MEMBOX: WEAVING TOPIC CONTINUITY INTO LONG-RANGE MEMORY FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term dialogue memory often turns interactions into separate records, leaving retrieval and linking to recover the surrounding exchanges needed to interpret them. We introduce *Membox*, a memory architecture that organizes local dialogue at construction time and anchors cross-session access in that organization. **Topic Loom** incrementally forms topic-continuous boxes, preserving original exchanges alongside extracted indexing features. Retrieving a box exposes its supporting messages without separately matching each one. **Trace Weaver** links events across boxes into overlapping traces and expands evidence through a box-to-event-to-trace access procedure. On LoCoMo with GPT-4o-mini, Membox-Box improves F1 by 13.00 points over the best tested retrieval-depth setting of our local Mem0/A-MEM implementations (53.95 vs. 40.95); trace expansion raises F1 to 55.28 with additional context. Gains also hold with GPT-4o, and semantic evaluation complements the lexical metrics. Supplementary DialSim results extend the evaluation to multi-party dialogue. These findings support the effectiveness of the combined organization and access design.

1 INTRODUCTION

Remembering an interaction requires preserving how its parts belong together. Cognitive research describes how continuous experience is organized into structured events: event segmentation theory models the maintenance of an ongoing event representation and its updating at boundaries, connecting this organization to subsequent memory encoding (Zacks et al., 2007). Discourse theory similarly explains coherent interpretation through the interaction of linguistic structure, communicative intentions, and attention (Grosz & Sidner, 1986). In dialogue, a question, a reply, and a qualification may therefore belong to one unfolding exchange despite differences in their surface semantics. These perspectives motivate a construction question for agent memory: what should be preserved together as a retrievable unit?

Local organization is only part of this problem. The event-indexing model of narrative comprehension describes connections among events along time, space, protagonists, causality, and intentionality (Zwaan et al., 1995). Drawing on these accounts, we formulate two complementary requirements for dialogue memory: preserve supporting exchanges within local units, and provide access between related experiences across unit boundaries. The resulting design must balance local coherence with cross-episode continuity as dialogue accumulates.

Existing memory systems use enrichment, updates, indexing, and association to maintain useful representations and connections (Zhong et al., 2024; Xu et al., 2025; Chhikara et al., 2025). These mechanisms can encode context, but retrieving a relevant record does not necessarily expose its surrounding exchange. Figure 1 illustrates the consequence: a question about continuing to run may require a training plan, recurring knee pain, and advice to reduce intensity. Retrieving the training and pain records can still omit the advice, whose importance comes from its role in the exchange rather than a strong direct match to the query. The facts may remain stored, while their joint availability depends on additional retrieval or linking.

Membox translates these requirements into a *context-organization layer* coupled to cross-box evidence access. Topic segmentation and cross-memory association are established components (Fang et al., 2026; Ong et al., 2025; Xu et al., 2026). Our design couples them through online original-text boxes: extracted features index each box, and events within retrieved boxes provide entry points to

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.

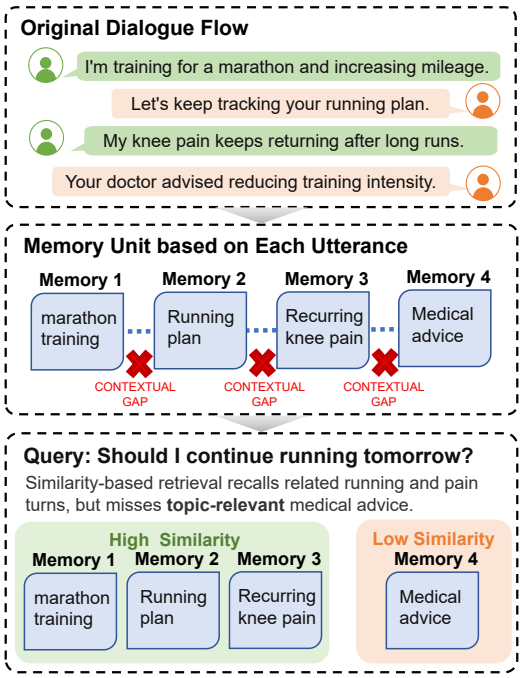


Figure 1: Illustrative context fragmentation. Separate retrieval can recover running and pain records while omitting advice from the same exchange. Preserving the exchange in one box makes these messages jointly available when that box is retrieved.

related traces. Preserving an exchange within the retrieval unit lets its supporting messages be read together without separately matching each one.

Topic Loom constructs the local units as messages arrive, using topic continuity without access to future turns. Each box retains consecutive original messages alongside extracted indexing features. The features locate an exchange; the returned original text preserves qualifications and details that extraction may omit. Local organization thus becomes a persistent part of memory construction and retrieval.

Trace Weaver supports continuity beyond these local units. It links events from different boxes into overlapping traces, allowing related activities and plans to remain connected across topic breaks and sessions. Retrieval selects boxes, uses their query-relevant events to reach traces, and adds the associated event evidence to the boxes' original text. This box-to-event-to-trace path extends access without requiring every related experience to be stored in one enlarged local unit (Figure 2).

Our contributions are threefold:

- We formulate a memory-organization perspective that connects local evidence preservation with cross-episode access, motivated by event segmentation, discourse structure, and situation models.
- We propose *Membox*, coupling online original-text box construction with event associations and a box-to-event-to-trace access path to implement these complementary requirements.
- On LoCoMo, box retrieval outperforms the tested local baselines, and trace expansion further improves answer quality with additional context. Semantic evaluation, quality–cost analyses, and retrieval cases characterize these findings; supplementary DialSim results extend the evaluation to multi-party dialogue.

2 Related Work

2.1 Long-Term Memory for LLM Agents

Long-term agent memory combines persistent storage with mechanisms for access and maintenance. MemoryBank (Zhong et al., 2024) and Ret-LLM (Modarressi et al., 2023) use embedding-based indexing, while MemGPT (Packer et al., 2023) and SCM (Wang et al., 2023) introduce hierarchical or controller-based management. Mem0 (Chhikara et al., 2025) evolves memory items, ReadAgent (Lee et al., 2024) compresses contexts into gists, and A-MEM (Xu et al., 2025) supports agent-driven memory operations. Membox complements these mechanisms by coupling the organization of original dialogue with cross-session evidence access.

Topic, scene, and event organization. LightMem combines buffered topic segmentation, summary-based entries, and sleep-time consolidation (Fang et al., 2026). THEANINE links memories through temporal and cause-effect relations (Ong et al., 2025), while StructMem preserves event-level bindings and induces cross-event connections (Xu et al., 2026). CAST aggregates time/place/topic scenes into character profiles and combines episodic and graph-based semantic memory (Ma et al., 2026). These approaches organize experience at different levels; Membox makes consecutive original-text boxes the local units and uses their events to access overlapping cross-box traces.

Context preservation and reconstruction. E-mem retains uncompressed contexts in assistant agents that reason locally before a master agent aggregates evidence (Wang et al., 2026). AtomMem extracts self-contained atomic facts, organizes them into event structures and temporal profiles, and retrieves through an associative graph (Yao et al., 2026). Membox shares the goal of context-sensitive access, retaining local exchanges as directly retrieved units and extending them through events in the selected boxes.

2.2 Discourse Topic Segmentation

Discourse segmentation identifies coherent topical spans. Classical methods use lexical cohesion and local topic shifts (Hearst, 1997; Galley et al., 2003); later work develops neural boundary predictors (Koshorek et al., 2018; Arnold et al., 2019; Lukasik et al., 2020) and supervised dialogue benchmarks (Jiang et al., 2023a). Membox adapts this perspective to online memory writing, assigning incoming turns before future context is available. Its output is a persistent memory object for retrieval and cross-episode linking, extending the role of segmentation beyond boundary prediction.

2.3 Retrieval-Augmented and Structured Context

Retrieval-Augmented Generation grounds outputs in retrieved evidence (Lewis et al., 2020; Gao et al., 2023); reflection, active retrieval, and graph traversal further guide evidence access (Asai et al., 2024; Jiang et al., 2023b; Linders & Tomczak, 2025; Sun et al., 2024). REMem builds a hybrid graph of time-aware gists and facts and uses tool-guided iterative retrieval for episodic reasoning (Shu et al., 2026). Membox uses a box-to-event-to-trace access path: box retrieval supplies local original dialogue, and selected events extend access to related trace evidence.

3 Method

3.1 Topic Loom: Building the Organization Layer

Topic Loom constructs the context-bearing units used by indexing and retrieval. A *topic-continuous dialogue episode* is a consecutive span developing a local discourse topic, including exchanges with different surface semantics. Topic Loom forms such episodes online and stores each as a box of original messages plus indexing features. This preserves local discourse in the memory object itself.

We use the two most recent messages in the active box as the default continuity window. A message is one speaker utterance; boxes are sealed at session boundaries. Upon arrival of a new message

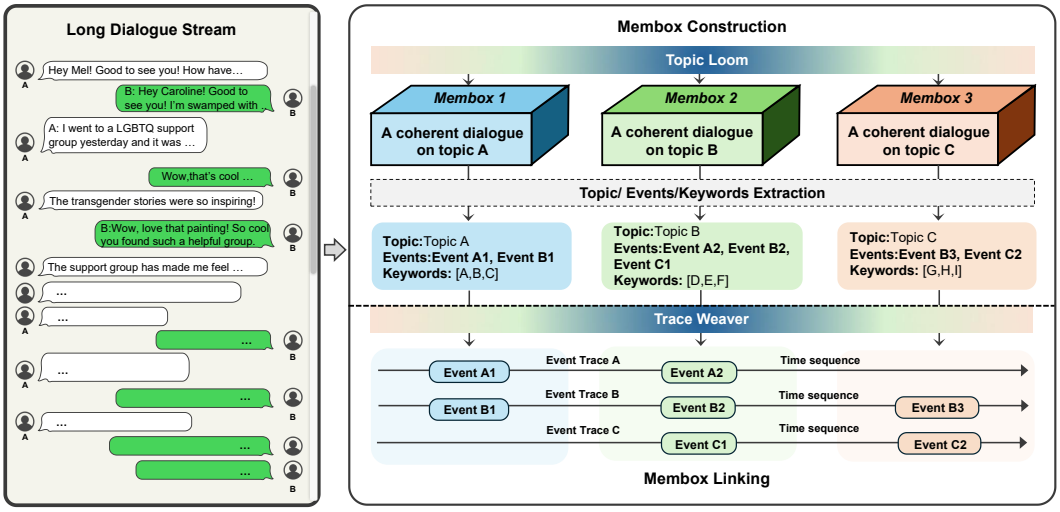


Figure 2: Membox architecture. Topic Loom builds the context-organization layer as original-text boxes. Trace Weaver reinforces it with cross-box event links, supporting evidence access from boxes through their events to related traces.

M_{k+1}, the Loom queries an LLM:

$$c_{k+1} \leftarrow \mathrm{LLM}\big(\text{window},\ M_{k+1},\ P_{\text{cont}}\big),$$

where P_{cont} is the classification prompt shown in Appendix Table 14, and $c_{k+1} \in \{\text{continuous},\ \text{partial shift},\ \text{discontinuous}\}$. In practice, most segment-worthy transitions are labeled as partial shifts rather than discontinuities. This is expected in continuous multi-turn conversations, where new topics often emerge through residual links to prior context rather than abrupt resets. For memory construction, partial shifts are treated as topic breaks because they indicate a noticeable change in conversational focus, even when the current turn remains locally related to previous turns.

If the label is *continuous*, the message is appended to the current box. If it is *partial shift* or *discontinuous*, the current box is sealed, and a new unsealed box is created with M_{k+1} as its first entry. Within a session, the next message joins a one-message box before continuity classification resumes. This retains a minimal local exchange when available; a session-ending box may contain only one message.

When a box transitions to the sealed state, the Loom uses the extraction prompt P_{extract} (Appendix Table 15) to produce its structured representation $B = \{M,\ \text{topic},\ \text{events},\ \text{keywords}\}$. The event set $\text{events}(B) = \{e_1, e_2, \dots\}$ is extracted from the messages in B and captures concrete actions, factual developments, or plans within the box's topic, such as marathon training, recurring knee pain, or reducing training intensity in the running example. Since our memory design centers on topic continuity, extracting events provides a natural, fine-grained representation of each topic, while keywords supply supplementary descriptive details. The extracted event set $E(B)$ becomes the input to the **Trace Weaver** stage (§3.2). The extracted fields index the box and connect it to traces; they do not replace its original messages. Retrieving a box therefore also exposes details, such as profile facts, that may not appear in the extracted events.

3.2 Trace Weaver: Reinforcing the Organization Layer

Trace Weaver reinforces the organization layer by connecting events across sealed boxes. Association is built from each box's event set, and retrieval later accesses traces through events in the selected boxes. This gives the local organization cross-session reach while retaining boxes as the initial evidence-access units.

A trace associates events through a recurring activity, narrative, or theme. After Topic Loom seals B_{new}, let $E^{(new)} = \{e_1, \dots, e_p\}$ be its extracted events, $\mathcal{T} = \{T_1, \dots, T_q\}$ the existing traces, and $E^{(T_i)}$ the events in trace T_i.

Trace Initialization (if $\mathcal{T} = \varnothing$). If there are no existing traces, we pass $E^{(new)}$ to an LLM with the initialization prompt P_{init} (Appendix Table 17), clustering the events into one or more new traces:

$$\mathcal{T} \leftarrow \mathcal{T} \cup \text{LLM}\big(E^{(new)} \parallel P_{\text{init}}\big).$$

This establishes initial macro-topic traces for subsequent linking.

Event-to-Trace Voting. For each event $e_k \in E^{(new)}$, we first identify the trace containing the most semantically similar stored event:

$$T^*(e_k) = \arg\max_{T_i \in \mathcal{T}} \Big[\max_{e' \in E^{(T_i)}} \text{sim}(e_k, e') \Big],$$

where $\text{sim}(\cdot, \cdot)$ is cosine similarity in embedding space. An event contributes its nearest trace only when the nearest-event cosine similarity is at least $\tau = 0.5$. The union of retained votes gives the candidate set $\mathcal{C}(B_{new})$ for LLM verification.

LLM Batch Verification. For each candidate trace $T_i \in \mathcal{C}(B_{new})$, we pass both (a) the trace's existing events $E^{(T_i)}$ and (b) the full set of events in the current box $E^{(new)}$ to the LLM with the verification prompt P_{verify} (Appendix Table 16). The LLM jointly considers topic context and event semantics to decide which events from $E^{(new)}$ should be appended to T_i. Thus, although each event selects only one trace during candidate generation, verification compares every new event with every trace in the candidate set; an event may be accepted into multiple traces when it fits multiple macro-topic continuities. This batch decision process allows cross-event reasoning within the same box, capturing cases where related events reinforce each other's topical fit.

Secondary Trace Initialization. Events from $E^{(new)}$ not accepted into any existing traces form E_{unlinked}. If $E_{\text{unlinked}} \neq \varnothing$, they are re-passed to P_{init} to form new traces.

Traces may overlap: an event can belong to multiple traces, and different events from one box can join different traces. This lets a local episode participate in several recurring activities or themes. Events provide the interface for cross-box association, while the box remains the unit that preserves the local original dialogue.

3.3 Retrieval

Retrieval is anchored in the organization layer. Given a query q, we rank sealed boxes using an embedding of the concatenated original text and extracted topic, events, and keywords, denoted $R(B)$:

$$\mathcal{B}_q = \text{TopK}^{k_b}_{B \in \mathcal{B}} \ \text{sim}\big(q, R(B)\big).$$

The answering model receives the original text of every selected box. Indexing identifies the relevant organized unit; reading the unit supplies its surrounding exchanges, including messages with weak direct similarity to the query.

For trace retrieval, the selected boxes induce candidate events $\mathcal{E}_q$. The top-k_e query-relevant events then select their associated traces:

$$\begin{aligned}
\mathcal{E}_q &= \bigcup_{B \in \mathcal{B}_q} E(B), \\
\mathcal{E}_q^{\text{top}} &= \text{TopK}^{k_e}_{e \in \mathcal{E}_q} \text{sim}(q, e), \\
\mathcal{T}_q &= \{T_i \in \mathcal{T} \mid E^{(T_i)} \cap \mathcal{E}_q^{\text{top}} \neq \varnothing\}.
\end{aligned}$$

Thus, trace retrieval is a structured expansion of the box-level results: selected boxes determine the candidate events, and the most query-relevant events determine the long-range traces used as additional evidence. In Trace mode, the QA context consists of the retrieved box contents plus all events contained in the selected traces; raw dialogue from non-retrieved boxes is not added. Similarity-based indexing and event association thus access the structures established by the organization layer: boxes supply local context, and their events provide routes to related evidence across sessions.

4 EXPERIMENT

4.1 DATASET AND EVALUATION

We evaluate on *LoCoMo* (Maharana et al., 2024), comprising long conversations spanning up to 35 sessions. We use 1,540 questions in four categories: single-hop, multi-hop, temporal, and open-domain QA. Adversarial questions are excluded because the generation evaluation does not implement a separate abstention protocol.

We compare **Membox** with six competitive baselines: **LoCoMo** (Maharana et al., 2024), **ReadAgent** (Lee et al., 2024), **MemoryBank** (Zhong et al., 2024), **MemGPT** (Packer et al., 2023), **A-MEM** (Xu et al., 2025), and **Mem0** (Chhikara et al., 2025). We report **F1** for precision–recall balance in answer generation and **BLEU-1** for lexical overlap with references. We additionally assess semantic answer correctness with the GPT-4o-mini judging protocol released with LightMem (Fang et al., 2026).

We additionally evaluate on the Friends subset of DialSim; Appendix E reports the setup, results, and published baseline comparisons.

4.2 IMPLEMENTATION DETAILS

We utilize **text-embedding-3-small** for text embedding and OpenAI's **GPT-4o** and **GPT-4o-mini** as the backbone LLMs in the main experiments.

We locally evaluate A-MEM (Xu et al., 2025) and Mem0 (Chhikara et al., 2025) at retrieval depths $k \in \{5, 10, 20, 30\}$ and report their best observed scores; other baseline rows are previously reported results. The main Membox comparison uses box top-$k = 10$ with two modes: *Membox-Box* returns original Topic Loom box text, and *Membox-Trace* adds associated trace events using event top-$k = 2$. The original plots label the box-only mode "Compact"; no compression of the retrieved box text is performed. Appendix Tables 6 and 7 report retrieval-depth and trace-expansion sweeps. Each system constructs memory with its own prompts and uses the same backbone for QA within each comparison.

4.3 EMPIRICAL RESULTS

Table 1 evaluates access to the organized memory. Membox-Box retrieves Topic Loom boxes without trace expansion and outperforms the tested local baselines in F1 across the four categories for both GPT backbones. On GPT-4o-mini, its overall F1 is 53.95, compared with 40.95 for the best tested Mem0/A-MEM retrieval-depth setting (Appendix Table 6).

Membox-Trace uses Trace Weaver to extend access from these boxes to cross-session events. It further improves F1 in all four categories for both backbones, with the largest category gain on open-domain QA. BLEU-1 generally follows the same trend, with small temporal and single-hop tradeoffs for GPT-4o.

4.4 RETRIEVAL-DEPTH ANALYSIS

Table 2 summarizes the GPT-4o-mini box-only sweep; Appendix Table 6 gives all settings. F1 rises from 39.88 at top-1 to 49.41 at top-3 and 53.95 at top-10. At top-5, Membox-Box obtains 51.72 F1 with 1,538 context tokens, versus 40.95 with 1,955 tokens for the best tested Mem0 setting. Thus, the observed advantage does not require more answer context, although these are different retrieval configurations rather than a matched-budget comparison.

4.5 MEMORY CONSTRUCTION AND LINKING COSTS

Storage and construction. Appendix Table 8 reports comparable memory sizes for Membox and Mem0, both smaller than A-MEM. Membox averages 6.59 utterances per box with GPT-4o-mini and 4.88 with GPT-4o; storage uses space-delimited word counts, distinct from LLM tokens. Online continuity checks are followed by box-level extraction and writing. Construction consumes 236.18 and 254.57 LLM tokens per utterance for the two backbones, respectively (Appendix Table 9). Trace

Table 1: Main LoCoMo results. Box retrieves box content; Trace adds linked events. Both use content top-$k = 10$, with event top-$k = 2$ for Trace. * marks local implementations with retrieval-depth sweeps over $k \in \{5, 10, 20, 30\}$; other baseline rows are reported results. Bold marks the best score in each backbone/category/metric.

Model	Method	Category							
		Multi-Hop		Temporal		Open Domain		Single Hop	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
GPT-4o-mini	LoCoMo	25.02	19.75	18.41	14.77	12.04	11.16	40.36	29.05
GPT-4o-mini	ReadAgent	9.15	6.48	12.60	8.87	5.31	5.12	9.67	7.66
GPT-4o-mini	MemoryBank	5.00	4.77	9.68	6.99	5.56	5.94	6.61	5.16
GPT-4o-mini	MemGPT	26.65	17.72	25.52	19.44	9.15	7.44	41.04	34.34
GPT-4o-mini	A-MEM	27.02	20.09	45.85	36.67	12.14	12.00	44.65	37.06
GPT-4o-mini	A-MEM*	27.08	20.46	29.14	24.08	16.60	13.80	40.70	32.63
GPT-4o-mini	Mem0	38.72	27.13	48.93	40.51	28.64	21.58	47.65	38.72
GPT-4o-mini	Mem0*	36.83	26.50	34.52	26.38	22.57	16.54	46.89	37.63
GPT-4o-mini	**Membox-Box**	39.88	26.39	58.03	45.17	27.96	20.15	60.09	47.45
GPT-4o-mini	**Membox-Trace**	**41.19**	**27.49**	**59.63**	**46.52**	**30.36**	**22.52**	**61.18**	**48.99**
GPT-4o	LoCoMo	28.00	18.47	9.09	5.78	16.47	14.80	61.56	**54.19**
GPT-4o	ReadAgent	14.61	9.95	4.16	3.19	8.84	8.37	12.46	10.29
GPT-4o	MemoryBank	6.49	4.69	2.47	2.43	6.43	5.30	8.26	7.10
GPT-4o	MemGPT	30.36	22.83	17.29	13.18	12.24	11.87	60.18	53.35
GPT-4o	A-MEM	32.86	23.76	39.41	31.23	17.10	15.84	48.43	42.97
GPT-4o	Mem0*	42.57	30.92	44.55	32.60	23.04	17.62	48.49	37.00
GPT-4o	A-MEM*	31.66	23.34	41.11	34.72	17.45	15.58	47.04	41.02
GPT-4o	**Membox-Box**	48.35	35.10	65.06	**54.81**	30.61	22.58	61.69	49.36
GPT-4o	**Membox-Trace**	**50.48**	**38.17**	**66.61**	54.15	**38.77**	**28.19**	**62.56**	48.95

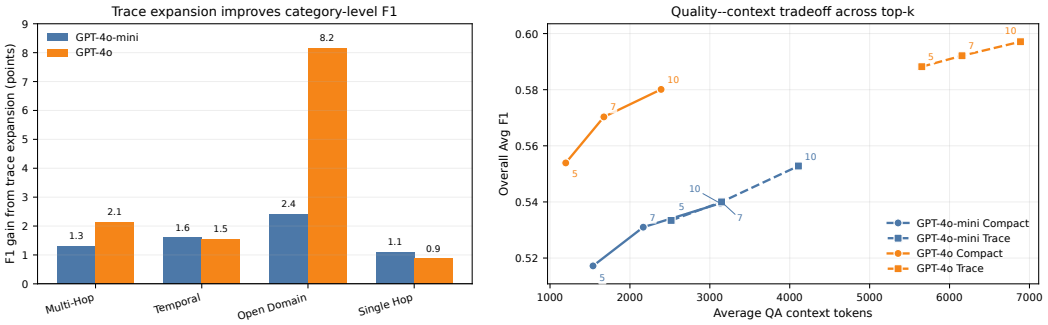


Figure 3: Effect of Trace Weaver at inference time. Left: F1 gains from trace expansion at box top-$k = 10$, event top-$k = 2$. Right: quality–context tradeoff across box top-$k \in \{5, 7, 10\}$, with event top-$k = 2$. The retained plot label "Compact" denotes Membox-Box.

linking costs about twice as much per box as construction, but their combined per-utterance usage remains below the reported local Mem0 and A-MEM costs (Appendix Table 10).

Inference-time tradeoff. Figure 3 and Table 3 show the cost of trace expansion. With GPT-4o-mini, F1 rises from 53.95 to 55.28 as mean QA context grows from 3130.72 to 4108.85 tokens; with GPT-4o, F1 rises from 58.01 to 59.71 as context grows from 2390.16 to 6887.84 tokens. Box-only retrieval offers lower context cost, while trace expansion improves answer quality with additional event evidence. Appendix Table 7 reports the full sweep.

Trace-expansion sensitivity. At box top-$k = 10$, increasing event top-k from 1 to 2 raises GPT-4o-mini F1 from 54.63 to 55.28, with mean context increasing from 3858.10 to 4108.85 tokens.

Table 2: Selected retrieval-depth results on GPT-4o-mini (1,540 questions). Scores use a 0–100 scale; context is mean QA tokens. Baselines are local implementations at their best overall F1 among tested depths. Full sweeps appear in Appendix Table 6.

Method	top-k	F1	BLEU-1	QA context
Mem0	30	40.95	31.93	1955.00
A-MEM	30	34.41	34.88	7246.66
Membox-Box	1	39.88	31.13	310.69
Membox-Box	3	49.41	38.18	917.03
Membox-Box	5	51.72	39.70	1538.10
Membox-Box	10	53.95	41.42	3130.72

Table 3: Overall LoCoMo performance at content top-$k = 10$; Trace uses event top-$k = 2$. Scores are on a 0–100 scale. Judge ACC uses GPT-4o-mini on 1,540 questions for each reported entry. Dashes denote unavailable complete-set judge or context statistics.

Backbone	Mode	F1	BLEU-1	Judge ACC	QA context
GPT-4o-mini	Box	53.95	41.42	–	3130.72
	Trace	55.28	42.89	77.53	4108.85
GPT-4o	Box	58.01	46.21	78.83	2390.16
	Trace	59.71	46.76	83.44	6887.84
DeepSeek-V3	Box	58.70	48.65	78.12	–

Expanding through all candidate events yields 55.60 F1 but uses 6527.00 tokens. GPT-4o shows a similar diminishing return beyond event top-$k = 2$: F1 rises from 59.71 to 59.86 with all events. These results support selective event expansion as the default, while exposing the additional context cost of broader access.

4.6 Semantic Evaluation and Model Extension

The semantic judge evaluates each answer as correct or wrong using the LightMem protocol. GPT-4o Trace improves judge accuracy from 78.83 to 83.44 over Box (Table 3), complementing the lexical-score gain. The category breakdown in Appendix Table 12 shows where this gain occurs: GPT-4o judge accuracy increases from 70.21 to 79.79 on multi-hop questions and from 46.88 to 58.33 on open-domain questions, compared with smaller gains on temporal (80.69 to 82.55) and single-hop questions (84.66 to 87.87). Thus, trace expansion improves semantic correctness across all four categories in this run, with larger gains on multi-hop and open-domain questions.

To examine the framework beyond the GPT-4o family, we also evaluate DeepSeek-V3 Box while keeping the embedding model fixed. It reaches 58.70 F1, 48.65 BLEU-1, and 78.12 judge ACC. This run uses a concise-answer prompt, removing explicit requests for step-by-step explanations; details of the prompt adjustment and answer-format sensitivity appear in Appendix C. The result extends the evidence for box-based memory access to another model family.

4.7 Multi-Party Dialogue Evaluation

On DialSim Friends, we evaluate multiple-choice dialogue understanding using GPT-4o-mini and the LoCoMo construction prompts. Trace raises accuracy from 52.78 to 55.33 with additional context (Table 4); mean QA context grows from 826.53 to 1872.54 tokens. Appendix E provides evaluation details and published reference baselines. These external results provide context rather than a controlled ranking, since a matched evaluation protocol has not been established.

4.8 Window-Size Sensitivity

Increasing the continuity window from 2 to 4 or 6 yields small F1 gains of 0.34 and 0.39 points, while construction tokens increase from 1.39M to 1.62M and 1.77M. This quality–cost pattern supports

Table 4: DialSim Friends in our evaluation, box top-$k = 10$. QA context is mean tokens.

Mode	Accuracy (%)	QA context
Box	52.78	826.53
Trace	55.33	1872.54

Table 5: Window sensitivity with GPT-4o-mini Box, content top-$k = 10$. Construction tokens exclude trace linking.

Window	F1	BLEU-1	Boxes	Build tokens	QA context
2	53.95	41.42	892	1.39M	3130.72
4	54.29	41.62	946	1.62M	2933.44
6	54.34	41.60	972	1.77M	2882.89

window 2 as the default. A local topic break can start a new box; Trace Weaver retains access to related events across these breaks.

4.9 Case Studies: Evidence Recovery and Failure Modes

We inspect recorded GPT-4o outputs at box top-$k = 10$ and Trace event top-$k = 2$. These cases illustrate behavior; they do not estimate error frequencies.

Preserving a local exchange. In S0B19, Melanie describes pottery as a way to express herself; Caroline replies, "I'm getting creative too, just learning the piano." Topic Loom groups the exchange under "creative activities and personal expression," preserving its conversational connection across different activities.

Retaining facts beyond extracted events. For "What are Melanie's pets' names?" (sample 0, question 52), both modes return Luna, Oliver, and Bailey (F1 1). S0B32 contains "Luna and Oliver!" and S0B56 contains "We got another cat named Bailey too." The extracted events omit the complete profile; S0B32's event instead describes new shoes. Original box text nevertheless keeps the names accessible, illustrating why extracted indexing features supplement rather than replace the underlying dialogue.

Recovering a missed event. For "What was Sam doing on December 4, 2023?" (sample 8, question 68), Box misses S8B72 and answers "No relevant memory available" (F1 0). A health-related trace reaches the missed meeting event, and Trace answers "Attended a Weight Watchers meeting" (F1 0.80). The underlying evidence says "yesterday," so interpreting the date also requires session-time context.

When expansion distracts. For "Which pet did Jolene adopt first—Susie or Seraphim?" (sample 7, question 54), Box correctly answers Susie (F1 1). A broad pet-history trace adds competing Seraphim mentions, and Trace answers Seraphim (F1 0). Additional evidence can therefore distract even when the retrieved boxes already suffice. Appendix D provides excerpts and identifiers, including an over-merging case: S0B22 combines a career update with a family museum visit under one broad topic, illustrating the loss of specificity from overly inclusive boundaries.

5 Conclusions

Membox couples online memory organization with evidence access. Topic Loom preserves local dialogue in original-text boxes; Trace Weaver connects their events into overlapping traces accessed from the retrieved boxes. On LoCoMo, box retrieval outperforms the tested local baselines, while trace expansion further improves answer quality with additional context. Semantic evaluation and retrieval cases complement these results, and DialSim extends the evaluation to multi-party dialogue.

AI USE STATEMENT

AI tools were used to assist with language editing, literature comparison, and preparation of this revision. The authors are responsible for verifying the manuscript, citations, experimental records, and final claims.

REPRODUCIBILITY STATEMENT

The method specifies online box formation, trace construction, and hierarchical retrieval. The appendix provides construction prompts, sensitivity tables, supplementary evaluation conditions, and case identifiers.

REFERENCES

Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. SECTOR: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184, 2019. doi: 10.1162/tacl_a_00261. URL https://aclanthology.org/Q19-1011/.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

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.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dyJ0GWpjJB.

Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 562–569, Sapporo, Japan, 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075167. URL https://aclanthology.org/P03-1071/.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023. URL https://arxiv.org/abs/2312.10997.

Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986. URL https://aclanthology.org/J86-3001/.

Marti A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997. URL https://aclanthology.org/J97-1003/.

Junfeng Jiang, Chengzhang Dong, Sadao Kurohashi, and Akiko Aizawa. SuperDialseg: A large-scale dataset for supervised dialogue segmentation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4086–4101, Singapore, 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.249. URL https://aclanthology.org/2023.emnlp-main.249/.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023b. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Jiho Kim, Woosog Chay, Hyeonji Hwang, Daeun Kyung, Hyunseung Chung, Eunbyeol Cho, Yeonsu Kwon, Yohan Jo, and Edward Choi. DialSim: A dialogue simulator for evaluating long-term multi-party dialogue understanding of conversational agents. *arXiv preprint arXiv:2406.13144*, 2024. URL https://arxiv.org/abs/2406.13144v6. Baseline results from version 6 (2025).

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 469–473, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2075. URL https://aclanthology.org/N18-2075/.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*, 2024. URL https://arxiv.org/abs/2402.09727.

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, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Jasper Linders and Jakub M. Tomczak. Knowledge graph-extended retrieval augmented generation for question answering. *Applied Intelligence*, 55(17), 2025. doi: 10.1007/s10489-025-06885-5. Article 1102.

Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. Text segmentation by cross segment attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4707–4716, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.380. URL https://aclanthology.org/2020.emnlp-main.380/.

Kexin Ma, Bojun Li, Yuhua Tang, Liting Sun, and Ruochun Jin. CAST: Character-and-scene episodic memory for agents. *arXiv preprint arXiv:2602.06051*, 2026. URL https://arxiv.org/abs/2602.06051.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. *arXiv preprint arXiv:2402.17753*, 2024. URL https://arxiv.org/abs/2402.17753v1.

Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. RET-LLM: Towards a general read-write memory for large language models. *arXiv preprint arXiv:2305.14322*, 2023. URL https://arxiv.org/abs/2305.14322.

Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seung-won Hwang, Dongha Lee, and Jinyoung Yeo. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8631–8661, 2025. doi: 10.18653/v1/2025.naacl-long.435. URL https://aclanthology.org/2025.naacl-long.435/.

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.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agent. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.13530.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *Proceedings of the 2024 International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2307.07697v6.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. SCM: Enhancing large language model with self-controlled memory framework. *arXiv preprint arXiv:2304.13343*, 2023. URL https://arxiv.org/abs/2304.13343.

Kaixiang Wang, Yidan Lin, Jiong Lou, Zhaojiacheng Zhou, Bunyod Suvonov, and Jie Li. E-mem: Multi-agent based episodic context reconstruction for LLM agent memory. *arXiv preprint arXiv:2601.21714*, 2026. URL https://arxiv.org/abs/2601.21714v5.

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. Structmem: Structured memory for long-horizon behavior in llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146, 2026. doi: 10.18653/v1/2026.acl-short.12. URL https://aclanthology.org/2026.acl-short.12/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. *arXiv preprint arXiv:2502.12110*, 2025. URL https://arxiv.org/abs/2502.12110.

Yanyu Yao, Shangze Li, Zhi Zheng, Hui Zheng, Qi Liu, Tong Xu, and Enhong Chen. AtomMem: Building simple and effective memory system for LLM agents via atomic facts. *arXiv preprint arXiv:2606.19847*, 2026. URL https://arxiv.org/abs/2606.19847.

Jeffrey M. Zacks, Nicole K. Speer, Khena M. Swallow, Todd S. Braver, and Jeremy R. Reynolds. Event perception: A mind/brain perspective. *Psychological Bulletin*, 133(2):273–293, 2007. doi: 10.1037/0033-2909.133.2.273.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

Rolf A. Zwaan, Mark C. Langston, and Arthur C. Graesser. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological Science*, 6(5):292–297, 1995. doi: 10.1111/j.1467-9280.1995.tb00513.x.

A Retrieval-Depth and Trace-Expansion Details

The sensitivity tables below report F1 and BLEU-1 on a 0–1 scale; the main overall table uses a 0–100 scale.

A.1 Trace-Expansion Sensitivity

Table 7 reports the full sensitivity sweep for trace-expanded retrieval. Across both GPT-4o and GPT-4o-mini, adding Trace Weaver evidence consistently improves over box-only retrieval at the same content top-k, confirming that macro-topic traces provide useful long-range evidence beyond the retrieved local episodes. Increasing the event top-k generally raises F1 but also increases context length, so the main experiments use event top-$k = 2$ as a balanced setting: it captures most of the trace benefit without adding all linked events to the QA context.

A.2 Category-Level Retrieval-Depth Sensitivity

Figure 4 provides a category-level view of retrieval-depth sensitivity for Membox-Box with GPT-4o-mini. The plots vary the number of retrieved Topic Loom boxes while keeping the retrieval mode fixed to box-only retrieval, so they isolate how much local topic-continuous evidence is needed for each question type.

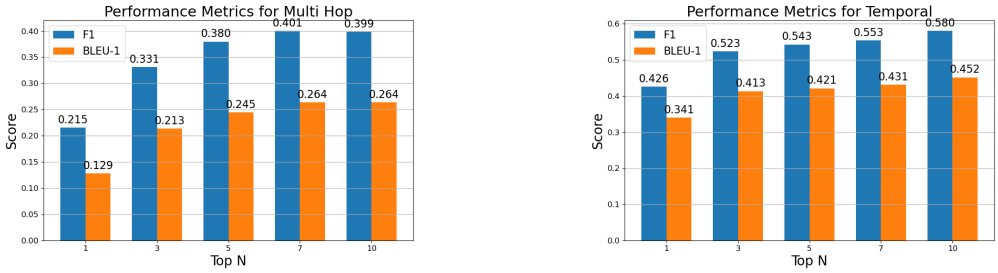
Table 6: Overall retrieval-depth comparison on GPT-4o-mini. Membox rows use *Membox-Box*. All rows are evaluated on 1540 QA instances.

Method	top-k	Avg F1	Avg BLEU-1	Avg ctx tok
Mem0	5	0.3836	0.2970	331.14
Mem0	10	0.3986	0.3102	656.89
Mem0	20	0.4035	0.3155	1306.11
Mem0	30	0.4095	0.3193	1955.00
A-MEM	5	0.3063	0.2524	1238.77
A-MEM	10	0.3277	0.2926	2449.88
A-MEM	20	0.3365	0.3273	4873.67
A-MEM	30	0.3441	0.3488	7246.66
Membox-Box	1	0.3988	0.3113	310.69
Membox-Box	3	0.4941	0.3818	917.03
Membox-Box	5	0.5172	0.3970	1538.10
Membox-Box	7	0.5310	0.4070	2166.88
Membox-Box	10	0.5395	0.4142	3130.72

Table 7: Full overall sensitivity of Box and Trace inference modes. Trace rows use the same content retrieval as Box and vary the event top-k. All rows are evaluated on 1540 QA instances.

Model	Mode	Content top-k	Event top-k	Avg F1	Avg BLEU-1	Avg ctx tok
GPT-4o	Box	5	–	0.5539	0.4394	1196.79
GPT-4o	Box	7	–	0.5703	0.4505	1674.21
GPT-4o	Box	10	–	0.5801	0.4621	2390.16
GPT-4o	Trace	5	1	0.5826	0.4616	5362.73
GPT-4o	Trace	7	1	0.5881	0.4617	5827.23
GPT-4o	Trace	10	1	0.5908	0.4643	6555.07
GPT-4o	Trace	5	2	0.5882	0.4664	5652.71
GPT-4o	Trace	7	2	0.5921	0.4649	6155.77
GPT-4o	Trace	10	2	0.5971	0.4676	6887.84
GPT-4o	Trace	5	all	0.5960	0.4711	5928.89
GPT-4o	Trace	7	all	0.5919	0.4646	6459.29
GPT-4o	Trace	10	all	0.5986	0.4697	7225.96
GPT-4o-mini	Box	5	–	0.5172	0.3970	1538.10
GPT-4o-mini	Box	7	–	0.5310	0.4070	2166.88
GPT-4o-mini	Box	10	–	0.5395	0.4142	3130.72
GPT-4o-mini	Trace	5	1	0.5301	0.4164	2268.74
GPT-4o-mini	Trace	7	1	0.5369	0.4187	2900.56
GPT-4o-mini	Trace	10	1	0.5463	0.4253	3858.10
GPT-4o-mini	Trace	5	2	0.5334	0.4185	2515.48
GPT-4o-mini	Trace	7	2	0.5400	0.4217	3146.83
GPT-4o-mini	Trace	10	2	0.5528	0.4289	4108.85
GPT-4o-mini	Trace	5	all	0.5400	0.4193	4042.98
GPT-4o-mini	Trace	7	all	0.5447	0.4209	5065.46
GPT-4o-mini	Trace	10	all	0.5560	0.4295	6527.00

Temporal and single-hop questions benefit steadily from larger retrieval depth, suggesting that additional topic-continuous boxes provide useful supporting evidence for event order, duration, and factual lookup. Multi-hop questions improve sharply from top-1 to top-7 and then saturate, indicating that a small set of coherent episodes already captures most of the required evidence. Open-domain questions peak around top-5 and then slightly decline, which suggests that adding more dialogue context can introduce distractors when the answer also depends on external commonsense knowledge. Overall, the category-level trends support the main retrieval-depth analysis: retrieving a few topic-continuous boxes yields most of the gain, while larger top-k values mainly trade additional evidence for more context.



(a) Multi-hop

(b) Temporal

(c) Open-domain

(d) Single-hop

Figure 4: Category-level retrieval-depth sensitivity for Membox-Box with GPT-4o-mini on LoCoMo. Top-N denotes the number of retrieved Topic Loom boxes.

B STORAGE AND PREPROCESSING STATISTICS

The storage table counts space-delimited words. The two construction/linking tables instead record LLM processing tokens. They should not be interpreted as the same tokenization measure.

Table 8: Memory base statistics. Utterances: total number of utterances; Word Ratio: (constructed memory words) / (original dialogue words); MB#: Membox count; Utter/MB: utterances per Membox; Words/MB: space-delimited words per Membox. Words are segmented by spaces; these are not LLM processing tokens.

Method	Utterances	Word Ratio	MB#	Utter/MB	Words/MB
Mem0 w/ GPT-4o-mini	5882	1.19	-	-	-
Mem0 w/ GPT-4o	5882	1.20	-	-	-
A-MEM w/ GPT-4o-mini	5882	1.72	-	-	-
A-MEM w/ GPT-4o	5882	1.73	-	-	-
Membox w/ GPT-4o-mini	5882	1.24	892	6.59	342.98
Membox w/ GPT-4o	5882	1.19	1206	4.88	252.63

Table 9: LLM call statistics during memory base construction. MB#: Membox count; Tok/MB: LLM tokens consumed per Membox; Tok/Ut: LLM tokens consumed per utterance.

Method	MB#	Tok/MB	Tok/Ut
Mem0 w/ GPT-4o-mini	-	-	2115.85
Mem0 w/ GPT-4o	-	-	1923.17
A-MEM w/ GPT-4o-mini	-	-	1755.57
A-MEM w/ GPT-4o	-	-	1526.39
Membox w/ GPT-4o-mini	892	1557.44	236.18
Membox w/ GPT-4o	1206	1241.61	254.57

Table 10: LLM usage statistics for Membox linking. MB#: Membox count; Calls/MB: LLM calls per Membox; Tok/MB: tokens consumed per Membox.

Model	MB#	Calls/MB	Tok/MB
GPT-4o-mini	892	2.30	3133.56
GPT-4o	1206	0.88	2716.89

C SUPPLEMENTARY DEEPSEEK-V3 AND JUDGE RESULTS

The DeepSeek-V3 experiment uses Box with content top-$k = 10$, preserving the embedding model. The implementation parses fenced JSON outputs and uses re-extracted box features. The original answer prompt mixed a short-answer request with explicit reasoning-output instructions; removing the latter improved lexical scores. Table 11 separates these stages. No DeepSeek trace-expansion result is reported. The final concise-answer run obtains 46.45/36.97 on multi-hop, 63.03/47.08 on temporal, 33.05/25.83 on open-domain, and 64.08/55.77 on single-hop QA (F1/BLEU-1). The main text reports the concise-answer setting; these conditions document how answer formatting affects lexical metrics.

Table 11: DeepSeek-V3 diagnostic stages (0–100). These are alternative processing/prompt conditions, not independent repeated trials.

Condition	F1	BLEU-1
Re-extracted features, original answer prompt	35.05	22.81
Re-extracted features, final-answer normalization	53.47	36.80
Re-extracted features, concise answer prompt	58.70	48.65

The supplementary judge is GPT-4o-mini with the binary `CORRECT`/`WRONG` protocol from the LightMem release. Table 12 provides category scores for complete runs. The incomplete mini Box run is documented separately: 1,153 regenerated rows, 76.24 overall ACC. It is not used to estimate a complete-set Box-to-Trace gain.

Table 12: Complete-set judge accuracy (%, 1,540 questions per row). MH: multi-hop; T: temporal; OD: open-domain; SH: single-hop.

Backbone	Mode	Overall	MH	T	OD	SH
GPT-4o-mini	Trace	77.53	68.79	74.45	50.00	84.78
GPT-4o	Box	78.83	70.21	80.69	46.88	84.66
GPT-4o	Trace	83.44	79.79	82.55	58.33	87.87
DeepSeek-V3	Box	78.12	74.82	74.14	50.00	83.95

D QUALITATIVE CASES

All cases below are from the recorded GPT-4o run, using content top-$k = 10$ and, for Trace, event top-$k = 2$. They were selected to illustrate behavior, not sampled to estimate error frequencies. Box and question identifiers are retained to facilitate checking against the experiment outputs.

D.1 TOPIC LOOM: A COHERENT LOCAL EPISODE

Sample 0, box S0B19, session 5, messages 4–11. The learned topic is *Creative activities and personal expression*. Extracted events include Melanie signing up for pottery and making a bowl, and Caroline learning piano.

> Melanie: I just signed up for a pottery class yesterday. It's like therapy for me, letting me express myself and get creative.
> Caroline: I'm getting creative too, just learning the piano.
> Melanie: Pottery is a huge part of my life, not just a hobby—it helps me express my emotions.

The box preserves a coherent local exchange across different activities.

D.2 TOPIC LOOM: OVER-MERGING

Sample 0, box S0B22, session 6, messages 3–4. The learned topic is *Personal updates and interests*.

> Caroline: I've been looking into counseling or mental health work more. I'm passionate about helping people and making a positive impact.
> Melanie: Yesterday I took the kids to the museum—it was so cool spending time with them and seeing their eyes light up!

These neighboring updates concern different factual threads. Storing them together can reduce specificity. Such broad grouping illustrates the tradeoff between local context and topic specificity.

D.3 TRACE WEAVER: RECOVERING A MISSED EVENT

Sample 8, question 68: *What was Sam doing on December 4, 2023?* Gold answer: *Attending a Weight Watchers meeting*. Box returns *No relevant memory available* (F1 0); Trace returns *Attended a Weight Watchers meeting* (F1 0.80).

The target box, S8B72, is absent from Box's top-10. Its evidence includes: *I attended a Weight Watchers meeting yesterday. Learned some great tips.* The recorded health-related trace connects diet changes, Weight Watchers coaching, a health scare, gym activity, the meeting, yoga, and running across boxes B59, B60, B64, B68, B72, B73, and B83. Trace expansion makes the missed event available to the answering model. Interpreting the relative date also requires its session-time context.

D.4 TRACE WEAVER: COMPETING PET HISTORIES

Sample 7, question 54: *Which pet did Jolene adopt first—Susie or Seraphim?* Gold: *Susie*. Box answers Susie (F1 1); Trace answers Seraphim (F1 0).

Box already retrieves evidence that Susie has been with Jolene for two years. Added trace context includes Seraphim leaving her cage and visiting the park. The broad pet-history trace supplies competing mentions despite sufficient local evidence. The example illustrates how broad traces can add distracting context to an otherwise sufficient local retrieval.

D.5 PROFILE FACTS RETAINED IN BOX CONTENTS

Sample 0, question 52: *What are Melanie's pets' names?* Both modes return Luna, Oliver, and Bailey (F1 1). S0B32 contains *Luna and Oliver!*, and S0B56 contains *We got another cat named Bailey too.*

The corresponding extracted events do not capture the full profile: S0B32's event instead describes new shoes. Retaining original box contents nevertheless keeps the names accessible. This illustrates how box content preserves profile facts alongside the event-trace representation.

E Supplementary Evaluation on DialSim

We additionally evaluate Membox on the Friends subset of DialSim (Kim et al., 2024), extending the evaluation to multi-party dialogue. We use GPT-4o-mini for question answering, retain the memory-construction prompts used for LoCoMo, and follow the DialSim multiple-choice answering format. Responses are evaluated by whether the selected option is correct; the scores below are reported as accuracy.

Table 13 places these results alongside the GPT-4o-mini baselines reported by Kim et al. (2024). Base LLM uses as much recent dialogue as the model's context window permits. Retrieval baselines index individual utterances, entire sessions, or session summaries, using either BM25 or OpenAI embeddings. The baseline values follow the original paper's evaluation, while the Membox rows come from our evaluation.

Table 13: DialSim Friends results with GPT-4o-mini. Baselines are published results from Kim et al. (2024); Membox results are from our evaluation described above, with box top-$k = 10$. QA context is the recorded mean token count; dashes denote unavailable comparable measurements for the published baselines.

Method	Accuracy (%)	QA context tokens
Published baselines		
Base LLM	48.11	–
BM25 / Utterance	35.46	–
BM25 / Session Entire	52.11	–
BM25 / Session Summary	44.15	–
OpenAI Embedding / Utterance	41.29	–
OpenAI Embedding / Session Entire	45.47	–
OpenAI Embedding / Session Summary	42.78	–
Our evaluation		
Membox-Box	52.78	826.53
Membox-Trace	55.33	1872.54

The published baselines provide reference values; a matched question set and evaluation protocol have not been established, so these numbers do not support a controlled ranking against Membox. Within our evaluation, trace expansion increases accuracy from 52.78 to 55.33, a gain of 2.55 percentage points, while mean QA context increases from 826.53 to 1872.54 tokens. This supplementary result extends the evaluation to multi-party dialogue and illustrates the answer-quality and context-cost tradeoff of expanding beyond the retrieved boxes.

F CONSTRUCTION PROMPT TEMPLATES

In this study, four types of prompts are employed, each serving a distinct function. PROMPT_MSG_CONTINUATION (Table 14) is used during memory construction to determine whether the current dialogue is continuous with the previous context, thereby deciding whether the active topic-continuous episode should be extended or sealed. PROMPT_DIALOG_EXTRACT (Table 15) extracts key information from the dialogue and converts it into a structured format stored in the memory module. PROMPT_TRACE_EVENT_FILTER (Table 16) and PROMPT_TRACE_INIT (Table 17) are used to construct event traces.

Table 14: PROMPT MSG CONTINUATION

PROMPT MSG CONTINUATION
Please determine whether the current message continues with the main topic of the previous messages. Only answer Yes/No/Partially Shifted.
previous messages: `ref`
current message: `curr`
Answer:

Table 15: PROMPT DIALOG EXTRACT

PROMPT DIALOG EXTRACT

Please analyze the relationships between the following entities in the given sentence.

Generate a structured analysis of the provided dialog by performing the following tasks:

1. **Identifying salient keywords**: Extract 3-8 most salient nouns, named entities, and key terminology that represent core concepts. Avoid common words (e.g., "good", "see") and prioritize specificity.

2. **Determining the core topic**: In one clear phrase, state the primary subject or objective of the discussion based on the actual content.

3. **Extracting explicit event and plan mentions**: Identify and list only the **events, factual developments, or specific future plans** that are **explicitly mentioned** in the dialog. Follow these strict rules:

3.1. **Focus on Verbatim or Near-Verbatim Content**: Each extracted item must be directly grounded in the dialog text. Do not infer, summarize, or combine information to create new "events."

3.2. **Distinguish Event Types**:
- **Past/Completed Events**: Actions or occurrences that are stated as having happened (e.g., "I went to...", "We completed the project").
- **Established Facts/Changes**: Concrete facts or changes presented as already true (e.g., "I am now the team lead", "The system is down").
- **Explicit Future Plans**: Specific plans for the future mentioned by the speakers (e.g., "We will meet on Friday", "I'm planning to visit Paris").

3.3. **Exclude Non-Events**: Do NOT include:
- General states of being (e.g., "I'm swamped", "I'm happy").
- Questions, greetings, or expressions of intent without a plan (e.g., "We should talk sometime").
- Vague aspirations or possibilities.

3.4. **Framing**: Phrase each extracted item as a concise, standalone clause that captures the core of what was mentioned.

Output Format: Provide the analysis as a valid JSON object with the following exact keys:

```
{
  "keywords": [
    "keyword1",
    "keyword2",
    ...
  ],
  "topic": "clear topic phrase",
  "explicit_mentions": [
    "A mentioned past event or established fact",
    "A mentioned specific future plan"
  ]
}
```

Content to analyze: {text}

Table 16: PROMPT_TRACE_EVENT_FILTER: instructions and output schema.

PROMPT_TRACE_EVENT_FILTER

You are a narrative coherence analyzer for constructing and maintaining event memory chains. Your task is to filter events from a new event list (Event List B) that are directly related to an existing event chain (Event Chain A).

Core Task:
Event Chain A represents an existing sequence of events (could be one or multiple events). Event List B is a set of newly observed events. Analyze each event in B to determine whether it should:
1. Serve as a **direct continuation** of Event Chain A (directly related to A's core narrative)
2. Be considered **unrelated** to Event Chain A (independent or belonging to a different event stream)

Analysis Principles:
- Identify the **core theme/activity** from Event Chain A's overall narrative
- Assess narrative continuity: Does the event from B advance, develop, or resolve A's core activity?
- Consider temporal/causal logic: Does the event naturally follow A's chain in time or logic?

Decision Criteria:
An event from B is **related** to Event Chain A if it:
1. Continues the **same core activity** as A's chain (not just similar topic)
2. Provides **progress, outcome, solution, or direct consequence** to A's chain
3. Is a **logical/temporal successor** to A's chain

An event from B is **unrelated** to Event Chain A if it:
1. Initiates a **new, distinct activity** (even if topic is similar)
2. Is a **parallel but independent** event to A's core activity
3. Concerns a **different aspect** unrelated to A's main thread
4. Is a **generic response** without specific progression

Output Format:
Strictly use this JSON format:

```
{
    "chain_summary": "Brief summary of Event Chain A's core theme (1-2 sentences)",
    "related_events": ["Exact text of related events from B"],
    "unrelated_events": ["Exact text of unrelated events from B"],
    "reasoning": {
        "related_reasons": ["Brief explanation for each related event"],
        "unrelated_reasons": ["Brief explanation for each unrelated event"]
    }
}
```

Table 16: PROMPT_TRACE_EVENT_FILTER (continued): examples and input fields.

PROMPT_TRACE_EVENT_FILTER (continued): examples

Example 1:
Event Chain A: ["I'm planning a weekend hike", "I checked the weather forecast", "I bought hiking shoes"]
Event List B: ["I mapped out the hiking route", "I replied to work emails", "I contacted hiking partners", "Went to see a movie in the evening"]
Output:

```
{
    "chain_summary": "Preparations for a weekend hiking trip",
    "related_events": ["I mapped out the hiking route", "I contacted hiking partners"],
    "unrelated_events": ["I replied to work emails", "Went to see a movie in the
    ↪  evening"],
    "reasoning": {
        "related_reasons": [
            "Mapping the route is a concrete step in hike preparation",
            "Contacting partners directly advances the hiking activity"
        ],
        "unrelated_reasons": [
            "Work emails concern a different domain (work vs. recreation)",
            "Movie watching is a separate leisure activity"
        ]
    }
}
```

Example 2:
Event Chain A: ["The project encountered technical difficulties", "The team met to discuss solutions"]
Event List B: ["I researched relevant documentation", "Decided to adopt a new framework", "Had pizza for lunch", "Client sent new requirements"]
Output:

```
{
    "chain_summary": "Addressing technical challenges in a project",
    "related_events": ["I researched relevant documentation", "Decided to adopt a new
    ↪  framework"],
    "unrelated_events": ["Had pizza for lunch", "Client sent new requirements"],
    "reasoning": {
        "related_reasons": [
            "Researching documentation directly addresses the technical problem",
            "Deciding on a new framework represents a solution to the technical
            ↪  challenge"
        ],
        "unrelated_reasons": [
            "Lunch is a routine activity unrelated to problem-solving",
            "New client requirements initiate a separate work thread"
        ]
    }
}
```

Now analyze:
Event Chain A: {content_a} (Note: This is an existing event chain)
Event List B: {content_b} (Note: This is a new event list)
Output your analysis.

Table 17: PROMPT TRACE INIT

PROMPT TRACE INIT

You are an event chain constructor for building coherent memory structures. Your task is to analyze a set of events and organize them into logical chains.

Task:
Given a set of events, identify the primary narrative thread and any associated events that form a coherent event chain.

Process:
1. Analyze all events to identify the most prominent theme or activity
2. Connect events that share temporal, causal, or thematic relationships
3. Form the most coherent sequence possible
4. Identify any events that don't fit into the main narrative thread

Output Format:

```
{
    "primary_chain": ["Events forming the most coherent narrative, in logical order"],
    "secondary_chains": [["Other potential chains, if any"]],
    "isolated_events": ["Events that don't fit into any chain"],
    "chain_summary": "Brief description of the primary chain's theme and context"
}
```

Examples:

Example 1:
Events: ["I woke up at 7 AM", "I checked my email", "I had breakfast", "Then I went for a run"]

Output:

```
{
    "primary_chain": ["I woke up at 7 AM", "I had breakfast", "Then I went for a run"],
    "secondary_chains": [],
    "isolated_events": ["I checked my email"],
    "chain_summary": "Morning routine including waking, eating, and exercise"
}
```

Example 2:
Events: ["Started a new project at work", "Researched design patterns", "Met with the client", "Created initial wireframes", "Had lunch with a colleague"]

Output:

```
{
    "primary_chain": ["Started a new project at work",
    "Researched design patterns", "Created initial wireframes"],
    "secondary_chains": [["Met with the client"]],
    "isolated_events": ["Had lunch with a colleague"],
    "chain_summary": "Work project initiation and initial design phase"
}
```

Now analyze:
Events: `{events}`
Output your analysis in JSON format.