126	642

Reader's calculation: mean gain across 30 cells = +2.454.
GPT-4o-mini effective context is 54.4% shorter than A-MEM.

Recomputed from Table 2, PDF p. 6. Equal weighting of cells, not example-weighted F1; tokens describe effective context, not total memory storage. Model-role ambiguity remains.

On DialSim, lexical and semantic scores both improve

Metric ↑	A-MEM	LightMem	Absolute gain
F1	3.45	4.12	+0.67
BLEU-1	3.37	3.95	+0.58
ROUGE-L	3.54	4.20	+0.66
ROUGE-2	3.60	4.15	+0.55
METEOR	2.05	2.48	+0.43
SBERT	19.51	23.40	+3.89

SBERT rises by 19.9% relative to A-MEM, on the table's score scale.
Embedding similarity is not a factual-accuracy or privacy guarantee.

Paper, Sec. 4.2 and Table 3, PDF pp. 6–7; GPT-4o-mini. Relative percentage is the reader's calculation.

MTM and semantic filtering both contribute to the full system

DialSim ablation	F1 ↑	Drop from full
Full LightMem	4.12	—
Without semantic reranking	3.83	0.29
Without HQ and routing	3.87	0.25
Without MTM	3.75	0.37
Without offline consolidation	3.96	0.16

- Removing the graph also lowers SBERT: 23.40 → 22.82.
- Component removals support incremental value in this setup; they do not establish interaction effects.

Exact values from Sec. 4.3, PDF p. 7, Llama-3.2-1B configuration. Fig. 3 on p. 8; both original panels in backup.

Latency improves over A-MEM, with a quality–speed tradeoff

Method	Retrieval P50/P95 (ms)	End-to-end P50/P95 (ms)
LoCoMo	0 / 0	2,054 / 3,658
ReadAgent	34 / 97	614 / 1,027
MemoryBank	16 / 51	439 / 1,047
MemGPT	143 / 451	2,087 / 3,451
A-MEM	856 / 1,583	914 / 3,682
LightMem	**83 / 167**	**581 / 1,325**

- Versus A-MEM: retrieval P50 −90.3%; end-to-end P50 −36.4%.
- MemoryBank is faster. End-to-end includes the remote API.

Paper, Sec. 4.4, PDF pp. 7–8; Table 4, p. 8. Percentages recomputed. P50 is a median; offline work is not a zero-cost operation.

Natural MTM growth widens the advantage over vector search

MTM entries	Vector retrieval F1	LightMem F1	Gain
∼100	3.95	3.98	+0.03
∼1,000	3.90	4.05	+0.15
∼5,000	3.86	4.09	+0.23
10,000	3.83	4.12	+0.29

A useful trend, with a confound

These are cumulative scores along the same dialogue trajectory. Memory size, dialogue stage and accumulated evidence change together.

Paper, Sec. 4.5 and Table 5, PDF p. 8; DialSim, Llama-3.2-1B. This is not an independent memory-capacity sweep.

Errors can compound across planning, selection and writing

DialSim perturbation	F1 ↑	SBERT ↑
Full LightMem	4.12	23.40
50% irrelevant HQs (SLM-1)	3.95	23.08
No reranking (SLM-2)	3.83	22.56
50% noisy writes (SLM-3)	3.78	22.45
All three perturbations	1.85	11.20

Reading takeaway

A bounded retrieval budget does not prevent memory pollution. Combined perturbations reduce F1 by 55.1% relative to full LightMem.

Paper, Sec. 4.6 and Table 6, PDF pp. 8–9. Fixed final Top-K. Artificial perturbations, not measured deployment error rates.

Reading both stores helps bridge offline update gaps

LoCoMo retrieval setting	Multi-hop F1 ↑
Full LightMem, normal gap	28.85
LTM only	19.45
MTM only	20.12
MTM noise saturation	23.10

- Recent unconsolidated episodes and older abstractions can be complementary.
- Heavy recent noise can still occupy the fixed retrieval budget.
- Gap duration and the exact saturation construction are underspecified.

Paper, Sec. 4.7 and Table 7, PDF p. 9; GPT-4o-mini. Mechanism is the authors' interpretation, not a traced causal decomposition.

The appendix supports two gains, with a numerical mismatch

Category	A-MEM F1	LightMem F1	Δ [95% CI]	p
Multi-hop	27.02	28.85	1.83 [1.24, 2.41]	.001
Temporal	45.85	46.20	0.35 [0.12, 0.58]	.008

- Three seeds; 1,000 paired bootstrap resamples over instances.
- Main Table 2 reports Temporal F1 **46.28**; Table 8 reports **46.20**. No reconciliation is given.
- Evidence is specific to the reported comparisons, not all ablations or latency measurements.

Paper, Sec. 9.1 and Table 8, PDF p. 12. Values align with the GPT-4o-mini block; Table 8 does not independently name the backbone.

The strongest conclusions stay within the reported setup

1. **Model roles need clarification.** Multiple backbone labels coexist with the statement of a fixed GPT-4o-mini response generator.
2. **The cost account is partial.** Offline compute, write latency and concurrent resource contention are not fully measured.
3. **Privacy and scale are design goals.** User filters and de-identification have no reported leakage audit; fixed K does not bound all index or model costs.

Reader's assessment of Secs. 3–4, 8.1.6 and 10–11. Authors' Sec. 6 (PDF p. 9) notes unexplored consolidation and control alternatives.

Three ideas are worth carrying into memory-system design

1. Match the model to the decision

Small models can perform structured online control and selection; heavier abstraction can run asynchronously.

2. Make evidence selection an explicit interface

Budget coarse candidates, verify semantic support, then expose a compact memory set to the response generator.

3. Evaluate the full memory lifecycle

The reported gains over A-MEM are consistent. Error accumulation, update gaps and background costs remain central constraints.

Synthesis from the paper's method and results. A useful follow-up is a matched-generator, matched-token-budget replication.

Source, scope and reproducibility materials

Lightweight LLM Agent Memory with Small Language Models

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Zhenzhen Huang, Pengcheng Zheng, Zhicheng Wang, Ping Guo, Fan Mo, Sung-Ho Bae, Jie Zou, Jiwei Wei, Yang Yang.

- arXiv:2604.07798v3; revised 22 April 2026. The arXiv record states acceptance to ACL 2026 main.
- Read from same-version author TeX and the 16-page PDF, including substantive appendices. Locators use PDF page numbers.
- Original figures credited to the authors, CC BY-NC-SA 4.0. Recomputed values are labeled.
- Companion Chinese notes, figure provenance and editable TeX are archived together. Build with `python3 build_slides.py slides.en.tex`.

This is an independent reading presentation; it is not an author presentation or a reproduction of system experiments.

Backup: narrow interfaces define each model's responsibility

SLM-1	Input: query + recent context. Output: HQs/routing in JSON. Does not search or answer.
SLM-2	Input: HQs + candidate summaries/metadata. Output: selection in JSON. Does not rewrite or expand memory.
SLM-3	Input: current interaction. Output: short reusable summaries. Does not retain full transcripts or perform cross-user abstraction.
Offline LLM	Input: selected MTM batch. Output: new/updated LTM units and links. Does not directly modify MTM.

Paper, Sec. 10, PDF pp. 13–14. Exact JSON schemas and host-side maintenance interfaces are not fully specified.

Backup: LTM graph schema and its maintenance footprint

Schema

Nodes: Entity, Concept.

Edges: IsA, HasProperty, RelatedTo, Implies.

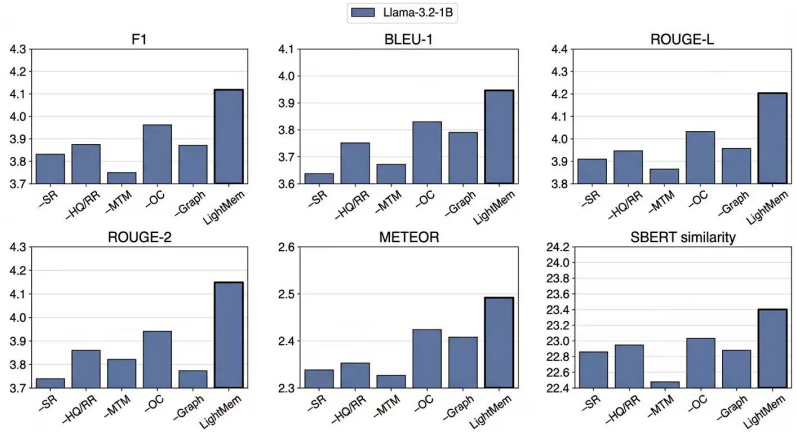
Reported dynamics

Update: every 10–15 turns.
Growth: $\sim$1 node / 4 turns.
Processing: $\sim$3.5 s / batch.

- Offline-update ablation: F1 4.12 $\rightarrow$ 3.96 (about 3.9% relative loss).
- Query-time graph traversal and the offline model are not fully specified.
- Table 10 calls 83 ms an average; Table 4 explicitly reports P50.

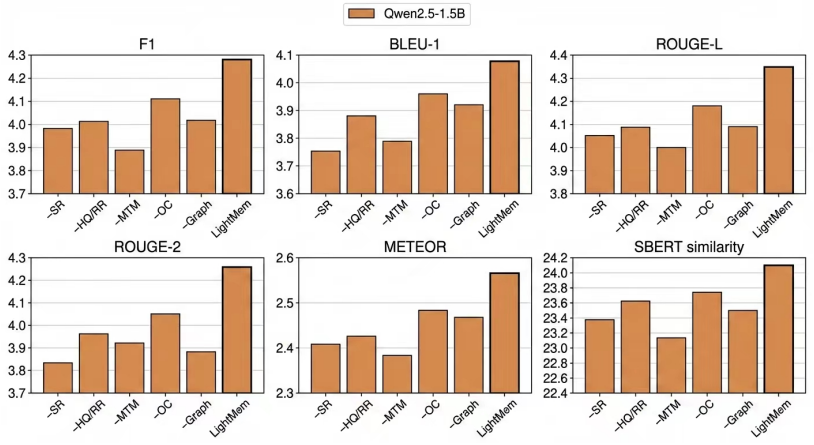
Paper, Sec. 11 and Tables 9–10, PDF pp. 15–16; F1 values match the DialSim ablation.

Backup: original Llama-3.2-1B ablation panel



Original Fig. 3(a), PDF p. 8. Truncated, metric-specific y-axes. SR: semantic reranking; HQ/RR: query generation/routing; OC: offline consolidation. Original figure: Zhang et al., CC BY-NC-SA 4.0.

Backup: original Qwen2.5-1.5B ablation panel



Original Fig. 3(b), PDF p. 8. Full system scores highest in each panel; axes are truncated and use different scales. Bars lack printed point values. Original figure: Zhang et al., CC BY-NC-SA 4.0.

Backup: a fixed item budget is not a complete complexity bound

What is explicitly bounded?

Algorithm 1 forms a candidate pool, truncates it to at most $2K$, then returns at most K entries.

- The method writes $|C| = 2K$, but duplicates or missing hits can reduce the actual count.
- The implementation reports coarse top-10. Interpreting this as $2K = 10$ gives $K = 5$, but a universal mapping is not established.
- Appendix 8.1.6 claims $O(K)$ retrieval. Index size, item length, HQ count and model context still affect end-to-end work.

Paper, Secs. 3.4 and 4.1.4, PDF pp. 3–5; Algorithm 1 and Sec. 8.1.6, pp. 11–12. Qualifications are the reader's analysis.