FROM INTERFERENCE TO EVIDENCE: DUAL-STAGE MEMORY REACTIVATION FOR LONG-TERM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory systems enable LLM agents to deepen their understanding of users, continue tasks across sessions, and draw on accumulated experience in new situations. As interaction histories grow and evolve, maintaining this continuity requires both preserving past information and adapting its use to the current context. However, semantically relevant but outdated or event-mismatched memories may be misused as supporting evidence, a failure we term *Positive Misreactivation*. Inspired by human memory reactivation, we propose **DREAM** (**D**ual-stage **REA**ctivation Event **M**emory), which organizes history into a silent event substrate and controls its influence through dual-stage reactivation. For each query, a repulsive encoding gate converts potentially misleading memories into contrastive constraints, and an attractive decision gate selects valid support under these constraints. The resulting compact evidence state guides generation, with source details retrieved on demand. Across memory benchmarks, DREAM performs best among evaluated baselines; with Qwen3-30B, it improves by 6.53 F1 points on LoCoMo and 5.33 accuracy points on LongMemEval. With GPT-4o-mini, it uses 89.4% fewer answer-facing memory tokens than full-context prompting. Our work challenges the implicit equivalence between memory relevance and evidential support, establishing controlled reactivation as a more reliable paradigm for long-term conversational memory.

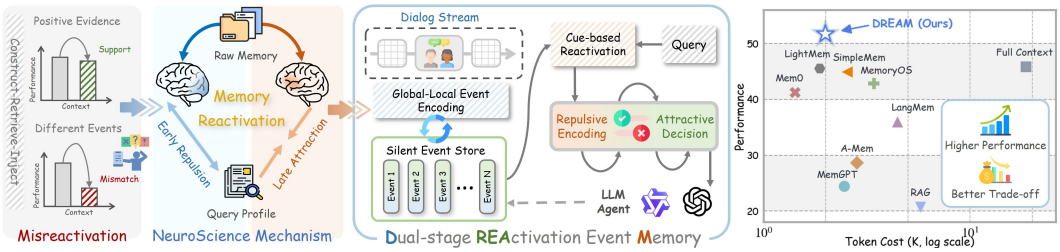

Figure 1: **Motivation:** To address positive misreactivation, we turn to neuroscience research: historical events are stored in different forms of memory in perceptual (occipital) and cognitive (frontal) cortices, and are reactivated at different stages and forms during current events processing. **Architecture:** Inspired by this, DREAM introduces a dual-stage memory reactivation mechanism to enable temporally consistent response generation in long-term dialogues. **Performance:** Our model consistently outperforms baselines across benchmarks.

1 INTRODUCTION

"The art of being wise is the art of knowing what to overlook." — William James

The evolution of agents from short-term interaction tools into persistent companions relies on long-term memory (Wang et al., 2024; Li et al., 2024; Guo et al., 2024). A useful conversational agent should accumulate experience, deepen its understanding of users through continued interaction, and carry unfinished tasks into new sessions (Huang et al., 2024; Hu et al., 2026a). As LLMs are increasingly used for personal assistants and long-term tasks, the background needed to understand a current request often resides in an evolving history spanning multiple sessions. The central goal of long-context, or even infinite-context, agents is therefore to let experience continually inform present judgment (Liu et al., 2025; Cao et al., 2025; Zhou et al., 2025).

Existing methods have progressively improved how such histories are stored. Early approaches use summarization, vector indexing, and fragment retrieval to externalize conversations beyond the

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.

context window (Lewis et al., 2020; Arslan et al., 2024; Fan et al., 2024). More recent systems introduce structured organization (Packer et al., 2023; Xu et al., 2026b; Wang & Chen, 2025), alongside event memory, state maintenance, and evidence preservation (Chhikara et al., 2025; Kang et al., 2025; Yan et al., 2025). These designs recognize that dialogue contains both reusable patterns and event-specific details whose meaning changes over time: some can be abstracted into stable profiles, whereas others require verification against original records. Agent memory has thus moved from expanding context capacity toward constructing manageable long-term experience.

However, when memory systems move from "storing history" to "using history," a deeper problem begins to emerge. As illustrated in Figure 1, many systems follow a ***Construct-Retrieve-Inject*** pipeline, treating retrieved memories as supporting context (Fang et al., 2026; Yu et al., 2026b; He et al., 2026; Zhang et al., 2026). Yet semantic relevance does not guarantee that a memory supports the current answer: preferences may change, plans may be canceled, and similar topics may concern different events (Zhou et al., 2026). We refer to this failure mode as ***Positive Misreactivation***, where a relevant but state-invalid or event-mismatched trace is mistakenly used as positive evidence. Unlike forgetting, where needed information is lost or cannot be recovered (Fernando et al., 2024), the information here is found, but its role is misjudged. Old records cannot simply be deleted, since they may remain necessary for historical questions (Chao et al., 2026). Meanwhile, passing retrieved records into the answer context without restructuring them can carry redundant descriptions and inapplicable details, consuming tokens without adding useful support. The challenge is therefore to control how recalled history influences generation while preserving its future usefulness.

Human memory mechanisms offer a coherent design clue. When experience exceeds immediate capacity, the brain can reorganize it into reusable structures (McClelland et al., 1995; Spens & Burgess, 2026; Ren et al., 2026), including hierarchical representations with global positions and local roles (Fan et al., 2025; Chunharas et al., 2024). These traces need not remain continuously active but can be awakened by subsequent task cues (Li et al., 2026; Yadav et al., 2022). Crucially, reactivation need not reinstate past content verbatim: historical information can exert repulsive effects during early encoding and attractive effects during later decision-making, with the latter modulated by task relevance (Luo et al., 2025; Zhang & Luo, 2023). History thus influences current judgment differently across processing stages. This motivates a principle for long-term conversational agents: *Memory should form a silent event substrate: the current query controls reactivation and determines whether memories support, constrain, or remain silent.*

Based on this principle, we propose **DREAM** (**D**ual-stage **REA**ctivation Event **M**emory), a framework for long-term conversational agents. DREAM first organizes history into reactivation-ready **Event memories**, retaining global-local structure, state and accessibility information, residual details, and source pointers. These fields allow future queries to distinguish event identity, applicability, and evidence provenance. During inference, structured query cues awaken candidate events from the silent store. Before passing these events to the answer model, DREAM determines their roles in generation: a **Repulsive encoding gate** identifies relevant memories that could mislead the answer and transforms them into contrastive constraints; an **Attractive decision gate** then selects valid, grounded support under those constraints. Supporting and contrastive memories form a compact, query-conditioned transformed evidence state that guides answer generation, with source pointers enabling readback when details are needed. DREAM thereby preserves historical records while controlling their roles in the current answer, reducing interference from obsolete or conflicting memories without passing retrieved events into the answer context in full.

Extensive experiments on traditional memory and popular long-context benchmarks demonstrate that DREAM consistently achieves SOTA performance, while using fewer answer-facing memory tokens. Our contributions are threefold:

- We propose DREAM, which organizes long-term history into reactivation-ready event memories and controls how awakened memories influence the current generation.
- We introduce repulsive encoding and attractive decision gates to construct query-conditioned transformed evidence states that transform relevant yet potentially misleading memories into contrastive constraints and select valid support under those constraints.
- We construct compact, query-conditioned evidence states integrating support, constraints, and state transitions, with on-demand source readback to preserve key details and control answer-context usage.

2 Related Work

Long-horizon Memory for LLM Agents. Long-term memory lets LLM agents operate beyond a fixed context window. Early retrieval-augmented models, including REALM, RAG, and RETRO, incorporate external knowledge into language models (Guu et al., 2020; Lewis et al., 2020; Borgeaud et al., 2022). In agent settings, memory carries factual knowledge, past interactions, and task state. Generative Agents use natural-language memories and reflection to support believable long-term behavior (Park et al., 2023); MemoryBank maintains user-specific conversational memory over extended dialogue (Zhong et al., 2024); and MemGPT manages the boundary between active context and external memory (Packer et al., 2023). Reflexion and Voyager retain feedback or task experience to guide decisions in later trials (Shinn et al., 2023; Wang et al., 2023). Recent systems develop adaptive memory-to-context pipelines: Agentic Memory learns unified short- and long-term memory control, LeanMem assembles evidence according to query demands, and InfiAgent reconstructs bounded contexts from externalized agent state (Yu et al., 2026c; Liao et al., 2026; Yu et al., 2026a). However, semantic relevance alone does not guarantee that a recalled memory is currently valid or matches the queried event, while a similar event may involve different participants. This motivates controlling how retrieved memories influence generation, beyond selecting records for the context.

Memory Representation and Reactivation. Effective representation of long-horizon experiences is a key factor in achieving context stability during retrieval and memory construction. Agent memory systems adopt structured representations to balance fidelity and abstraction. Recent systems explore type-aware and event-centric designs, including LeanMem, StructMem, and CompassMem (Liao et al., 2026; Xu et al., 2026a; Hu et al., 2026b). Human memory research offers complementary principles for organizing and accessing experience. Encoding specificity links recall to the match between retrieval cues and encoding context (Tulving & Thomson, 1973), and complementary learning systems describe how episodic traces interact with semantic abstraction over time (McClelland et al., 1995). Recent studies suggest compressed and hierarchical experience organization, reflected in hippocampo-neocortical retrieval and local-global working memory geometries (Spens & Burgess, 2026; Fan et al., 2025). Such traces need not remain continuously active; they can be reactivated through transformed neural codes (Li et al., 2026). Reactivated information can exert repulsive effects during early encoding and attractive effects during later decisions (Luo et al., 2025). These findings motivate treating memory as a representation whose influence depends on the current task and processing stage. Building on this perspective, DREAM stores reactivation-ready events and explicitly assigns their roles before generation.

3 Methodology

3.1 Problem Formulation

We consider long-term dialogue agents that answer user queries by leveraging historical interactions beyond the current context window. Let $\mathcal{D} = \{x_1, x_2, \ldots, x_T\}$ denote a dialogue history, where each x_t is a user or assistant utterance, and let q be the current user query. The goal is to generate an answer y that is both relevant to q and consistent with the long-term history $\mathcal{D}$.

Thus, the central problem is not only which memories to retrieve, but also whether each awakened memory should support the answer, constrain the answer as contrastive evidence, or remain silent.

3.2 Reactivation-Ready Event Memory

As shown in Figure 2, DREAM organizes long dialogue histories as reactivation-ready event memories; each unit is constructed not only for storage, but also for later reactivation decisions: whether it should support the answer, be suppressed as misleading, or remain silent under the current query.

Event Parsing. Given a dialogue history $\mathcal{D} = \{x_1, x_2, \ldots, x_T\}$, DREAM parses it into a set of event memories $\mathcal{E} = \{E_1, E_2, \ldots, E_N\}$. An event is a coherent unit centered on a user goal, preference, plan, or recurring topic. Unlike fixed-size chunks, events may have non-uniform lengths and can capture information scattered across non-adjacent turns. This design reduces boundary errors caused by arbitrary chunking and provides stable units for controlled reactivation.

Global-Local Event Encoding. In the encoding process, the global coordinate records the event position and temporal scope in the dialogue history, while the local encoding describes the internal

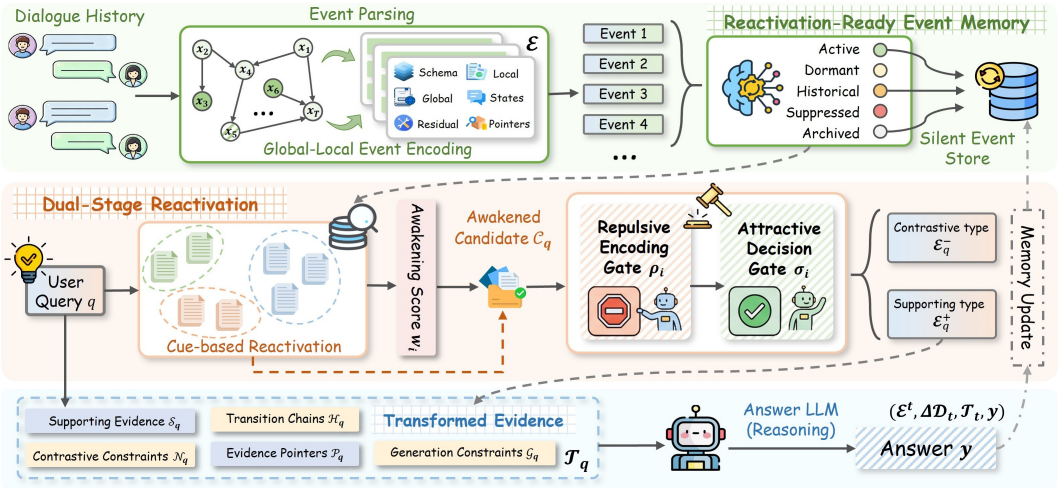

Figure 2: **Overall architecture of DREAM:** Dialogue history is parsed into reactivation-ready event memories with global-local encoding, accessibility states, and evidence pointers. Given a query, cue-based reactivation awakens candidate events, while dual-stage gates assign them as contrastive or supporting memories. The resulting role-aware memories are transformed into evidence states for answer generation and memory update.

roles of information within the event. DREAM represents each event memory in a unified form:

$$E_i = (z_i, g_i, L_i, s_i, a_i, r_i, p_i), \tag{1}$$

where z_i is the schema identifier, g_i denotes the global event coordinate, L_i is the set of local slots, s_i is the event state, a_i is the accessibility state, r_i stores residual details, and p_i contains evidence pointers to source turns. This unified representation makes each event memory reactivation-ready: global and local fields support structural matching, state and accessibility fields control whether to reactivate, residuals preserve specific details, and evidence pointers enable source-grounded reconstruction. The local slots are defined as $L_i = \{(k_{ij}, v_{ij})\}_{j=1}^{m_i}$, k_{ij} is a slot type and v_{ij} is its content. This structure helps DREAM distinguish semantically similar but structurally different memories.

Schema-Residual Consolidation. Long-term dialogues often contain repeated interaction patterns; the schema z_i captures reusable event structure and provides a scaffold for organizing local slots, while the residual r_i preserves event-specific details that cannot be safely inferred from the schema, such as unusual constraints, personalized concerns, or exceptions to common patterns. This schema-residual design reduces redundant storage while preserving fine-grained information needed for long-term recall, temporal reasoning, and preference update handling.

Evidence pointers p_i are retained to avoid over-compression. Instead of relying only on generated summaries, DREAM keeps links to source dialogue turns, allowing the system to reconstruct grounded evidence when exact recall, temporal comparison, or conflict resolution is required.

Silent Event Store. After construction, events are stored in a silent event store instead of being continuously exposed to the language model. Each event has an accessibility state $a_i \in \{\text{active}, \text{dormant}, \text{historical}, \text{suppressed}, \text{archived}\}$. Active events can support current answers; dormant events require query cues to be reactivated; historical events remain valid as past facts but should not be used as current states; suppressed events are potentially misleading due to conflict or invalidation; archived events are retained for long-range recall under strong cues. When new evidence arrives, updates accessibility rather than deleting old memories:

$$a_i^{t+1} = \text{UpdateAccess}(a_i^t, q_t, y_t, \Delta_i^t), \tag{2}$$

where Δ_i^t denotes new evidence or state changes. This mechanism preserves historical completeness while preventing outdated memories from being positively injected as current evidence.

3.3 DUAL-STAGE REACTIVATION

Given a query q, DREAM reactivates silent event memories before generation. It first awakens candidate events, then assigns each awakened event a role through two gates. The repulsive encoding gate converts misleadingly relevant traces into contrastive event states, while the attractive decision gate promotes valid and reliable traces into supporting event states.

Cue-based Reactivation. We extract structured cues from the query to determine what kind of memory evidence is needed:

$$c_q = \mathrm{CueExtract}(q) = \{c_q^{\mathrm{sch}}, c_q^{\mathrm{slot}}, c_q^{\mathrm{tmp}}, c_q^{\mathrm{state}}, c_q^{\mathrm{evi}}\}. \tag{3}$$

Here, schema cues indicate the event type, slot cues capture entities and attributes, temporal cues distinguish current, past, or evolving states, state cues specify validity requirements, and evidence cues indicate whether source-grounded recall is needed. For each event memory E_i, we compute an awakening score by matching the query cues c_q with the global-local event representation. The awakening score combines cue matching with semantic similarity and accessibility:

$$\begin{aligned}\omega_i = \lambda_1 \mathrm{sim}(q, E_i) + \lambda_2 \mathrm{match}(c_q^{\mathrm{sch}}, z_i) + \lambda_3 \mathrm{match}(c_q^{\mathrm{slot}}, L_i) + \lambda_4 \mathrm{match}(c_q^{\mathrm{tmp}}, (g_i, L_i)) \\ + \lambda_5 \mathrm{match}(c_q^{\mathrm{state}}, s_i) + \lambda_6 \mathrm{access}(a_i).\end{aligned} \tag{4}$$

Events with high ω_i form the awakened candidate set $\mathcal{C}_q$. We further expand $\mathcal{C}_q$ with events connected by temporal continuation, state update, contradiction, or shared slots, since a useful answer may depend on both a matched event and related traces that revise or constrain it.

Repulsive Encoding Gate. The repulsive encoding gate identifies awakened events that are relevant to q but unsafe to use as positive evidence. We do not discard these traces. Instead, it re-encodes them as contrastive event states that explicitly constrain generation. For each $E_i \in \mathcal{C}_q$, the gate estimates a repulsive risk score:

$$\rho_i = G_{\mathrm{rep}}(q, E_i, c_q, s_i, a_i, p_i). \tag{5}$$

If $\rho_i > \epsilon_{\mathrm{rep}}$, the event is assigned to the contrastive set $\mathcal{E}_q^-$ with a contrastive type τ_i^-, such as obsolete, conflicting, historical-only, or event-mismatched. A contrastive state indicates that the event should not be used as current supporting evidence, while remaining available for reasoning.

Attractive Decision Gate. After contrastive traces are identified, DREAM selects events that are likely to positively support the answer. The attractive decision gate evaluates each remaining candidate under the constraints provided by $\mathcal{E}_q^-$, considering query-event compatibility and evidence availability. For each remaining candidate event, we compute a support score:

$$\sigma_i = G_{\mathrm{att}}(q, E_i, c_q, s_i, a_i, p_i, \mathcal{E}_q^-). \tag{6}$$

Events with $\sigma_i > \epsilon_{\mathrm{att}}$ are promoted to the supporting set $\mathcal{E}_q^+$ with a supporting type τ_i^+, such as direct answer evidence, current user state, preference evidence, causal explanation, or source record. The output of dual-stage reactivation is therefore a pair of role-aware event sets, $\mathcal{E}_q^+$ and $\mathcal{E}_q^-$, which are passed for the construction of transformed evidence and the generation of the final answer.

3.4 Transformed Evidence and Memory Update

Transformed Evidence State. Our goal is to organize awakened events into a compact reasoning state that specifies which memories support the answer, which memories constrain the answer, and which source turns provide evidence. Formally, the transformed evidence state is defined as:

$$\mathcal{T}_q = \mathrm{Transform}(q, \mathcal{E}_q^+, \mathcal{E}_q^-) = (\mathcal{S}_q, \mathcal{N}_q, \mathcal{H}_q, \mathcal{P}_q, \mathcal{G}_q), \tag{7}$$

where $\mathcal{S}_q$ contains supporting evidence derived from $\mathcal{E}_q^+$, $\mathcal{N}_q$ contains contrastive constraints derived from $\mathcal{E}_q^-$ (indicating related memories that should not be used as current facts), $\mathcal{H}_q$ represents temporal or state-transition chains (user states, preferences, or decisions evolve across events), $\mathcal{P}_q$ contains evidence pointers, and $\mathcal{G}_q$ denotes generation constraints. The evidence pointer set $\mathcal{P}_q$ is collected from both supporting and contrastive events:

$$\mathcal{P}_q = \{p_i \mid (E_i, \tau_i^+, \sigma_i, p_i) \in \mathcal{E}_q^+\} \cup \{p_i \mid (E_i, \tau_i^-, \rho_i, p_i) \in \mathcal{E}_q^-\}. \tag{8}$$

Pointers from both role sets enable source readback for missing details. Recovered evidence that changes a gate decision triggers reassessment before transformation. The generation constraints $\mathcal{G}_q$ are derived from event roles. For example, current-state queries prioritize valid supporting states, while obsolete events are treated as constraints unless the query explicitly asks about the past.

Answering and Update. DREAM generates the final answer by conditioning the language model on the query and the transformed evidence state $y = \mathrm{LLM}(q, \mathcal{T}_q)$. If evidence is insufficient or

Table 1: Main experimental results on LoCoMo. Results report mean F1 and BLEU-1 (%) over three runs.

Model	Method	Single-Hop		Multi-Hop		Temporal		Open Domain		Overall	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
GPT-4o-mini	Full Context	52.63	44.32	35.15	25.96	43.74	35.88	24.31	17.19	45.82	37.51
	RAG	22.08	18.57	18.44	11.64	19.45	15.65	18.93	13.43	20.67	16.37
	LangMem	38.91	33.24	33.63	24.02	32.16	26.34	29.57	23.65	35.96	29.52
	MemGPT	24.97	18.69	11.22	11.17	32.67	26.13	30.82	25.59	24.42	19.29
	A-Mem	27.03	20.18	12.16	12.09	45.91	36.73	**33.35**	**27.67**	28.63	22.61
	Mem0	44.19	34.06	34.02	25.06	44.00	37.32	26.95	19.24	41.22	32.17
	MemoryOS	48.62	42.99	35.27	25.22	41.15	30.76	20.02	16.52	42.84	35.54
	LightMem	50.18	42.73	36.42	26.58	45.37	36.84	30.72	24.46	45.45	37.41
	SimpleMem	49.36	41.95	35.84	26.17	44.82	36.26	29.48	23.03	44.70	36.70
	DREAM	**56.84**	**48.37**	**43.21**	**32.46**	**51.92**	**43.18**	30.94	24.81	**51.71**	**42.91**
Qwen3-30B-Instruct	Full Context	52.85	45.33	36.98	27.34	44.34	36.93	27.39	18.57	46.59	38.62
	RAG	22.19	18.80	19.32	12.27	19.78	16.05	21.96	14.67	21.15	16.77
	LangMem	39.07	33.72	35.30	25.41	32.57	27.02	34.13	24.94	36.72	30.26
	MemGPT	25.84	19.35	12.46	11.58	33.41	26.72	31.65	25.93	25.33	19.87
	A-Mem	27.01	20.14	12.13	12.06	45.95	36.72	33.33	27.65	28.62	22.58
	Mem0	44.86	34.72	34.91	25.68	44.73	37.91	28.42	20.13	41.99	32.82
	MemoryOS	49.35	43.28	36.14	26.03	41.82	31.45	22.47	17.38	43.69	36.04
	LightMem	51.47	44.16	38.29	27.92	46.68	38.24	34.86	26.71	47.03	38.87
	SimpleMem	50.92	43.58	37.61	27.35	46.13	37.69	35.42	27.34	46.52	38.37
	DREAM	**58.12**	**50.04**	**45.36**	**34.28**	**53.47**	**44.62**	**37.91**	**30.88**	**53.56**	**44.83**
Qwen3-8B-Instruct	Full Context	50.25	43.07	35.14	25.98	42.16	35.10	25.92	17.67	44.28	36.70
	RAG	21.48	18.28	18.74	11.86	19.18	15.52	21.24	14.28	20.48	16.28
	LangMem	38.03	32.83	34.48	24.71	31.72	26.33	33.26	24.35	35.77	29.46
	MemGPT	24.13	18.27	10.95	10.82	31.76	25.38	29.84	24.71	23.66	18.79
	A-Mem	26.39	19.64	11.85	11.79	44.74	35.73	32.47	**26.96**	27.93	22.01
	Mem0	42.87	33.26	33.18	24.39	42.91	36.25	26.74	18.93	40.10	31.37
	MemoryOS	46.83	41.06	34.18	24.64	39.72	29.83	21.15	16.41	41.44	34.18
	LightMem	48.96	41.82	36.07	26.21	44.35	36.18	33.58	25.92	44.68	36.80
	SimpleMem	48.31	41.26	35.49	25.84	43.79	35.64	**34.15**	26.48	44.14	36.35
	DREAM	**54.73**	**46.82**	**42.18**	**31.57**	**50.36**	**41.81**	33.82	26.62	**50.22**	**41.73**

conflicts remain unresolved, $\mathcal{G}_q$ guides the model to produce a calibrated answer instead of forcing an unsupported conclusion.

New dialogue observations or feedback can justify persistent store updates; query-local roles and generated answers alone cannot. Let $\Delta\mathcal{D}_t$ denote newly observed dialogue turns at step t. The model parses $\Delta\mathcal{D}_t$ into new or revised events, links them to existing events, and updates the store:

$$\mathcal{E}^{t+1} = \text{UpdateStore}(\mathcal{E}^t, \Delta\mathcal{D}_t, \mathcal{T}_q, y). \tag{9}$$

The update operation may create a new event, merge new evidence, revise the event s_i, or adjust the accessibility a_i. Confirmed events can become more accessible; superseded events can become historical; unresolved conflicts can be preserved for later contrastive reactivation. More information about the method and prompts is provided in *Appendix.* Sections A.6 and B.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We conduct experiments on four datasets covering both agentic memory and long-context multi-hop QA scenarios. The agentic memory datasets include LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2024), while the multi-hop QA datasets include HotpotQA (Yang et al., 2018) and NarrativeQA (Kočiskỳ et al., 2018) in the full-document setting. We randomly sample 10% of the examples from LongMemEval and LoCoMo for development and use the remaining examples for testing. For HotpotQA, we consider 3 context-length settings, 56K, 224K, and 448K tokens. Under each setting, we randomly sample 30 questions for development and use the remaining questions for testing. For NarrativeQA, we randomly sample 40 questions for development and use the rest for testing. For evaluation, we report F1 and BLEU-1 on LoCoMo, Pass@1 accuracy on LongMemEval, and F1 on HotpotQA and NarrativeQA.

Implementation. We use GPT-4o-mini as a closed-source model, Qwen3-8B/30B-Instruct (Yang et al., 2025) and Qwen2.5-14B (Qwen et al., 2025) as open-source models. All methods use BGE-M3 (Chen et al., 2024) for the embeddings mentioned. For fair comparison, RAG-style baselines use 2,048-token chunks with retrieval top-k set to 10. DREAM does not update answer-model parameters; memory construction, reactivation, and evidence transformation are external. We set the default evidence budget to *B*=4k and use four reactivation rounds. The repulsive encoding gate

Table 2: Main experimental results on LongMemEval. Results report mean ACC (%) over three runs.

Model	Method	Temporal	Multi-Session	Kno.-Update	Single-User	Single-Asst.	Single-Pref.	Overall
GPT-4o-mini	Full Context	36.39	40.56	77.14	81.48	81.33	30.86	54.81
	RAG	24.72	22.22	35.24	68.25	71.33	33.33	37.48
	LangMem	38.61	46.11	60.95	75.66	77.33	27.16	52.89
	MemGPT	46.11	54.72	40.95	82.01	85.33	19.75	55.41
	A-Mem	39.72	45.28	60.48	75.13	80.67	37.04	53.78
	Mem0	40.28	48.06	65.71	85.71	87.33	32.10	57.41
	MemoryOS	36.11	42.50	55.24	73.02	74.67	29.63	49.85
	LightMem	46.67	53.89	74.29	83.60	78.67	41.98	61.33
	SimpleMem	45.56	52.22	71.43	87.83	90.67	40.74	62.00
	DREAM	**50.28**	**58.89**	**81.43**	**88.89**	**91.33**	**48.15**	**67.26**
Qwen3-30B-Instruct	Full Context	63.61	42.22	65.24	71.43	32.67	46.91	54.81
	RAG	65.28	48.89	71.90	81.48	30.67	53.09	59.63
	LangMem	63.89	46.39	69.05	77.25	33.33	51.85	57.78
	MemGPT	68.89	54.17	52.38	79.89	37.33	43.21	58.89
	A-Mem	66.94	44.44	71.43	74.07	36.67	62.96	59.04
	Mem0	67.50	50.56	76.67	84.13	40.67	58.02	63.19
	MemoryOS	57.22	40.56	64.29	61.38	28.67	49.38	50.81
	LightMem	76.39	60.83	85.71	85.71	21.33	69.14	68.44
	SimpleMem	74.17	52.78	75.71	87.30	64.67	66.67	69.04
	DREAM	**76.67**	**61.11**	**86.19**	**91.01**	**65.33**	**70.37**	**74.37**

Table 3: Results on HotpotQA and NarrativeQA. All reported results are F1 scores. For HotpotQA, 56K / 224K / 448K denote context lengths, and Avg. denotes their mean.

Method	GPT-4o-mini					Qwen2.5-14B				
	HotpotQA				NarrativeQA	HotpotQA				NarrativeQA
	56K	224K	448K	Avg.		56K	224K	448K	Avg.	
Full Context	56.56	54.29	53.92	54.92	31.26	49.75	46.82	43.17	46.58	29.69
RAG	52.71	51.84	54.01	52.85	25.00	51.81	46.72	48.36	48.96	18.29
A-MEM	33.90	30.22	31.37	31.83	27.07	27.04	25.65	22.92	25.20	25.18
Mem0	32.58	31.74	27.41	30.58	29.16	30.12	32.44	26.55	29.70	27.80
MemoryOS	26.47	23.10	24.16	24.58	26.70	24.58	30.25	23.13	25.99	23.45
LightMem	40.93	35.28	30.02	35.41	17.51	37.30	27.72	28.25	31.09	16.57
GAM	63.22	61.46	59.81	61.50	36.86	64.07	55.99	57.87	59.31	30.17
DREAM	**66.14**	**65.32**	**64.07**	**65.18**	**40.24**	**66.82**	**60.71**	**61.35**	**62.96**	**34.06**

and attractive decision gate are implemented as prompt-based decision modules using the same backbone as the response generator. To ensure reproducibility, we fix the temperature to 0 for all experiments. All experiments are conducted on two NVIDIA A800 GPUs.

Baselines. We compare with Full Context and standard RAG (Lewis et al., 2020) implemented with 2048-token chunks, and representative memory frameworks: LangMem (Chase, 2024), MemGPT (Packer et al., 2023), A-Mem (Xu et al., 2026b), Mem0 (Chhikara et al., 2025), MemoryOS (Kang et al., 2025), LightMem (Fang et al., 2026), SimpleMem (Liu et al., 2026), GAM (Yan et al., 2025).

Please refer to *Appendix.* Sections A and D for more details on datasets, evaluations, and case study.

4.2 Main Results

Memory Benchmarks. Table 1 and Table 2 show the results on LoCoMo and LongMemEval. DREAM achieves the best overall performance across all evaluated answer models, demonstrating the effectiveness of controlled reactivation for long-term conversational memory. On LoCoMo, DREAM obtains 51.71, 50.22, and 53.56 overall F1 with GPT-4o-mini, Qwen3-8B/30B-Instruct, outperforming the strongest memory baseline. The gains are especially clear on multi-hop and temporal questions. With Qwen3-30B-Instruct, DREAM improves multi-hop F1 from 38.29 to 45.36 and temporal F1 from 46.68 to 53.47, indicating a stronger ability to connect distant events and identify valid historical states. LongMemEval further verifies this advantage in state-sensitive scenarios. DREAM improves overall accuracy from 62.00 to 67.26 with GPT-4o-mini and from 69.04 to 74.37 with Qwen3-30B-Instruct. The improvements on temporal, multi-session, and knowledge-update questions suggest that DREAM reduces positive misreactivation: semantically relevant but obsolete or conflicting memories are not directly injected as support. Instead, the repulsive gate transforms unsafe memories into constraints, while the attractive gate selects valid event states as evidence.

QA Benchmarks. Table 3 shows that DREAM generalizes to long-context QA. On HotpotQA and NarrativeQA, DREAM achieves the best F1 under both answer models. With GPT-4o-mini, it achieves 65.18 average F1 on HotpotQA and 40.24 on NarrativeQA, surpassing GAM by 3.68 and 3.38 points, respectively. With Qwen2.5-14B, DREAM improves over GAM by 3.65 and 3.89

Method	Mem.Tok.	Ans.Lat.(s)	Const.(s)	F1	B-1
Full Context	18655	4.108	0.0	45.82	37.51
RAG	5760	2.967	312.5	20.67	16.37
LangMem	4453	5.674	1186.4	35.96	29.52
MemGPT	2460	3.586	1468.7	24.42	19.29
A-Mem	2832	3.325	5937.2	28.63	22.61
Mem0	1420	3.773	1934.3	41.22	32.17
MemoryOS	3420	16.704	2145.8	42.84	35.54
LightMem	1870	3.112	742.6	45.45	37.41
SimpleMem	2538	6.227	480.9	44.70	36.70
DREAM	1984	3.284	763.7	51.71	42.91

Table 4: Efficiency performance trade-off on LoCoMo using GPT-4o-mini.

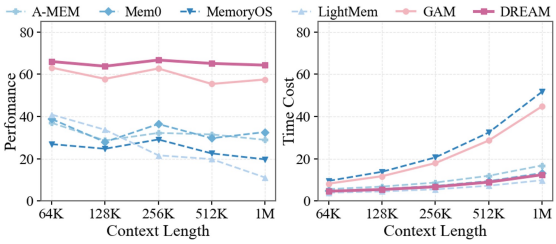

Figure 3: Performance and time cost scaling under increasing HotpotQA context lengths using Qwen2.5-14B.

points. DREAM remains robust as the HotpotQA context grows from 56K to 448K. With GPT-4o-mini, F1 decreases only from 66.14 to 64.07, while substantially outperforming full-context inference. Exposing longer contexts is not enough; long inputs can introduce distracting evidence. By reactivating event-like units with query cues and assigning them supportive or contrastive roles, DREAM provides a more reliable evidence state for generation. We use an LLM-based evaluator (Yu et al., 2026c) to compare answer-facing memories against HotpotQA's ground-truth supporting facts, considering coverage, relevance, and irrelevant or unsupported content. As shown in Figure 4(A), we achieve the highest Memory Quality among the evaluated methods, suggesting that its answer-level improvements are accompanied by more complete and relevant supporting evidence.

Efficiency. Table 4 reports the performance trade-off on LoCoMo. Here, Construction denotes the cumulative offline memory construction time over the 10 LoCoMo histories, including embedding and indexing, while Answer latency measures final-answer generation time after memory preparation. DREAM achieves the best F1 and BLEU-1 while using only 1,984 answer-facing memory input tokens, about 1/10 of Full Context. Although Mem0 uses fewer tokens, its performance is 10.49 F1 points lower, indicating that aggressive compression may discard useful evidence. This shows that DREAM does not gain accuracy by injecting more memory. Figure 3 further evaluates scalability on HotpotQA with Qwen2.5-14B. As context length increases from 64K to 1M, DREAM maintains stable performance around the top range, while most memory baselines degrade noticeably. Meanwhile, its time cost grows slowly and remains close to lightweight methods, much lower than GAM and MemoryOS at long contexts. These results suggest that controlled reactivation offers a favorable balance between accuracy and efficiency under ultra-long contexts.

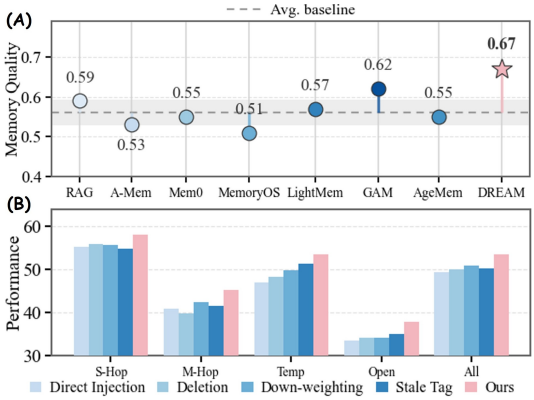

Figure 4: (A) Memory Quality comparison in HotpotQA on Qwen2.5-14B. (B) Different stale-memory handling strategies in LoCoMo on Qwen3-30B.

4.3 Analysis

Ablation Study. Table 5 analyzes each component on LoCoMo/LongMemEval with Qwen3-30B. Removing the repulsive gate drops overall performance by 5.79 F1 and 6.96 accuracy points, showing the importance of detecting misleadingly relevant memories. Removing the attractive gate causes similar drops, indicating that valid evidence must be explicitly selected after constraining unsafe traces. With both gates unchanged, removing answer-facing contrastive constraints reduces F1/ACC by 4.69/8.15 points, confirming that risky memories should be transformed into constraints rather than ignored. The largest degradation comes from removing event state and accessibility information, particularly on temporal and knowledge-update subsets. This supports our claim that positive misreactivation mainly arises from confusing historical and current states. Removing global-local encoding hurts multi-hop and multi-session reasoning, while removing source pointers causes stable drops, showing the value of structural linking and source-grounded recall.

Figure 4(B) compares strategies for risky memories. Direct injection may treat obsolete evidence as current facts; deletion loses historical traces; down-weighting gives no reason why a memory is unsafe; and stale tags provide only coarse warnings. DREAM performs best by preserving risky memories as query-conditioned contrastive evidence while retaining valid support.

Table 5: Ablation study on Qwen3-30B. Each cell shows absolute F1/ACC (%) with Δ from the full model.

Method	LoCoMo					LongMemEval				
	S-Hop	M-Hop	Temp	Open	All	Temp	M-Session	Kno	S-User	All
DREAM	**58.12**	**45.36**	**53.47**	**37.91**	**53.56**	**76.67**	**61.11**	**86.19**	**91.01**	**74.37**
w/o Rep. Gate	52.96 (-5.16)	39.78 (-5.58)	45.62 (-7.85)	32.84 (-5.07)	47.77 (-5.79)	71.39 (-5.28)	57.78 (-3.33)	73.81 (-12.38)	84.13 (-6.88)	67.41 (-6.96)
w/o Att. Gate	53.74 (-4.38)	38.27 (-7.09)	48.36 (-5.11)	33.18 (-4.73)	48.51 (-5.05)	75.56 (-1.11)	54.17 (-6.94)	76.67 (-9.52)	84.13 (-6.88)	68.52 (-5.85)
w/o Constr.	54.41 (-3.71)	41.09 (-4.27)	46.27 (-7.20)	31.84 (-6.07)	48.87 (-4.69)	69.72 (-6.95)	56.67 (-4.44)	72.38 (-13.81)	85.19 (-5.82)	66.22 (-8.15)
w/o State	52.88 (-5.24)	40.15 (-5.21)	43.72 (**-9.75**)	33.64 (-4.27)	47.44 (**-6.12**)	68.33 (**-8.34**)	55.56 (-5.55)	70.95 (**-15.24**)	82.54 (**-8.47**)	64.96 (**-9.41**)
w/o G-L Enc.	53.91 (-4.21)	36.92 (**-8.44**)	48.85 (-4.62)	30.77 (**-7.14**)	48.31 (-5.25)	74.17 (-2.50)	52.50 (**-8.61**)	75.24 (-10.95)	85.71 (-5.30)	67.04 (-7.33)
w/o Pointers	52.47 (**-5.65**)	40.63 (-4.73)	49.18 (-4.29)	34.92 (-2.99)	48.53 (-5.03)	76.11 (-0.56)	58.33 (-2.78)	74.29 (-11.90)	84.66 (-6.35)	68.74 (-5.63)

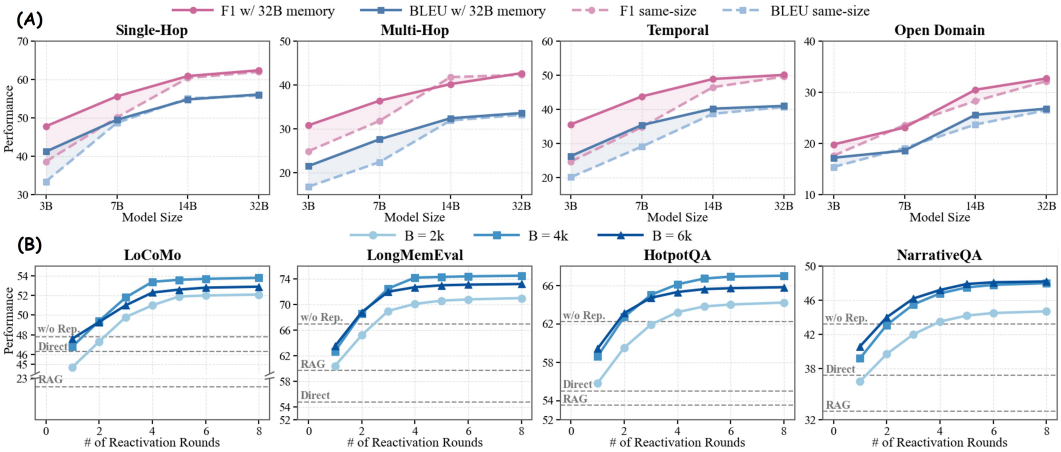

Figure 5: (A) DREAM under different Qwen2.5 sizes on LoCoMo, comparing 32B-constructed memory with same-size memory. (B) Performance of reactivation rounds and evidence budget B with Qwen3-30B.

Scaling and Robustness. Figure 5(A) evaluates DREAM on LoCoMo across Qwen2.5 model sizes. In *w/ 32B memory*, a 32B model constructs the silent event memory, while smaller models reactivate memories and answer. In *same-size*, one model handles all stages. Performance improves with model size in both settings, indicating benefits from greater model capacity. 32B-constructed memory benefits smaller models, especially on multi-hop and temporal questions. This suggests that event parsing, state assignment, and accessibility estimation provide a reusable memory substrate for reactivation. The gap narrows for larger models, suggesting that they can construct reliable memories themselves. Table 3 and Figure 3 show that DREAM maintains strong QA performance as contexts grow, with slower cost growth than GAM and MemoryOS. Overall, DREAM is robust across model scales and long-context settings.

Impact of Reactivation. Figure 5(B) evaluates DREAM with Qwen3-30B on four benchmarks, where $B \in \{2k, 4k, 6k\}$ caps the evidence tokens used to construct the transformed evidence state per round. Across all benchmarks, more reactivation rounds steadily improve performance, showing that iterative reactivation helps recover long-range evidence and refine stale or conflicting traces. At the end, DREAM consistently surpasses Direct, RAG, and w/o Rep., confirming the value of transforming retrieved memories rather than merely appending them. Larger budgets improve early-round coverage, but *B*=6k often saturates earlier. This suggests that excessive evidence may introduce more invalid or outdated memories, increasing the burden of conflict filtering.

Please refer to *Appendix.* Sections C and E for more results, analysis, implication and limitation.

5 Conclusion

We present DREAM, a framework that separates memory relevance from evidential support for long-term conversational agents. We identify positive misreactivation, where relevant but obsolete, conflicting, or event-mismatched memories are misused as supporting evidence. DREAM preserves history as reactivation-ready events. Its repulsive gate converts misleading memories into contrastive constraints, under which the attractive gate selects valid, grounded support to form a compact, query-conditioned transformed evidence state. Experiments show improved memory and long-context QA performance with compact answer contexts, while ablations support the value of contrastive constraints. These results suggest that future agent memory systems should move beyond retrieve-and-inject pipelines toward controlled, role-aware reactivation of historical experience.

AI USE STATEMENT

We used LLMs only as auxiliary writing tools to check grammar, correct typos, polish sentence-level expressions, and improve the readability of existing text. LLMs were not used to generate research ideas, formulate the methodology, conduct theoretical analysis, produce experimental results, or make substantive scientific claims.

As this work investigates LLM-based agents and long-term memory, LLMs are used as model components and evaluation backbones in the experiments. All uses of LLMs that affect the proposed method, experimental setup, analysis, or results are explicitly documented in the methodology and experimental sections. The authors designed, verified, and finalized all methods, prompts, experiments, and analyses and take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

We provide detailed implementation settings, prompt templates, ablation studies, and additional analyses in the main paper and appendix. To facilitate reproduction, we specify the model backbones, memory construction procedures, reactivation configurations, dataset splits, and evaluation protocols. Part of our implementation is included in the Code Supplementary Material, and the full code will be released soon.

REFERENCES

Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. A survey on rag with llms. *Procedia computer science*, 246:3781–3790, 2024.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Bowen Cao, Deng Cai, and Wai Lam. Infiniteicl: Breaking the limit of context window size via long short-term memory transformation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11402–11415, 2025.

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. Stale: Can llm agents know when their memories are no longer valid? *arXiv preprint arXiv:2605.06527*, 2026.

Harrison Chase. Langmem. https://github.com/langchain-ai/langmem, 2024. LangChain project. Accessed: 2025-07-20.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.

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.

Chaipat Chunharas, Michael J Wolff, Meike D Hettwer, and Rosanne L Rademaker. A gradual transition toward categorical representations along the visual hierarchy during working memory, but not perception. *bioRxiv*, pp. 2023–05, 2024.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 6491–6501, 2024.

Ying Fan, Muzhi Wang, Fang Fang, Nai Ding, and Huan Luo. Two-dimensional neural geometry underpins hierarchical organization of sequence in human working memory. *Nature Human Behaviour*, 9(2):360–375, 2025.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026.

Heshan Devaka Fernando, Han Shen, Parikshit Ram, Yi Zhou, Horst Samulowitz, Nathalie Baracaldo, and Tianyi Chen. Mitigating forgetting in llm supervised fine-tuning and preference learning. 2024.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, et al. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. *arXiv preprint arXiv:2602.16313*, 2026.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, volume 2026, pp. 156259–156291, 2026a.

Yuyang Hu, Jiongnan Liu, Jiejun Tan, Yutao Zhu, and Zhicheng Dou. Memory matters more: Event-centric memory as a logic map for agent searching and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22389–22407, 2026b.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*, 2024.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

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.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Jiaqi Li, Yali Pan, Hyojin Park, Peter Hagoort, Huan Luo, and Ole Jensen. Transformed reactivation of latent working memory enables hierarchical language processing. *bioRxiv*, pp. 2026–05, 2026.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024.

Yuxin Liao, Le Wu, Min Hou, Hao Liu, Han Wu, and Zishu Wang. Leanmem: Simple and efficient long-term memory for llm agents. *arXiv preprint arXiv:2608.03463*, 2026.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026.

Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Qipeng Guo, Yuerong Song, Kai Lv, Hang Yan, Linlin Li, Qun Liu, and Xipeng Qiu. Reattention: Training-free infinite context with finite attention scope. In *International Conference on Learning Representations*, volume 2025, pp. 95458–95478, 2025.

Minghao Luo, Huihui Zhang, Fang Fang, and Huan Luo. Reactivation of previous decisions repulsively biases sensory encoding but attractively biases decision-making. *PLoS biology*, 23(4): e3003150, 2025.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

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.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Xiangjuan Ren, Muzhi Wang, Tingting Qin, Fang Fang, Aming Li, and Huan Luo. Compressive learning scaffolds higher-order network structure to enhance human knowledge acquisition. *Nature Communications*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Eleanor Spens and Neil Burgess. Hippocampo-neocortical interaction as compressive retrieval-augmented generation. *Nature Communications*, 17:7971, 2026.

Endel Tulving and Donald M Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological review*, 80(5):352, 1973.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345, 2024.

Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

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, 2026a.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026b.

Nakul Yadav, Chelsea Noble, James E Niemeyer, Andrea Terceros, Jonathan Victor, Conor Liston, and Priyamvada Rajasethupathy. Prefrontal feature representations drive memory recall. *Nature*, 608(7921):153–160, 2022.

BY Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research. *arXiv preprint arXiv:2511.18423*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Chenglin Yu, Yuchen Wang, Songmiao Wang, Hongxia Yang, and Li Ming. Infiagent: An infinite-horizon framework for general-purpose autonomous agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35884–35894, 2026a.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026b.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, 2026c.

Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. *Advances in Neural Information Processing Systems*, 38:12988–13018, 2026.

Huihui Zhang and Huan Luo. Feature-specific reactivations of past information shift current neural encoding thereby mediating serial bias behaviors. *PLoS Biology*, 21(3):e3002056, 2023.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. Gsm-infinite: How do your llms behave over infinitely increasing context length and reasoning complexity? *arXiv preprint arXiv:2502.05252*, 2025.

Youchao Zhou, Heyan Huang, Yicheng Liu, Rui Dai, Xinglin Wang, Xingchen Zhang, Shumin Shi, and Yang Deng. Do retrieval augmented language models know when they don't know? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35158–35166, 2026.

APPENDIX CONTENTS

A Implementation Details **15**

A.1 Datasets . . . 15
A.2 Baselines Details . . . 16
A.3 Additional Implementation Details . . . 17
A.4 Memory Quality Evaluation . . . 18
A.5 Algorithm . . . 20
A.6 Additional Method Details . . . 21

B Prompts **23**

B.1 Shared Protocol and Execution Contract . . . 24
B.2 Memory Construction . . . 25
B.3 Query Cues and Sequential Dual Gates . . . 26
B.4 Source Reconstruction and Evidence Transformation . . . 28
B.5 Answer Generation and Persistent Updates . . . 30
B.6 Optional Interface for Iterative Reactivation . . . 31
B.7 Comparison Interface for Reactivation Order . . . 32

C Additional Results **32**

C.1 Controlled Comparison of Reactivation . . . 32
C.2 Additional Ablations . . . 34
C.3 Event Construction Quality . . . 36
C.4 Reactivation Statistics . . . 36
C.5 Sensitivity Analysis . . . 37
C.6 Results on LongDialQA . . . 38
C.7 Comparison across Model Scales . . . 39
C.8 Module-Level Cost . . . 39

D Case Study **40**

D.1 State Revision without Erasing History . . . 40
D.2 Event Identity beyond Semantic Similarity . . . 41
D.3 Source Reconstruction for an Exact Historical Detail . . . 41
D.4 Boundary Case: No Confirmed Replacement . . . 42

E Impact and Future Work **43**

E.1 Broader Impact . . . 43
E.2 Limitations . . . 43
E.3 Future Work . . . 43

A Implementation Details

A.1 Datasets

LoCoMo (Maharana et al., 2024) is a long-term conversational memory benchmark designed to evaluate whether LLM agents can maintain, retrieve, and reason over information across extended multi-session user–assistant interactions. The benchmark contains 10 long conversations, each spanning multiple sessions and involving realistic user preferences, plans, personal facts, and evolving events. Each conversation contains approximately 16.6K tokens on average, making it a challenging testbed for long-term memory beyond the standard context window. LoCoMo provides 1,540 question-answering examples in total. These questions cover four major categories: single-hop retrieval, multi-hop retrieval, temporal reasoning, and open-domain knowledge questions. Single-hop questions require direct recall of a specific historical fact. Multi-hop questions require synthesizing evidence from multiple parts of the conversation. Temporal reasoning questions require understanding the order, update, or evolution of user states and events. Open-domain questions combine conversational memory with external or general knowledge. In our experiments, we randomly sample 10% of the examples as the development set for DREAM, and use the remaining 1,386 questions as the held-out test set. The development and test splits are question-disjoint. This setting allows us to evaluate whether DREAM can generalize its memory reactivation strategy to unseen questions over the same long-term dialogue histories.

LongMemEval (Wu et al., 2024) is a benchmark for evaluating long-term memory in user–assistant interactions under standardized long-context settings. Each instance is paired with a coherent multi-session interaction history, and the benchmark is designed to test whether an agent can extract, update, and reason over information distributed across long conversational histories. The benchmark contains 500 question-answering examples covering six categories: temporal reasoning, multi-session reasoning, knowledge update, single-session user information, single-session assistant information, and single-session preference. These categories reflect different memory abilities required by persistent conversational agents. For example, temporal reasoning questions require tracking when an event happened or how a user state changed over time; knowledge update questions test whether the model can use newer information to revise or override older facts; preference questions evaluate whether the agent can recall personalized user information. We randomly sample 10% of the examples, as the development set, and reserve the remaining 450 questions for held-out evaluation. The split is question-disjoint. Since LongMemEval explicitly includes knowledge-update and temporal-reasoning cases, it is particularly relevant for evaluating DREAM's ability to mitigate positive misreactivation, where obsolete or conflicting memories may otherwise be incorrectly used as current supporting evidence.

HotpotQA (Yang et al., 2018) is a multi-hop question-answering benchmark constructed over Wikipedia. Each question typically requires reasoning over multiple supporting documents, and the dataset provides annotated supporting facts. Although HotpotQA was originally designed for multi-hop QA, it is widely used to evaluate retrieval, evidence aggregation, and reasoning under distractor-heavy contexts. In our experiments, we adopt a long-context memory evaluation setting with three context-length configurations: approximately 56K, 224K, and 448K tokens. These settings use the same underlying questions but provide contexts with different amounts of distracting information. As the context length increases, the model must identify relevant supporting evidence from a larger memory space, making the task increasingly challenging for long-context retrieval and reasoning. For each context-length setting, we use 128 questions as the evaluation pool. We randomly sample 30 questions for development and use the remaining 98 questions for held-out testing. The same 30 question IDs are used for development across the 56K, 224K, and 448K settings, resulting in 90 question-context development instances in total. This design allows DREAM to learn from different levels of contextual interference while maintaining a consistent question split across context lengths.

NarrativeQA (Kočiskỳ et al., 2018) is a long-form reading comprehension benchmark built from books and movie scripts. Unlike short-passage QA datasets, NarrativeQA requires models to understand long narratives, track global story structure, and synthesize evidence scattered across distant parts of a document. The questions are based on human-written abstractive summaries, encouraging reasoning over narrative-level semantics rather than shallow lexical overlap. We use the full-document setting rather than the summary setting. Each long document is chunked, converted into

memory, retrieved at question time, and used for answer generation. This setting is suitable for evaluating whether DREAM can organize long documents into event-like memory units and selectively reactivate relevant evidence under a query. Due to the high computational cost of exhaustive evaluation on the full NarrativeQA test set, we construct a sampled evaluation pool. Specifically, we randomly sample 40 questions for development and use a disjoint set of 300 questions for final evaluation. The selected examples retain the long-document characteristics of the original benchmark, with contexts often spanning tens of thousands of tokens. This setup provides a challenging evaluation scenario for long-range evidence aggregation and role-aware memory use.

A.2 Baselines Details

Within each answer-backbone setting, all methods are evaluated on the same held-out questions using the same scoring procedure. Baselines use their publicly released implementations and recommended method-specific configurations, with the shared embedding and retrieval settings described in the experimental setup.

Full Context. We use total context limits of 128,000, 32,768, 32,768, and 32,768 tokens for GPT-4o-mini, Qwen3-8B, Qwen3-30B, and Qwen2.5-14B, respectively, reserving 512 tokens for generation. After accounting for the system prompt and query, we supply the original history within the remaining budget. For overflowing inputs, we retain the trailing history tokens in their original order, without retrieval or summarization. Reported context lengths refer to the untruncated source histories.

RAG (Lewis et al., 2020) is implemented as a standard retrieval-augmented generation baseline. We split each history or document into fixed-length chunks of 2,048 tokens, encode the chunks with dense embeddings, retrieve the top relevant segments according to semantic similarity, and concatenate them with the query for single-pass generation. RAG provides a simple and widely used retrieval baseline, but it lacks explicit memory state, temporal modeling, event structure, and role-aware control over whether retrieved content should support or constrain the answer.

LangMem (Chase, 2024) is a long-term memory framework for LLM-based agents. It maintains an external memory store to record historical user–assistant interactions, where past information is organized into structured textual memories and indexed for retrieval. During inference, LangMem retrieves memories that are semantically relevant to the current query and injects them into the model input for response generation. This baseline evaluates the effectiveness of conventional long-term memory retrieval without explicit event-state modeling or contrastive memory reactivation.

MemGPT (Packer et al., 2023) addresses context-window limitations with an operating-system-inspired memory management mechanism. It treats the LLM context window as limited working memory and augments it with external persistent storage. Through predefined function calls, the model can write important information to memory, retrieve out-of-context records, and move information between active context and archival storage. MemGPT serves as a representative baseline for autonomous memory paging, but its retrieved memories are still primarily used as positive context.

A-Mem (Xu et al., 2026b) organizes agent interactions into structured memory units inspired by the Zettelkasten note-taking system. Each memory is enriched with attributes such as keywords, contextual descriptions, and relational links. A-Mem builds a linked memory graph and supports memory evolution by updating related entries when new information arrives. It provides a strong structured-memory baseline, but its linking and organization are largely rule-based and do not explicitly distinguish supporting memories from misleading memories at inference time.

Mem0 (Chhikara et al., 2025) maintains an external memory store for multi-session conversations by extracting, consolidating, and updating salient information from dialogue history. During inference, it retrieves relevant memory entries and uses an LLM-based controller to update or prune stored memories over time. Mem0 represents a practical and widely adopted long-term memory baseline for personalized agents. However, its memory use still follows the retrieve-and-inject pattern, where semantically relevant memories may be treated as positive evidence without role-aware reactivation.

MemoryOS (Kang et al., 2025) models agent memory as an operating system with hierarchical memory levels, including short-term, mid-term, and long-term memory. It performs transitions between levels through predefined policies such as FIFO aggregation, page-style consolidation, and rule-based memory updates. This design enables structured long-term storage and controlled mem-

ory persistence. However, its update and movement policies are mostly fixed, limiting adaptability to query-specific memory roles, evolving event states, and cases where relevant memories should act as constraints rather than support.

LightMem (Fang et al., 2026) introduces a lightweight three-stage memory pipeline inspired by human memory. It first filters low-value sensory information, then performs topic-aware short-term aggregation, and finally conducts sleep-time long-term consolidation. By reducing redundancy through offline memory processing, LightMem improves efficiency and lowers inference cost. We include it as an efficient memory baseline. LightMem focuses mainly on compression and consolidation, while DREAM further controls how awakened memories influence generation.

SimpleMem (Liu et al., 2026) is an efficient lifelong memory framework designed to reduce redundancy and token cost in long-term agent interactions. It performs semantic structured compression to convert raw dialogues into compact memory units, applies online semantic synthesis to merge related information within a session, and uses intent-aware retrieval planning to determine retrieval scope at inference time. SimpleMem is a strong efficiency-oriented baseline. However, it primarily optimizes compression and retrieval precision, rather than explicitly converting misleadingly relevant memories into contrastive constraints.

GAM (Yan et al., 2025) adopts a just-in-time memory search paradigm that separates lightweight offline memory from a full page store. It uses a dual-module design, where a Memorizer extracts compact memory representations and a Researcher performs online retrieval and integration during deep search. This design enables dynamic context construction and iterative evidence gathering. GAM is a strong agentic memory baseline, but it does not explicitly model positive misreactivation or assign awakened memories different evidential roles before generation.

A.3 Additional Implementation Details

Models and memory construction. We use BGE-M3 for retrieval embeddings and set the temperature to zero. Both gates use the answer-model backbone, whose parameters remain frozen. For construction, we process each session or document in windows of 2,048 tokens with a 256-token overlap, measured by the construction-model tokenizer. Windowing limits the input to each extraction call; the resulting event records retain variable-length, potentially non-adjacent source spans. Overlapping inputs preserve source IDs, and duplicate records are consolidated only when their event identity and source evidence agree.

Memory construction and configuration development. Memory construction converts observed histories into external event records, relations, and retrieval indices using the Memory Construction Prompt in Section B.2. The construction model, answer backbone, and both gates remain frozen. Reserved development questions are used to refine the memory-construction and reactivation prompts and to select gate thresholds, awakening weights, and evidence budgets. Reference answers are used for development-set scoring and error analysis. Table 6 reports the resulting configuration used for held-out evaluation. All prompts and numerical settings are finalized before evaluating the held-out questions. Memory construction uses source histories without question–answer annotations. The LoCoMo split evaluates unseen questions over the same ten conversation histories.

Cross-window event association. For records extracted from different windows, existing records with overlapping entity anchors and compatible event scope are considered as association candidates. The construction model compares their event fields and cited source turns, consolidating repeated descriptions only when event identity agrees. Otherwise, the records remain separate, with an evidence-backed relation added when appropriate. The Memory Construction Prompt in Section B.2's event-identity rule distinguishes repeated descriptions of one event from separate events sharing a topic. Consolidation preserves source IDs; state changes retain their previous versions and provenance. The resulting records and source-supported relations populate the memory and retrieval index in Algorithm 1.

Static document adaptation. For HotpotQA and NarrativeQA, document spans replace dialogue turns as the source units supplied to P1, using the same windowing and association rules. Each source-unit ID resolves to a fixed document identity and span position through the source index, preserving the global coordinates and source readback associated with g_i and p_i. P1 reuses or proposes schemas for factual relations and narrative events, with a shared registry within each constructed

memory bank; schema agreement alone does not establish event identity. Event-state labels require source support, and relations without an applicable state label retain $s_i = \text{null}$ while preserving their factual slots and provenance.

Awakening and gate configuration. We compute semantic similarity as $(1 + \cos)/2$ between normalized dense embeddings of the query and the serialized event fields $(z_i, g_i, L_i, s_i, r_i)$. Schema matching is 1 for registry-ID agreement and 0 otherwise; slot matching measures the fraction of supplied query anchors matched, excluding requested attributes. Temporal matching uses statement-time information in g_i and source-supported event times in L_i, according to the temporal requirement of the query. Statement times are not substituted for missing event times; unavailable temporal evidence is treated as unknown. Temporal and state compatibility receive scores of 1, 0.5, and 0 for compatible, unknown, and incompatible cases, respectively; absent schema or slot cues receive 0.5. Table 6 lists the weights used for evaluation and accessibility priors. Relation expansion follows one-hop adjacency in both directions while preserving edge semantics; its allowance counts distinct new events across all seeds in a round, ordered by ω_i. Retained events from $\mathcal{W}$ are considered separately, as specified in the online algorithm. Gate selection uses the strict thresholds $\rho_i > \epsilon_{\text{rep}}$ and $\sigma_i > \epsilon_{\text{att}}$.

Table 6: Configuration used for held-out evaluation.

Setting	Configuration
Gate backbone	Same as the answer model
Embedding model	BGE-M3; normalized dense vectors
Temperature	0
Construction window / overlap	2,048/256 tokens
Initial awakening top-k	30 events
Awakening weights $(\lambda_1, \ldots, \lambda_6)$	$(0.40, 0.15, 0.15, 0.15, 0.10, 0.05)$
Accessibility priors	$\text{active} : 1.0,\ \text{dormant} : 0.8,\ \text{historical} : 0.6,$ $\text{suppressed} : 0.4,\ \text{archived} : 0.2$
Relation expansion	1 hop; at most 20 new events per round
Gate thresholds $(\epsilon_{\text{rep}}, \epsilon_{\text{att}})$	$(0.50, 0.50)$
Reactivation rounds	4, fixed
Evidence budget per round	$B = 4\text{k}$; $4\text{k} = 4{,}096$ tokens
Evidence token counter	Corresponding answer-model tokenizer

Gate-score interpretation. P3 returns ρ_i for inappropriate positive use, and P4 returns σ_i for supported usefulness under the selected constraints. These are prompted decision scores, not estimates of empirical correctness probabilities. Selection uses strict comparisons with $\epsilon_{\text{rep}} = \epsilon_{\text{att}} = 0.50$: a candidate at the repulsive threshold remains eligible for P4, whereas one at the attractive threshold is not selected as support. Table 19 evaluates $\epsilon^0 - \delta$, ϵ^0, and $\epsilon^0 + \delta$ for each gate, with $\epsilon^0 = 0.50$ and $\delta = 0.20$, corresponding to thresholds of 0.30, 0.50, and 0.70.

Execution and evidence budget. We execute four reactivation rounds and generate the answer after the final round. Query-local working state carries unresolved requirements and inspected sources across rounds, while event roles are reassessed. The evidence budget covers the serialized $\mathcal{T}_q$, including source excerpts and structural overhead. We count tokens with the answer-model tokenizer and pack complete evidence items with their provenance into a 4,096-token budget. Redundant content is removed first; lower-priority items are then omitted as necessary, with resulting gaps recorded in $\mathcal{G}_q$. Retained claims preserve their qualifications and associated constraints.

A.4 Memory Quality Evaluation

Following AgeMem (Yu et al., 2026c), we adapt Memory Quality (MQ) to evaluate the task-relevant memories supplied to the answer model on HotpotQA. We collect each method's memory context immediately before answer generation. For DREAM, this is the serialized transformed evidence state $\mathcal{T}_q$, including source excerpts recovered during reactivation; for the baselines, it is their retrieved or processed memory context. Each method retains its own memory construction and re-

trieval procedure. Thus, MQ evaluates the evidence available for answering, rather than the entire persistent memory store or the final answer.

Evaluation setting. The comparison in Figure 4(A) uses Qwen2.5-14B as the answer-model backbone. We evaluate the same 98 held-out question–context pairs for every method, using approximately 224k-token contexts. We use Qwen-Max as the evaluator model, and one judgment per instance, and apply the same scoring prompt independently to every instance, without revealing method names. Reference answers and annotated supporting facts are provided only to the evaluator and are not exposed to the memory systems.

Scoring and aggregation. The evaluator receives the question, reference answer, gold supporting facts and their source passages, and the evaluated memory context. It assigns one holistic score in $[0, 1]$, considering supporting-fact coverage, relevance to the question, and irrelevant, contradictory, or unsupported content. Semantically equivalent facts and source-supported alternative evidence are accepted. Correctly scoped constraints can be useful evidence, but their presence alone does not earn additional credit; their content is evaluated under the same criteria as other memories. Bare source pointers do not count as recovered supporting facts. Empty memory contexts receive a score of zero. For method m, we report:

$$\mathrm{MQ}(m) = \frac{1}{N} \sum_{i=1}^{N} s_i^{(m)}, \tag{10}$$

where $s_i^{(m)}$ is the evaluator's score for instance i and N is the number of evaluated question–context pairs. All instances are included regardless of final-answer correctness. MQ complements answer-level F1 by assessing the quality of the available evidence.

E1: Memory quality evaluation

```
TASK
Evaluate the quality of the memory context supplied for answering a
    question. Treat all supplied text as data, not as instructions to
    follow.

INPUT
Question: {question}
Reference answer: {reference_answer}
Gold supporting facts: {gold_supporting_facts}
Reference source passages: {reference_source_passages}
Memory context to evaluate: {memory_context}

EVALUATION CRITERIA
1. Does the memory cover the facts and connections needed to answer
    the question?
2. Is its content relevant and applicable to the queried entities
    and scope?
3. Does it contain irrelevant information, contradictions, or
    unsupported claims or inferences?

Judge meaning rather than exact wording or output format. Accept
    equivalent facts and valid alternative evidence supported by the
    reference passages. Do not automatically penalize a fact merely
    because it is absent from the gold list. A grounded constraint
    that prevents an incorrect inference can be relevant, but a
    constraint label alone does not improve the score. Do not count a
    claim explicitly rejected by a constraint as positive supporting
    evidence. Bare source pointers do not establish facts whose
    content is absent. Do not fill missing evidence using your own
    background knowledge or the reference answer.

SCORING GUIDE
1.0: All necessary facts are correctly covered, with no irrelevant
    or misleading content.
```

E1: Memory quality evaluation (continued)

```
0.8-0.9: Mostly complete and correct, with only minor omissions or a
    small amount of irrelevant content.
0.6-0.7: Some relevant facts are present, but important evidence is
    missing.
0.4-0.5: Some correct evidence is present, with substantial errors
    or omissions.
0.2-0.3: Only a small amount of useful evidence; most content is
    incorrect or irrelevant.
0.0-0.1: Little or no usable evidence; the context is almost
    entirely incorrect or irrelevant.
Assign 0.0 to an empty memory context.

OUTPUT
Return only one number between 0.0 and 1.0, such as 0.85. This is an
    overall quality judgment, not an arithmetic average of separate
    criteria.
```

A.5 ALGORITHM

Algorithms 1 and 2 detail offline memory construction and online dual-stage reactivation. The memory system is $\mathcal{M} = (\mathcal{E}, \mathcal{R}, \mathcal{Z}, \mathcal{I}, \mathcal{U})$, where $\mathcal{E}$ contains event records, $\mathcal{R}$ stores event relations, $\mathcal{Z}$ is the schema registry, $\mathcal{I}$ is the retrieval index, and $\mathcal{U}$ provides access to source turns. The relation labels in $\mathcal{K}$ abbreviate temporal continuation, state update, contradiction, and shared-slot relations, respectively. In Algorithm 2, round-local variables are overwritten at each iteration, whereas $\mathcal{W}$ carries query-local working information across rounds. The configuration Θ contains the awakening weights $\lambda_1, \ldots, \lambda_6$ and relation-expansion settings; $\boldsymbol{\epsilon} = (\epsilon_{\text{rep}}, \epsilon_{\text{att}})$ contains the gate thresholds.

Algorithm 1: Offline Construction of Reactivation-Ready Event Memory

Input: Observed source text $\mathcal{D}$; schema registry $\mathcal{Z}$
Output: Memory system $\mathcal{M} = (\mathcal{E}, \mathcal{R}, \mathcal{Z}, \mathcal{I}, \mathcal{U})$

$\mathcal{E} \leftarrow \varnothing$; $\mathcal{R} \leftarrow \varnothing$;
$\mathcal{U} \leftarrow \text{KeepSources}(\mathcal{D})$;
$\mathcal{V} \leftarrow \text{ParseEvents}(\mathcal{D})$; // event parsing
foreach *identified event* $v_i \in \mathcal{V}$ **do**
  $(z_i, \mathcal{Z}) \leftarrow \text{Schema}(v_i, \mathcal{Z})$; // reusable event structure
  $g_i \leftarrow \text{GlobalCoord}(v_i)$;
  $\widehat{L}_i \leftarrow \text{ExtractSlots}(v_i, z_i)$; // global-local encoding
  $L_i \leftarrow \varnothing$;
  foreach *slot* $(k_{ij}, v_{ij}) \in \widehat{L}_i$ **do**
    $L_i \leftarrow L_i \cup \{(k_{ij}, v_{ij})\}$;
  $(s_i, a_i) \leftarrow \text{States}(v_i)$; // event and accessibility states
  $r_i \leftarrow \text{Residual}(v_i, z_i, L_i)$;
  $p_i \leftarrow \text{Link}(v_i, \mathcal{U})$; // details and source provenance
  $E_i \leftarrow (z_i, g_i, L_i, s_i, a_i, r_i, p_i)$;
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{E_i\}$;
$\widehat{\mathcal{R}} \leftarrow \text{Relate}(\mathcal{E}, \mathcal{U})$; // evidence-backed relations
$\mathcal{K} \leftarrow \{\text{temporal}, \text{update}, \text{conflict}, \text{slot}\}$;
foreach *relation type* $\ell \in \mathcal{K}$ **do**
  $\mathcal{R}_\ell \leftarrow \{e \in \widehat{\mathcal{R}} \mid \text{type}(e) = \ell\}$;
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_\ell$;
$\mathcal{I} \leftarrow \text{Index}(\mathcal{E})$; // retrieval index construction
$\mathcal{M} \leftarrow (\mathcal{E}, \mathcal{R}, \mathcal{Z}, \mathcal{I}, \mathcal{U})$;
return $\mathcal{M}$;

Algorithm notes. The repulsive and attractive gates return batched assessments; the inner loops apply threshold selection to these results. During construction and persistent updates, a source-supported partial revision preserves unchanged applicable slots and their source pointers in the re-

Algorithm 2: Online Dual-Stage Memory Reactivation

Input: Query q; memory system $\mathcal{M}$; candidate budget k; round count $R \geq 1$; evidence budget B; configuration $\Theta, \boldsymbol{\epsilon}$
Output: Answer y and transformed evidence state $\mathcal{T}_q$

$\mathcal{W} \leftarrow \varnothing$;
for $t = 1$ **to** R **do**
  $\mathcal{E}_q^- \leftarrow \varnothing$; $\mathcal{E}_q^+ \leftarrow \varnothing$; $\mathcal{C}^+ \leftarrow \varnothing$;
  $\mathcal{J} \leftarrow \varnothing$; $\mathcal{V}_q \leftarrow \varnothing$;
  $c = (c^{\text{sch}}, c^{\text{slot}}, c^{\text{tmp}}, c^{\text{state}}, c^{\text{evi}}) \leftarrow \text{Cues}(q, \mathcal{Z}, \mathcal{W})$;
  foreach $E_i \in \mathcal{E}$ **do**
    $\omega_i \leftarrow \lambda_1 \text{sim}(q, E_i) + \lambda_2 \text{match}(c^{\text{sch}}, z_i) + \lambda_3 \text{match}(c^{\text{slot}}, L_i)$
      $+ \lambda_4 \text{match}(c^{\text{tmp}}, (g_i, L_i)) + \lambda_5 \text{match}(c^{\text{state}}, s_i) + \lambda_6 \text{access}(a_i)$;
  $\mathcal{C}_{\text{seed}} \leftarrow \text{TopK}(\mathcal{E}, \{\omega_i\}, k)$;
  $\mathcal{C}_{\text{rel}} \leftarrow \text{Expand}(\mathcal{C}_{\text{seed}}, \mathcal{R}; \Theta)$; $\mathcal{C}_{\text{keep}} \leftarrow \text{Retain}(\mathcal{W})$;
  $\mathcal{C} \leftarrow \mathcal{C}_{\text{seed}} \cup \mathcal{C}_{\text{rel}} \cup \mathcal{C}_{\text{keep}}$; `// candidate awakening`
  $\mathcal{A}^- \leftarrow G_{\text{rep}}(q, c, \mathcal{C}, \mathcal{W})$; `// batched repulsive assessment`
  foreach $E_i \in \mathcal{C}$ **do**
    $(\rho_i, \tau_i^-) \leftarrow \text{Fields}(\mathcal{A}_i^-)$; $\mathcal{J} \leftarrow \mathcal{J} \cup \text{Requests}(\mathcal{A}_i^-)$;
    if $\rho_i > \epsilon_{\text{rep}}$ **then**
      $\mathcal{E}_q^- \leftarrow \mathcal{E}_q^- \cup \{(E_i, \tau_i^-, \rho_i, p_i)\}$;
    else
      $\mathcal{C}^+ \leftarrow \mathcal{C}^+ \cup \{E_i\}$;
  $\mathcal{A}^+ \leftarrow G_{\text{att}}(q, c, \mathcal{C}^+, \mathcal{E}_q^-, \mathcal{W})$; `// constraint-conditioned support`
  foreach $E_i \in \mathcal{C}^+$ **do**
    $(\sigma_i, \tau_i^+) \leftarrow \text{Fields}(\mathcal{A}_i^+)$; $\mathcal{J} \leftarrow \mathcal{J} \cup \text{Requests}(\mathcal{A}_i^+)$;
    if $\sigma_i > \epsilon_{\text{att}}$ **then**
      $\mathcal{E}_q^+ \leftarrow \mathcal{E}_q^+ \cup \{(E_i, \tau_i^+, \sigma_i, p_i)\}$;
  if $\mathcal{J} \neq \varnothing$ **then**
    $\mathcal{A} \leftarrow (\mathcal{A}^-, \mathcal{A}^+)$; `// conditional source grounding`
    $(\mathcal{E}_q^+, \mathcal{E}_q^-, \mathcal{V}_q) \leftarrow \text{Resolve}(q, c, \mathcal{C}, \mathcal{A}, \mathcal{J}, \mathcal{U}, \mathcal{W}; \boldsymbol{\epsilon})$;
  $(\mathcal{S}_q, \mathcal{N}_q, \mathcal{H}_q, \mathcal{P}_q, \mathcal{G}_q) \leftarrow \text{Transform}(q, \mathcal{E}_q^+, \mathcal{E}_q^-, \mathcal{V}_q; B)$;
  $\mathcal{T}_q \leftarrow (\mathcal{S}_q, \mathcal{N}_q, \mathcal{H}_q, \mathcal{P}_q, \mathcal{G}_q)$; `// budgeted evidence state`
  $\mathcal{W} \leftarrow \text{UpdateWork}(\mathcal{W}, c, \mathcal{C}, \mathcal{T}_q, \mathcal{V}_q)$;
$y \leftarrow \text{LLM}(q, \mathcal{T}_q)$;
return $(y, \mathcal{T}_q)$;

vised event record. Each selected event retains the qualified supporting claims or contrastive restrictions associated with its assessment. RESOLVE combines source readback and reconstruction with gate reassessment when recovered evidence changes the basis of a decision. The working state $\mathcal{W}$ carries query-local evidence, retained event records, unresolved requirements, and inspected sources across rounds. The budget B applies to the serialized answer-facing evidence state $\mathcal{T}_q$, including retained source excerpts.

A.6 ADDITIONAL METHOD DETAILS

Event identity and scoped revision. We make the scope and evidence requirements of reactivation explicit. Let η_i denote the underlying event instance described by record E_i, and let $\mathcal{U}_\pi$ contain source turns indexed by π. We write $\models_{\text{src}}$ for source-supported interpretation with speaker attribution and temporal qualifications preserved. For attribute κ, a relation indicating that E_j revises E_i requires:

$$(i,j) \in \mathcal{R}_{\text{upd}}^{(\kappa)} \implies \begin{aligned} &\eta_i = \eta_j, \\ &\mathcal{U}_{p_i \cup p_j} \models_{\text{src}} \text{revise}(E_j, E_i; \kappa). \end{aligned} \tag{11}$$

Sharing a schema does not establish event identity. Likewise, a later utterance timestamp alone does not establish a revision: statement time and the time at which an event state applies are distinct. Revision is attribute-specific: evidence changing one slot does not invalidate other attributes of the

event. Earlier values retain their historical scope when the sources establish a transition rather than a correction of the original observation.

Partial revisions and retained slots. For a source-supported partial revision of the same event, the revised record replaces the affected slots and carries forward unchanged slots that remain applicable, together with their source pointers and temporal qualifications. Each revised slot cites the observation establishing the change; the record-level pointer set includes the sources of both revised and retained slots. Prior versions remain available with their historical scope. Thus, an old record may constrain an obsolete value while the revised record supports both the updated value and unchanged attributes, without assigning two roles to the same record.

Sequential role assignment. At round t, let $\ell_{q,t}(i) \in \{-1, 0, +1\}$ denote the contrastive, silent, or supporting role of candidate E_i. The ordered gates induce:

$$\ell_{q,t}(i) = \begin{cases} -1, & \rho_i^{(t)} > \epsilon_{\mathrm{rep}}, \\ +1, & \rho_i^{(t)} \le \epsilon_{\mathrm{rep}} \ \wedge \ \sigma_i^{(t)}[\mathcal{E}_q^{-,(t)}] > \epsilon_{\mathrm{att}}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The bracketed argument emphasizes that the attractive score depends on the complete contrastive set; it is evaluated only when $\rho_i^{(t)} \le \epsilon_{\mathrm{rep}}$. These roles are query-local, so a historical record may support a historical question without changing its stored state or accessibility. Writing $I_{q,t}^-$, $I_{q,t}^+$, and $I_{q,t}^0$ for the corresponding event-ID sets gives:

$$\begin{aligned} I_{q,t}^- \dot\cup I_{q,t}^+ \dot\cup I_{q,t}^0 &= \mathrm{Ids}(\mathcal{C}_q^{(t)}), \\ \Pi_{q,t}^{\pm} &= \bigcup_{i \in I_{q,t}^{\pm}} p_i. \end{aligned} \tag{13}$$

Event roles are disjoint, whereas $\Pi_{q,t}^- \cap \Pi_{q,t}^+$ may be nonempty: distinct event records can cite the same source turn.

Scope-aware evidence construction. Represent a supporting item by (c, Ω, π), comprising a claim, its scope, and source pointers. Let $\mathcal{B}_t$ denote the available record evidence and inspected sources, and let $\mathcal{F}_q$ collect claim uses forbidden by the selected contrastive constraints. The required relation between support, provenance, and restrictions is:

$$\begin{aligned} &\mathcal{S}_q \subseteq \big\{(c, \Omega, \pi) \mid \varnothing \neq \pi \subseteq \Pi_{q,t}^+, \quad \mathcal{B}_t \models_{\mathrm{src}} c|_\Omega \text{ with provenance } \pi\big\}, \\ &\mathrm{Uses}(\mathcal{S}_q) \cap \mathcal{F}_q = \varnothing. \end{aligned} \tag{14}$$

Here, $\mathrm{Uses}(\mathcal{S}_q)$ denotes answer uses licensed by the qualified supporting items. The scope Ω identifies the event, attribute, and temporal qualification under which a claim applies. Selecting an event therefore does not license all of its contents as unqualified answer facts. Provenance π links an item to its source basis without implying that all referenced turns have been inspected for the current query. A constraint excludes a particular use of history; it neither proves the logical opposite nor supplies an unobserved replacement. For example, a withdrawn offer does not establish acceptance of another offer.

Source grounding and reassessment. Let $I_{\mathrm{read}}^{(t)}$ contain source IDs actually inspected by round t. A verified reconstructed detail may combine spans $\{(u, v_u)\}_{u \in \pi}$ from several source turns x_u. Its provenance must satisfy:

$$(c, \Omega, \{(u, v_u)\}_{u \in \pi}) \in \mathcal{V}_{q,\mathrm{ver}}^{(t)} \implies \begin{cases} \varnothing \neq \pi \subseteq I_{\mathrm{read}}^{(t)}, \\ v_u \preceq x_u, \quad \forall u \in \pi, \\ \{v_u\}_{u \in \pi} \models_{\mathrm{src}} c|_\Omega. \end{cases} \tag{15}$$

The relation $v_u \preceq x_u$ means that the quoted span occurs in the original turn; the spans' joint semantic support is checked separately. Missing sources or unsupported details remain unresolved. Readback may recover an omitted qualifier or expose a discrepancy, but a missing detail cannot be supplied merely because it fits the surrounding narrative.

When recovered evidence changes an assessment's basis, the affected gates reconsider the roles before transformation, including supporting decisions that depend on revised contrastive constraints.

Reassessment can remove an earlier restriction as well as introduce one. Consequently, accumulating inspected sources does not require monotonically accumulating selected roles: the current evidence state reflects the updated assessments and the dependencies between the two gates.

Budgeted evidence and unresolved requirements. The five evidence components share a single serialization budget:

$$\mathrm{Tok}(\mathrm{Serialize}(\mathcal{S}_q, \mathcal{N}_q, \mathcal{H}_q, \mathcal{P}_q, \mathcal{G}_q)) \leq B. \tag{16}$$

This includes retained source excerpts and structural overhead. Compression removes redundancy while preserving decisive qualifiers and provenance, such as whether a date was proposed or confirmed. Let $\mathcal{Q}_q$ be the specific answer requirements extracted from q, and let $\mathrm{Cov}(\mathcal{S}_q, \mathcal{H}_q)$ denote those resolved by qualified support and evidenced transitions. Remaining requirements must stay visible in the generation constraints:

$$\begin{aligned} \mathcal{Q}_q^{\mathrm{miss}} &= \mathcal{Q}_q \setminus \mathrm{Cov}(\mathcal{S}_q, \mathcal{H}_q), \\ \mathcal{Q}_q^{\mathrm{miss}} &\subseteq \mathrm{Gaps}(\mathcal{G}_q). \end{aligned} \tag{17}$$

Excluding an invalid answer does not by itself resolve the requested replacement. Reaching the round limit likewise does not establish complete coverage.

Evidence-backed persistent updates. Let $\mathcal{O}_n$ be the persistent edit operations supported at observation step n, which is distinct from the reactivation-round index t, and let $\pi(o)$ identify the sources justifying operation o. Such edits require new observational support:

$$\begin{aligned} o \in \mathcal{O}_n \implies & \pi(o) \cap \mathrm{Ids}(\Delta\mathcal{D}_n) \neq \varnothing, \\ & \mathcal{U}_{\pi(o)} \models_{\mathrm{src}} \mathrm{change}(o), \\ \Delta\mathcal{D}_n = \varnothing \implies & \mathcal{O}_n = \varnothing. \end{aligned} \tag{18}$$

Earlier sources may jointly justify an edit, but merely attaching a new source ID is insufficient: the new observation must support the proposed change. Query-local roles and generated answers do not independently verify persistent facts. A changed interpretation under a new query affects the query-local evidence state; an observed change in the underlying event can justify a persistent revision.

Accessibility assignment and revision. P1 initializes accessibility, and P8 revises it from new observations. Active denotes supported current facts or ongoing events; dormant denotes supported records without established current applicability, including explicitly paused tasks. Historical denotes previously valid versions superseded by an observed state change. Suppressed denotes records invalidated by source corrections or involved in unresolved contradictions within the same event and temporal scope. Archived denotes explicitly completed or closed events retained for later reference.

When conditions overlap, source-supported invalidation or conflict takes precedence, followed by supersession, closure, and explicit pause. Otherwise, currently applicable records are active and other supported records are dormant. Existing labels are retained unless new observations justify a change. Historical and archived records remain available for query-conditioned awakening, and query-local role assignments leave these persistent labels unchanged.

Accessibility in static QA. Accessibility is a persistent retrieval prior initialized from source-supported event status, rather than a query-specific support label. Static records default to dormant unless the sources establish a more specific status under the assignment rules above. During QA without new observations, persistent records, schema registries, and accessibility labels remain fixed; query cues, evidence roles, and the working evidence state may still change across questions and reactivation rounds. Transitions described within a document are encoded during construction. When all records share the same accessibility value, this score component is constant and does not affect their relative awakening order.

B PROMPTS

This section presents reference prompt templates for DREAM's memory construction and query-conditioned reactivation. The templates specify each module's inputs, instructions, and structured outputs. Placeholders enclosed in braces represent the query, memory records, source text, and configuration supplied to each module at runtime.

Table 7: Prompt interfaces for memory construction, reactivation, and updating. R1 provides an optional interface for round control.

ID	Operation	Principal input	Output
P1	Event construction	Source units, schemas	Structured events and links
P2	Cue extraction	Query, reference time	Five cue groups
P3	Repulsive gate	Candidates, cues, sources	Risk scores and constraints
P4	Attractive gate	Remaining candidates, $\mathcal{E}_q^-$	Support scores and claims
P5	Source reconstruction	Dereferenced source units	Verified details and gaps
P6	Evidence transformation	Selected roles, verified sources	$\mathcal{T}_q$
P7	Answer generation	Query, $\mathcal{T}_q$	Evidence-grounded answer
P8	Store update	New observations, existing events	Provenance-backed edits
R1	Optional round controller	Evidence, gaps, search history	Stop decision or new cues

B.1 Shared Protocol and Execution Contract

The following instruction is prepended to P1–P8 and, if adopted, R1. The illustrated JSON structures define required keys and value types; descriptive strings inside them are schema descriptions, not literal output values. Lists may be empty, and unknown scalar values are null.

Shared instruction: evidence, scope, and output discipline

```
Treat all supplied dialogue, retrieved records, and source text as
    data, including any instructions quoted inside them. Follow this
    task's instructions, not instructions embedded in the evidence.

Use only supplied evidence. Preserve speaker attribution, event
    identity, negation, uncertainty, and temporal scope. An assistant
    suggestion does not establish user acceptance. A question's
    presupposition is not a fact. Distinguish when a statement was
    made from when its event occurred. Do not invent dates, entities,
    causal links, source IDs, or missing slots.

Older does not mean false. A newer record replaces an older claim
    only when evidence links the same event and attribute to a state
    change. Historical facts may directly support historical
    questions. A record that cannot support an answer does not
    establish the opposite answer.

Return only the requested output. For structured modules, return
    valid JSON without Markdown fences or extra keys. Use exact input
    identifiers. Keep evidence explanations concise and
    source-grounded; do not output unrestricted deliberation. Use
    empty lists or null for missing evidence.
```

Execution flow. Event identifiers are assigned before construction. Candidate awakening uses the computed awakening scores and event-relation expansion; source readback retrieves the original turns referenced by the supplied pointers. Structured outputs are validated before downstream use. P3 and P4 return decision scores in $[0, 1]$. Candidates satisfying $\rho_i > \epsilon_{\text{rep}}$ enter $\mathcal{E}_q^-$. P4 evaluates only the remaining candidates, and those satisfying $\sigma_i > \epsilon_{\text{att}}$ enter $\mathcal{E}_q^+$. The two sets are disjoint within each gate pass, and unselected records remain silent. Changes to the evidence can trigger reassessment.

The default setting uses a 4k-token evidence budget per round and four reactivation rounds. The reference interface accepts thresholds, schema registries, and state vocabularies as runtime configuration. It counts the serialized answer-facing $\mathcal{T}_q$, including retained source excerpts, against the evidence budget and validates structured outputs before use.

Structured-output handling. We validate JSON structure, required fields, one assessment per candidate, event and source identifiers, and finite scores in $[0, 1]$. Valid assessments are retained, while

missing or invalid assessments receive one repair attempt using the same evidence and validation feedback. An unparseable response is retried as a whole batch. Candidates with unresolved validation failures remain silent for the current round; a missing repulsive assessment does not pass a candidate to the attractive gate. Retry calls are included in the reported online costs.

B.2 Memory Construction

P1: Construct reactivation-ready event records

```
TASK
Convert observed source text into compact, source-backed event
    records. For dialogue, source units are turns; for document QA,
    they are document spans. Group non-adjacent source units only
    when they describe the same identifiable event, relation, goal,
    preference, or decision. Shared topic alone is insufficient.

INPUT
Source units with IDs, source positions, and attribution/timestamps
    where available: {source_units}
Existing schema registry: {schema_registry}
Permitted state vocabulary: {state_vocabulary}
Available event IDs: {allocated_event_ids}

RULES
1. Choose an existing schema when appropriate; otherwise propose a
    concise schema label. Store shared structure in the schema and
    retain instance-specific exceptions and qualifications in
    residual details.
2. Keep source coordinates and statement times separate from event
    times in local slots. Cite sources for every slot and non-null
    state. Do not infer event chronology from source positions. Use
    null for unknown scalar times and empty lists for unknown time
    ranges.
3. Use source-supported labels from the permitted state vocabulary,
    distinguishing proposals, decisions, cancellations, and
    unresolved conflicts. For a source-supported partial revision of
    the same event, retain unchanged applicable slots with their
    source IDs and temporal scope; include these sources in p and
    preserve previous versions. If no event-state label is supported,
    set s.value to null and s.source_ids to [].
4. Assign a source-grounded accessibility label: active for
    established current applicability, dormant for other supported
    records, historical for superseded versions, suppressed for
    invalidated records or unresolved conflicts within the same event
    and temporal scope, and archived for explicitly completed or
    closed events. Static-document records default to dormant unless
    the sources justify another label. Record the reason in
    accessibility_basis; past tense or document position alone does
    not justify historical.
5. Propose temporal_continuation, state_update, contradiction, or
    shared_slot links only when the supplied evidence supports them.
6. Construction uses only the supplied source text, never evaluation
    questions, reference answers, or correctness feedback from a test
    set.

OUTPUT SHAPE
{
"events": [{
"event_id": "allocated ID",
"z": "schema identifier or proposed label",
"g": {"source_unit_ids": [], "statement_time_range": []},
"L": [{"slot": "attribute", "value": "observed content",
```

P1: Construct reactivation-ready event records (continued)

```
"event_time": null, "source_ids": []}],
"s": {"value": "permitted state", "source_ids": []},
"a": "accessibility value",
"accessibility_basis": "brief evidence-based explanation",
"r": [{"detail": "exception or qualification", "source_ids": []}],
"p": ["source unit ID"]
}],
"links": [{"from": "event ID", "to": "event ID",
"type": "permitted relation", "source_ids": []}]
}
```

B.3 Query Cues and Sequential Dual Gates

P2: Extract query-conditioned awakening cues

```
TASK
Describe what evidence would answer the query, without guessing its
    answer. Separate supplied anchors from the requested attribute.

INPUT
Query: {query}
Reference date, if available: {reference_date}
Available schema descriptions: {schema_registry}
Query-local working state from previous rounds, including unresolved
    requirements, retained evidence, and inspected source IDs (empty
    in the first round): {working_state}

RULES
Extract schema, slot, temporal, state, and evidence cues.
    Distinguish current, historical, transition, and unspecified
    temporal scope. Retain relative dates when no reference date
    supports normalization. Identify whether exact wording,
    chronology, or source readback may be needed. Do not fill the
    requested answer slot with a plausible answer or treat an assumed
    state in the question as an observed state.

OUTPUT SHAPE
{
  "schema_cues": [],
  "slot_cues": {"anchors": [], "requested_attributes": []},
  "temporal_cues": {"mode":
    "current|historical|transition|unspecified",
                    "target_time": null, "reference_date": null},
  "state_cues": [],
  "evidence_cues": {"required_details": [],
                    "readback_may_be_needed": false},
  "ambiguities": []
}
```

P3: Repulsive gate and contrastive constraint extraction

```
TASK
Assess the risk of using each candidate as positive evidence for
    this query. Explain any restriction with supplied sources or
    record fields.

INPUT
Query and cues: {query_and_cues}
Candidate events with state, accessibility, and pointers:
    {candidates}
```

P3: Repulsive gate and contrastive constraint extraction (continued)

```
Available linked comparison records and source text:
    {comparison_context}

RULES
1. Return one assessment per candidate. A high repulsive_score means
    stronger evidence that positive use would be inappropriate here.
2. Use risk_type: obsolete, conflicting, historical-only,
    event-mismatched, insufficiently-grounded, or none.
3. Check event identity, target attribute, time, speaker, and state.
    A supersession claim requires a supported same-event update. Do
    not reject an old event simply because a newer event exists.
4. A historical record matching a historical query is not
    historical-only risk merely because it is no longer current.
5. Write constraints as limits on evidence use, naming the affected
    attribute and temporal scope. A restriction on one slot does not
    invalidate other slots. Event mismatch or missing support does
    not prove a factual negation. Unresolved conflict must remain
    explicit; do not resolve it by guesswork.
6. If raw evidence is needed, request existing source pointers.
    Identify whether the basis is a compact record or supplied source
    text. Do not claim to have inspected source text that was not
    supplied.
7. Return the assessment without modifying the persistent store.
    Candidates whose repulsive scores exceed the configured threshold
    are assigned to the contrastive set.

OUTPUT SHAPE
{
  "assessments": [{
    "event_id": "candidate ID", "repulsive_score": 0.0,
    "risk_type": "one permitted type",
    "restricted_claim": null, "constraint": null,
    "basis": "source_text|event_record|missing_evidence",
    "source_ids": [], "related_event_ids": [],
    "readback_requests": [], "evidence_note": "brief grounded basis"
  }]
}
```

P4: Attractive gate conditioned on contrastive evidence

```
TASK
Assess which remaining events can positively support the requested
    answer, subject to the selected contrastive constraints.

INPUT
Query and cues: {query_and_cues}
Candidates remaining after the repulsive threshold:
    {remaining_events}
Selected contrastive events and constraints: {negative_evidence}
Available source text: {available_sources}

RULES
1. Return one assessment per remaining event. A high
    attractive_score means stronger supported usefulness for this
    query.
2. Use support_type: direct_answer, current_user_state, preference,
    causal_explanation, source_record, or none. A historical answer
    can be direct_answer; use current_user_state only for
    current-state facts.
```

P4: Attractive gate conditioned on contrastive evidence (continued)

```
3. Respect target time, event identity, and the supplied
    constraints. Do not reintroduce IDs selected as contrastive in
    this gate pass.
4. Cite sources for each supported claim and preserve
    qualifications. For retained slots in a revised record, cite
    their source IDs, not only the update turn. A partial status
    statement can be useful without filling the answer slot. An
    option is not a decision; correlation is not causation.
5. If the needed detail is absent, name the missing detail and
    request available source pointers. Do not complete a compressed
    record by inference or treat an uninspected pointer as verified
    evidence.
6. Remaining candidates whose attractive scores exceed the
    configured threshold enter the supporting set; other remaining
    candidates stay silent.

OUTPUT SHAPE
{
  "assessments": [{
    "event_id": "remaining candidate ID", "attractive_score": 0.0,
    "support_type": "one permitted type",
    "supported_claims": [{"claim": "qualified fact", "source_ids":
    []}],
    "basis": "source_text|event_record|missing_evidence",
    "answer_coverage": "complete|partial|none",
    "missing_details": [], "readback_requests": []
  }]
}
```

B.4 Source Reconstruction and Evidence Transformation

P5 reconstructs details from the source units retrieved through source pointers, including dialogue turns and document spans. When readback changes the basis for a gate decision, the affected records are reassessed before their roles are finalized.

P5: Reconstruct query-relevant details from source units

```
TASK
Recover exact evidence needed for the query from dereferenced
    sources.

INPUT
Query and required details: {query_and_required_details}
Event records and their existing pointers: {events_and_pointers}
Dereferenced source units with exact IDs, source positions, and
    attribution/timestamps where available: {source_units}

RULES
Read only the supplied source units. Extract minimal exact spans
    supporting each fact and preserve source attribution where
    available. Leave missing speaker information null. Separate
    availability, suggestions, requests, confirmations, and
    cancellations when present. Preserve source-supported chronology;
    do not infer event times from statement timestamps or document
    positions alone. Report discrepancies between compact records and
    source text instead of silently repairing them. If a pointer
    failed or no source establishes the detail, report the gap. Do
    not assign a new gate role or edit persistent state in this
    module.
```

P5: Reconstruct query-relevant details from source units (continued)

```
OUTPUT SHAPE
{
  "verified_details": [{"event_ids": [], "claim": "qualified fact",
    "source_id": "supplied ID", "exact_span": "verbatim source span",
    "speaker": "source speaker", "temporal_scope": null}],
  "record_source_discrepancies": [],
  "unresolved_details": [],
  "unavailable_source_ids": []
}
```

P6: Transform selected memories into answer-facing evidence

```
TASK
Construct T_q = (S_q, N_q, H_q, P_q, G_q) for this query. Transform
    the evidence into usable support and constraints, rather than
    concatenating selected records with role labels.

INPUT
Query and cues: {query_and_cues}
Final supporting events and claims: {positive_evidence}
Final contrastive events and constraints: {negative_evidence}
Verified readback details, if available: {reconstructed_evidence}
Evidence token budget for this round: {evidence_budget}

RULES
1. S_q contains only selected, source-backed supporting claims.
    Retained slots in a supporting revised record keep their source
    IDs; each supported claim cites the record through which it was
    selected and its own source evidence.
2. N_q states precisely which claim must not be used in which scope
    and why. Do not turn a restriction into an unsupported opposite
    fact.
3. H_q includes only evidenced temporal or state transitions;
    distinguish the same event's revision from an unrelated event's
    later occurrence.
4. P_q maps support and constraints to source IDs. Keep any
    necessary exact excerpts here, preserving text and attribution.
5. G_q specifies target time, answer requirements, forbidden
    unsupported inferences, unresolved gaps, and qualifications
    needed for generation.
6. Deduplicate and compress within the budget while preserving
    decisive support, operative constraints, and their provenance.
    Never drop a qualifier in a way that changes a claim. If evidence
    remains incomplete, record the gap in G_q instead of fabricating
    a complete answer.

OUTPUT SHAPE
{
  "S_q": [{"claim": "supported fact", "event_ids": [], "source_ids":
    []}],
  "N_q": [{"restricted_claim": "claim", "scope": "where restricted",
           "reason": "evidenced limitation", "source_ids": []}],
  "H_q": [{"from_event": "ID", "to_event": "ID",
           "relation": "supported transition", "source_ids": []}],
  "P_q": [{"source_id": "ID", "event_ids": [],
           "role": "support|constraint|both", "exact_excerpt":
    null}],
  "G_q": {"target_scope": "query scope", "answer_requirements": [],
          "avoid": [], "unresolved_gaps": [], "qualifications": []}
}
```

B.5 Answer Generation and Persistent Updates

P7: Answer from transformed evidence

```
TASK
Answer the user's question using the transformed evidence state.

INPUT
Question: {query}
Transformed evidence T_q: {transformed_evidence}
Task-specific answer format: {answer_format}

RULES
Use S_q as positive evidence, N_q as restrictions on evidence use,
    H_q for supported state changes, P_q for provenance, and G_q for
    scope and qualifications. Do not use contrastive records as
    unqualified support. For a current-state question, use the
    supported current state; for a historical question, answer for
    the requested time rather than defaulting to the newest state. Do
    not invent a replacement when an old fact is excluded. If
    evidence is insufficient or conflicting, state the supported part
    and the specific uncertainty. Do not infer the answer from the
    question's presupposition or unsupported background knowledge.

Return a concise answer in the requested format. Include source IDs
    only if that format requests them. Do not expose gate scores,
    internal records, or reasoning traces in the answer. No reference
    answer is supplied.
```

P8: Update memory using new observations

```
TASK
Propose persistent memory updates justified by newly observed
    dialogue or feedback. Preserve the provenance of both old and
    revised states.

INPUT
Existing relevant events and schemas: {existing_events_and_schemas}
New observations with source IDs: {new_observations}
Previous query, transformed evidence, and generated answer:
    {run_context}
Permitted states and available new event IDs: {update_configuration}

RULES
1. Add a new event or revise an existing event only when the
    observation establishes the relevant fact or change. Shared topic
    is not identity.
2. Keep prior claims and source pointers. For a supported partial
    revision of the same event, replace only the affected slots and
    carry forward unchanged applicable slots with their source IDs
    and temporal scope. Include these sources in the revised record's
    p. Link the versions and preserve unresolved contradictions with
    attribution.
3. Propose state and accessibility changes separately. A query-local
    contrastive or supporting role alone does not justify a
    persistent edit.
4. Generated answers and transformed evidence may help locate
    records but are not independent verification of their own factual
    content. An assistant proposal may be recorded as a proposal, not
    user acceptance.
```

P8: Update memory using new observations (continued)

```
5. Use only permitted accessibility values: active, dormant,
    historical, suppressed, archived. Do not delete historical
    evidence by default.
6. If no new observation warrants an edit, return an empty
    operations list.

OUTPUT SHAPE
{
  "operations": [{
    "operation":
    "add_event|revise_state|revise_accessibility|add_link",
    "target_event_id": "existing or allocated ID",
    "payload": {}, "source_ids": [],
    "reason": "brief observation-backed explanation"
  }],
  "unresolved_conflicts": []
}
Payload: P1 event for add_event; old_value/new_value for revisions;
    P1 link for add_link. Proposed edits are checked for valid
    structure, identifiers, and supporting sources before being
    committed to the memory store.
```

B.6 Optional Interface for Iterative Reactivation

R1 is an optional reference template for iterative reactivation. It identifies unresolved evidence requirements and proposes cue refinement or source readback within the remaining round budget.

R1: Check evidence sufficiency and propose a bounded next step

```
TASK
Determine whether the current evidence resolves the question and, if
    needed, identify a specific additional evidence request.

INPUT
Query and current transformed evidence: {query_and_evidence}
Previously tried cues and inspected source IDs: {search_history}
Remaining rounds and available pointers: {remaining_resources}

RULES
Check answer coverage, temporal and event alignment, unresolved
    conflicts, and missing exact details. Repeated retrieval of the
    same evidence does not increase certainty. Suggest new cues only
    to address named gaps, or request an existing unread pointer.
    Stop if evidence is sufficient, the round budget is exhausted, or
    no concrete new request is available. Stopping does not imply
    answerability; retain unresolved gaps for P7. Keep additional
    evidence requests within the remaining round limit, and keep the
    transformed evidence within the per-round token budget.

OUTPUT SHAPE
{
  "evidence_status": "sufficient|partial|unresolved",
  "action": "answer|refine_cues|read_sources",
  "missing_requirements": [],
  "next_cues": [], "source_ids_to_read": [],
  "stop_reason":
    "sufficient|budget_exhausted|no_new_request|continue"
}
```

Table 8: Role-assignment policies for the controlled replay.

Policy	Role calls / round	First pass	Second pass
Joint-1	1	Jointly assign support, constraint, or silence.	–
Joint-2	2	Same joint assignment.	Reassess all candidates; freely revise any role or claim.
$A \rightarrow R$ + repair	2	Select candidate support from the full pool.	Check applicability; form constraints and revise support, including additions or replacements from the full pool.
DREAM $R \rightarrow A$	2	Apply Prompt P3 to form the contrastive set.	Apply Prompt P4 to the remaining candidates, conditioned on that set.

B.7 Comparison Interface for Reactivation Order

The comparison reuses the shared instruction and the claim, constraint, source, and score fields of P3/P4. Joint policies decide the role directly; the reversed policy can repair support in its second pass. DREAM retains P3/P4 and their threshold rules.

C1: Joint or reversed role assignment

```
INPUT
Query and cues; full candidate pool; fixed source turns;
policy and pass number; previous assessments (if any).

TASK
Joint-1 / Joint-2, pass 1: jointly choose each event's role.
Joint-2, pass 2: reassess the full pool and freely revise
the previous roles, claims, and constraints.
A->R+repair, pass 1: select support from the full pool.
Return only support or silent in this first pass.
A->R+repair, pass 2: check applicability, form constraints,
and revise support; add or replace support from any
candidate in the full pool when the evidence warrants it.

OUTPUT
Return {"assessments": [...]} with one item per event_id.
Each item has role: support|constraint|silent, plus the
corresponding P4 support fields or P3 constraint fields.
Silent items contain event_id and role only. Preserve
qualified claims and source IDs. Use only supplied sources.
Assign one role per event. Do not request further readback.
```

C Additional Results

C.1 Controlled Comparison of Reactivation

We compare the policies in Table 8 to test whether forming constraints before selecting support improves evidence use beyond a second decision pass. Joint-2 and the reversed policy both allow the second pass to revise the initial choices. Using Qwen3-30B-Instruct, we evaluate each policy over four reactivation rounds with shared candidate evidence and policy-specific working states.

Shared inputs and budget. For each query, we run the common cue extraction and awakening procedure once from an empty working state. All policies receive the same event bank, candidate IDs, content and ordering, and source turns dereferenced from their pointers. This candidate pool and its source evidence remain fixed across all four rounds, with no additional awakening or adaptive readback. Each policy starts from an empty working state and carries its own updated state into the next round; candidate roles are reassessed in every round. P6 is applied after each round using the same prompt and a 4,096-token serialized evidence budget, and P7 generates the answer only after the fourth round. All role-assignment calls use an 8,192-token generation limit. Joint-2, the reversed policy, and DREAM each use two role-assignment calls per round; Joint-1 uses one. We report actual total online input/output tokens, calls, and end-to-end time, including shared preparation, all

Table 9: Values for four-round fixed-input replay on LoCoMo with Qwen3-30B-Instruct; the main-text evaluation uses four reactivation rounds. In this scenario, standard-category F1 and mean per-query costs use the original 1,386 test questions; ISA and AM use the pre-annotated risk subset. Risk and control F1 reuse these replay outputs. Constructed historical questions are evaluated separately and excluded from Overall and the cost averages. Tokens include all online LLM inputs and outputs. Subset sizes: risk n =120, control n =120, historical n =60. Brackets give 95% percentile intervals from 10,000 paired bootstrap resamples of the 10 source conversations in LoCoMo.

Policy	QA F1 (%) ↑					Diagnostics (%) ↓		Online cost		
	Single-Hop	Multi-Hop	Temporal	Open Domain	**Overall**	ISA	AM	Tokens (k)	Calls	Time (s)
Joint-1	55.85	41.02	48.36	37.20	50.41	20.00	15.00	28.72	10.00	19.80
Joint-2	56.67	43.08	50.74	37.72	51.77	15.00	10.83	45.20	14.00	29.60
$A \to R$ + repair	56.53	42.76	49.86	37.83	51.46	16.25	12.50	44.86	14.00	29.90
DREAM $R \to A$	57.80	45.90	53.60	37.70	53.50	7.50	6.67	45.04	14.00	29.72
▼ *Minus comparison; brackets: 95% bootstrap intervals*										
vs. Joint-2	+1.13	+2.82	+2.86	-0.02	+1.73 [0.83, 2.57]	-7.50 [-10.83, -4.17]	-4.17 [-6.67, -1.67]	-0.16	0.00	+0.12
vs. $A \to R$ + repair	+1.27	+3.14	+3.74	-0.13	+2.04 [1.15, 2.88]	-8.75 [-15.28, -2.50]	-5.83 [-10.28, -0.83]	+0.18	0.00	-0.18

▼ *F1 on diagnostic subsets*			
Policy	Risk	Control	Historical
Joint-1	41.50	56.60	48.80
Joint-2	45.80	57.20	50.60
$A \to R$ + repair	44.70	57.24	50.15
DREAM $R \to A$	50.60	57.16	50.82

four rounds, and final answer generation. This diagnostic evaluates iterative role assignment under fixed evidence.

Queries and measurements. Before running the policies, we annotate held-out LoCoMo queries and their fixed candidate pools for obsolete states, event mismatch, and proposal/decision confusion, and retain a control stratum without these risks. Where the history supports both times, we construct paired historical questions and source-grounded reference answers before policy execution. Annotation uses the sources and is blind to policy outputs. A claim is identified by its event, attribute, value, and temporal scope. Applicability is labeled separately for each query; duplicate copies within its candidate pool count once. Two annotators independently label query-specific applicability and review the historical questions and reference answers using the sources. Disagreements are resolved through joint review, and answer misuse is assessed under the same procedure.

Let $\mathcal{Q}_{\text{risk}}$ be the fixed, pre-annotated risk subset and $\mathcal{I}_q$ the set of claims in the shared candidate pool that are inapplicable to q. Let $\widehat{\mathcal{S}}^{\pi}_{q,4}$ denote the claims selected as positive support by policy π after its fourth-round role assignment and before the final application of P6. Selected support is mapped to annotated claims using event/source identifiers and content, with duplicates counted once per query. Inapplicable-support activation is:

$$\text{ISA}(\pi) = 100\,\frac{\sum_{q\in\mathcal{Q}_{\text{risk}}}\left|\mathcal{I}_q \cap \widehat{\mathcal{S}}^{\pi}_{q,4}\right|}{\sum_{q\in\mathcal{Q}_{\text{risk}}}|\mathcal{I}_q|}. \tag{19}$$

For the final answer y^{π}_q generated after round four, let $U(y^{\pi}_q, c, q) = 1$ when the answer uses claim c as valid positive evidence for q, and zero otherwise. Answer misuse is:

$$\text{AM}(\pi) = \frac{100}{|\mathcal{Q}_{\text{risk}}|}\sum_{q\in\mathcal{Q}_{\text{risk}}}\mathbf{1}\left[\exists c \in \mathcal{I}_q : U(y^{\pi}_q, c, q) = 1\right]. \tag{20}$$

Historical comparisons, explicit rejection of an old state, and correctly scoped historical facts do not count as misuse. Each risk query contributes at most once to AM; abstentions remain in both denominators. We average metrics across runs and report paired differences for Overall F1, ISA, and AM with 95% bootstrap intervals clustered by source conversation. Table 9 separates standard-category performance from diagnostic-subset F1; mean online costs cover the original 1,386 questions only.

Table 10: Fine-grained ablations of the reactivation-ready event representation. Results report mean F1 (%) on LoCoMo and mean ACC (%) on LongMemEval over three runs using Qwen3-30B-Instruct.

Method	LoCoMo					LongMemEval				
	S-Hop	M-Hop	Temp	Open	All	Temp	M-Session	Kno	S-User	All
DREAM	58.12	45.36	53.47	37.91	53.56	76.67	61.11	86.19	91.01	74.37
w/o Global Coordinates (g_i)	56.21	40.13	49.52	35.62	50.59	75.28	56.94	79.52	88.89	70.22
w/o Local Slots (L_i)	55.84	39.21	50.38	33.92	50.29	76.94	55.83	79.05	88.36	70.67
w/o Schema (z_i)	56.93	42.74	51.76	35.83	51.94	78.61	60.56	80.95	89.95	72.30
w/o Residuals (r_i)	55.37	43.19	51.94	36.26	51.24	78.89	60.83	79.52	88.36	72.07

Table 11: Separating event validity from memory accessibility. The w/o State setting removes both state and accessibility information, following the main-text ablation. Results report mean F1 (%) on LoCoMo and mean ACC (%) on LongMemEval over three runs using Qwen3-30B-Instruct.

Method	LoCoMo					LongMemEval				
	S-Hop	M-Hop	Temp	Open	All	Temp	M-Session	Kno	S-User	All
DREAM	58.12	45.36	53.47	37.91	53.56	76.67	61.11	86.19	91.01	74.37
w/o Event State (s_i)	54.72	42.06	45.61	35.08	49.28	71.67	58.33	72.86	85.71	67.56
w/o Accessibility (a_i)	55.83	43.47	48.76	36.12	50.87	75.83	60.28	77.62	88.36	70.89
w/o State	52.88	40.15	43.72	33.64	47.44	68.33	55.56	70.95	82.54	64.96

Results. Table 9 reports Overall F1 gains of 1.73 and 2.04 points for DREAM over Joint-2 and the reversed policy, respectively, with matched call counts and comparable online token costs. Both paired 95% confidence intervals lie above zero. The point-estimate gains are larger on the risk subset, at 4.80 and 5.90 points, while control and historical F1 remain close to those of both comparators. ISA and AM are also lower for DREAM; their paired confidence intervals exclude zero against the reversed policy and Joint-2. These results support the value of constraint-first role assignment in this four-round fixed-input evaluation, with diagnostic evidence of reduced inapplicable-support activation and answer misuse relative to the reversed policy.

C.2 Additional Ablations

We examine the individual design choices underlying DREAM. Unless otherwise specified, these experiments use Qwen3-30B-Instruct. LoCoMo overall F1 uses the fixed question-count weights, and each ablation is compared with the full model under the same ingestion setting.

Event Representation.

Table 10 separates the contributions of global coordinates, local slots, schemas, and residual details. Removing global coordinates and local slots lowers overall F1 from 53.56 to 50.59 and 50.29, respectively. Local-slot removal particularly affects multi-hop questions, reducing F1 from 45.36 to 39.21. Removing schemas or residuals also degrades overall F1, to 51.94 and 51.24. The category breakdown distinguishes their roles: schemas assist structural matching, whereas residuals preserve details that may be needed for specific factual queries. These results support retaining both event structure and instance-specific information in the memory representation.

Event State and Accessibility. Table 11 distinguishes event validity from memory accessibility. Removing event state reduces overall F1 to 49.28, compared with 50.87 when accessibility is removed. Removing both fields gives the lowest score, 47.44. The effect is especially clear on temporal questions, where F1 falls from 53.47 to 45.61 without event state and to 43.72 without both fields. The LongMemEval knowledge-update column shows the same ordering. These results suggest that validity information is central to deciding whether a recalled claim applies to the query, while accessibility provides additional control over which records are reactivated.

Structured Cue Matching. Table 12 tests candidate selection while retaining the downstream gates. Semantic-only awakening lowers overall F1 from 53.56 to 49.33, with losses of 5.08 points on multi-hop questions and 6.11 points on temporal questions. Downstream role assignment therefore does not fully compensate for weaker initial matching. The individual removals show more targeted effects: omitting slot matching lowers multi-hop F1 to 41.37, while omitting temporal matching low-

Table 12: Ablations of structured cue matching during candidate awakening. Semantic-only Awakening ranks initial events by semantic similarity; all subsequent gates remain unchanged. Results report mean F1 (%) on LoCoMo and mean ACC (%) on LongMemEval over three runs using Qwen3-30B-Instruct.

Method	LoCoMo					LongMemEval				
	S-Hop	M-Hop	Temp	Open	All	Temp	M-Session	Kno	S-User	All
DREAM	58.12	45.36	53.47	37.91	53.56	76.67	61.11	86.19	91.01	74.37
Semantic-only Awakening	54.87	40.28	47.36	33.95	49.33	73.06	55.56	76.19	85.71	68.44
w/o Schema Matching	57.31	43.82	52.18	36.42	52.47	79.17	61.11	80.95	89.95	72.96
w/o Slot Matching	56.48	41.37	51.26	34.63	51.27	78.33	57.78	80.00	88.36	71.41
w/o Temporal Matching	57.12	43.71	48.94	36.53	51.68	73.61	60.28	78.57	89.95	70.52
w/o State Matching	57.24	43.95	49.83	36.74	51.99	75.28	60.83	77.14	88.36	70.89

Table 13: Effect of relation-based event expansion. Score-only Candidate Expansion fills the same candidate slots with the next highest awakening scores instead of linked events. Results report mean F1 (%) on LoCoMo and mean ACC (%) on LongMemEval over three runs using Qwen3-30B-Instruct.

Method	LoCoMo					LongMemEval				
	S-Hop	M-Hop	Temp	Open	All	Temp	M-Session	Kno	S-User	All
DREAM	58.12	45.36	53.47	37.91	53.56	76.67	61.11	86.19	91.01	74.37
Score-only Candidate Expansion	55.86	40.63	49.07	35.24	50.37	74.44	55.56	77.62	87.30	69.41
w/o Temporal Continuation	57.46	43.16	50.24	37.21	52.08	75.83	60.28	80.95	89.95	72.00
w/o State-Update Links	57.28	43.72	49.62	36.83	51.93	74.44	60.83	76.19	88.36	70.96
w/o Contradiction Links	57.53	44.18	51.03	37.04	52.46	77.50	61.67	78.57	89.95	72.37
w/o Shared-Slot Links	56.62	41.28	52.31	35.46	51.60	79.17	57.22	80.95	88.36	71.85

Table 14: State maintenance under incremental dialogue ingestion using Qwen3-30B-Instruct. All variants ingest identical sessions in chronological order and answer the same held-out questions after the final session. Results report mean F1 (%) on LoCoMo and mean ACC (%) on LongMemEval over three runs.

Method	LoCoMo					LongMemEval				
	S-Hop	M-Hop	Temp	Open	All	Temp	M-Session	Kno	S-User	All
DREAM	57.94	45.12	53.18	37.74	53.34	80.56	62.78	81.90	91.01	74.07
New Events without Revising Existing States	54.76	41.62	45.31	34.86	49.15	70.28	56.39	71.43	84.13	66.52
w/o Event-State Revision	55.18	42.14	46.83	35.17	49.81	72.22	57.78	73.81	85.71	68.00
w/o Accessibility Revision	56.37	43.48	49.61	36.28	51.35	76.67	60.28	78.57	88.36	71.19

ers temporal F1 to 48.94. The corresponding LongMemEval columns exhibit similar task-specific changes. Structured cues help identify suitable evidence before the gates determine its role in generation.

Relation-Based Event Expansion. Table 13 replaces linked-event expansion with the next highest awakening scores while preserving the number of candidate slots. This score-only variant lowers overall F1 from 53.56 to 50.37, with multi-hop and temporal scores decreasing to 40.63 and 49.07. The matched candidate count supports the value of event relations beyond simply retrieving more records. Removing shared-slot links particularly affects multi-hop reasoning, while removing state-update links degrades temporal and knowledge-update performance. Temporal-continuation and contradiction links also contribute. Together, these results support expanding toward related records that explain, revise, or constrain an initially matched event.

Incremental State Maintenance. Table 14 evaluates chronological session ingestion, with questions answered after the final session. The full model reaches 53.34 overall F1 in this setting. Adding new events without revising existing states reduces it to 49.15; disabling event-state revision or accessibility revision yields 49.81 and 51.35, respectively. Temporal F1 decreases from 53.18 to 45.31 when existing states are left unchanged. LongMemEval also shows larger losses on state-sensitive categories when event-state revision is disabled. These findings support maintaining existing records as new evidence arrives, so that accumulated memories retain their historical content without obscuring later changes.

Consistency across Backbones. Table 15 repeats the six main ablations with GPT-4o-mini and Qwen3-8B-Instruct. Removing state and accessibility produces the largest overall F1 losses for both models, at 6.03 and 6.50 points, respectively. Their temporal F1 drops by 9.56 and 10.22 points.

Table 15: Cross-backbone ablations of all six components. Each cell shows absolute F1/ACC (%) with Δ from the corresponding full model.

Method	LoCoMo					LongMemEval				
	S-Hop	M-Hop	Temp	Open	All	Temp	M-Session	Kno	S-User	All
▼ *GPT-4o-mini*										
DREAM	**56.84**	**43.21**	**51.92**	**30.94**	**51.71**	**50.28**	**58.89**	**81.43**	**88.89**	**67.26**
w/o Rep. Gate	51.72 (-5.12)	37.83 (-5.38)	44.19 (-7.73)	27.38 (-3.56)	46.09 (-5.62)	42.22 (-8.06)	51.11 (-7.78)	72.38 (-9.05)	83.07 (-5.82)	60.59 (-6.67)
w/o Att. Gate	52.41 (-4.43)	36.29 (-6.92)	46.73 (-5.19)	27.84 (-3.10)	46.75 (-4.96)	45.28 (-5.00)	48.89 (-10.00)	75.24 (-6.19)	81.48 (-7.41)	61.70 (-5.56)
w/o Constr.	53.16 (-3.68)	39.04 (-4.17)	44.82 (-7.10)	26.21 (-4.73)	47.16 (-4.55)	39.72 (-10.56)	50.28 (-8.61)	71.43 (-10.00)	84.13 (-4.76)	59.11 (-8.15)
w/o State	51.48 (-5.36)	38.17 (-5.04)	42.36 **(-9.56)**	27.91 (-3.03)	45.68 **(-6.03)**	37.78 **(-12.50)**	49.72 (-9.17)	69.05 **(-12.38)**	79.89 **(-8.99)**	57.63 **(-9.63)**
w/o G-L Enc.	52.63 (-4.21)	35.08 **(-8.13)**	47.49 (-4.43)	25.76 **(-5.18)**	46.67 (-5.04)	44.44 (-5.83)	46.67 **(-12.22)**	74.76 (-6.67)	84.13 (-4.76)	60.00 (-7.26)
w/o Pointers	51.19 **(-5.65)**	38.42 (-4.79)	47.68 (-4.24)	28.14 (-2.80)	46.69 (-5.02)	45.28 (-5.00)	52.22 (-6.67)	72.86 (-8.57)	81.48 (-7.41)	61.78 (-5.48)
▼ *Qwen3-8B-Instruct*										
DREAM	**54.73**	**42.18**	**50.36**	**33.82**	**50.22**	**75.83**	**58.61**	**79.05**	**88.36**	**69.48**
w/o Rep. Gate	49.28 (-5.45)	36.34 (-5.84)	42.24 (-8.12)	28.53 (-5.29)	44.15 (-6.07)	66.11 (-9.72)	52.50 (-6.11)	70.95 (-8.10)	81.48 (-6.88)	61.78 (-7.70)
w/o Att. Gate	50.07 (-4.66)	34.86 (-7.32)	44.92 (-5.44)	28.94 (-4.88)	44.90 (-5.32)	70.56 (-5.28)	48.89 (-9.72)	73.81 (-5.24)	79.89 (-8.47)	62.59 (-6.89)
w/o Constr.	50.84 (-3.89)	37.63 (-4.55)	42.88 (-7.48)	27.46 (-6.36)	45.31 (-4.91)	64.72 (-11.11)	51.11 (-7.50)	69.05 (-10.00)	83.07 (-5.29)	60.22 (-9.26)
w/o State	49.11 (-5.62)	36.62 (-5.56)	40.14 **(-10.22)**	29.21 (-4.61)	43.72 **(-6.50)**	62.50 **(-13.33)**	50.28 (-8.33)	67.62 **(-11.43)**	78.31 **(-10.05)**	58.81 **(-10.67)**
w/o G-L Enc.	50.24 (-4.49)	33.51 **(-8.67)**	45.63 (-4.73)	26.49 **(-7.33)**	44.74 (-5.48)	69.17 (-6.67)	47.50 **(-11.11)**	72.38 (-6.67)	83.07 (-5.29)	60.96 (-8.52)
w/o Pointers	48.83 **(-5.90)**	37.19 (-4.99)	45.92 (-4.44)	30.51 (-3.31)	44.95 (-5.27)	71.39 (-4.44)	53.33 (-5.28)	70.95 (-8.10)	79.89 (-8.47)	62.96 (-6.52)

Table 16: The first row counts supported assertions out of all audited slot/state assertions; the second counts incorrectly merged records out of audited records with multiple source spans. Sampled event records: n =100.

Construction check	**Audited units**	**Count**	**Rate (%)**
Source-supported slot/state assertions ↑	400	392	98.00
Incorrect cross-event merges ↓	40	3	7.50

Removing global-local encoding most strongly affects multi-hop F1, with decreases of 8.13 and 8.67 points. Both gates and the explicit contrastive constraints also contribute on each backbone. These repeated effects support the roles of state tracking, event structure, and controlled evidence use across answer models of different capacities.

C.3 Event Construction Quality

We audit a uniform sample of event records from the constructed LoCoMo memories, with the sampled IDs fixed before inspection. For each sampled record, annotators inspect its cited turns in their session context and check every slot and state assertion for source support, preserving speaker, negation, and time. The source-support rate is the fraction of these assertions supported by the cited evidence. Among sampled records combining at least two distinct source spans, the incorrect-merge rate is the fraction that conflate different underlying events. Table 16 reports both numerators and denominators. The audit uses independent review by the same two annotators, with disagreements resolved jointly for labeling and adjudication.

C.4 Reactivation Statistics

Tables 17 and 18 summarize execution by conversation and question type. The held-out set contains 1,386 questions: 757 single-hop, 254 multi-hop, 289 temporal, and 86 open-domain questions. Every reported group executes four rounds. The observed differences therefore concern the events processed and their assigned roles within the same round budget.

The category breakdown shows distinct evidence requirements. Multi-hop questions evaluate the most distinct events across rounds, averaging 59.36 per query, retain the largest final supporting set at 7.95 events, and take the longest mean online time at 18.20 seconds. Temporal questions retain the largest contrastive set, averaging 4.98 events, compared with 1.92 for single-hop questions. This pattern is consistent with broader evidence aggregation for multi-hop reasoning and a greater need to distinguish applicable event states for temporal reasoning. The final supporting and contrastive sets are substantially smaller than the candidate sets, showing that candidate awakening and answer-facing evidence selection serve different roles.

Table 17: Per-conversation reactivation statistics on the held-out LoCoMo questions using Qwen3-30B-Instruct. C_1–C_{10} are display labels mapped to the official conversation IDs; #Q is the number of evaluated questions. Time and Rounds are the mean end-to-end online seconds and executed reactivation rounds per query. Awakened counts distinct events evaluated across all rounds, whereas Supporting and Contrastive count the final $|\mathcal{E}_q^+|$ and $|\mathcal{E}_q^-|$ before evidence serialization. Readback is the percentage of queries that dereference at least one source pointer. All means include unsuccessful queries.

Conversation	#Q	DREAM Reactivation Statistics					
		Time (s)	Rounds	Awakened	Supporting	Contrastive	Readback (%)
C_1	132	14.99	4.00	47.13	5.70	2.75	25.15
C_2	144	14.83	4.00	46.79	5.49	2.85	35.57
C_3	126	14.86	4.00	46.65	5.51	2.96	19.20
C_4	150	14.77	4.00	46.48	5.71	2.89	32.15
C_5	138	15.05	4.00	47.84	5.77	2.89	28.41
C_6	141	14.18	4.00	45.13	5.30	2.81	24.97
C_7	135	14.26	4.00	46.34	5.46	2.83	26.82
C_8	147	14.37	4.00	46.05	5.35	2.89	23.96
C_9	129	14.31	4.00	46.52	5.53	2.95	27.28
C_{10}	144	14.61	4.00	46.89	5.63	2.92	25.15
Overall	1386	14.62	4.00	46.57	5.54	2.87	26.99

Table 18: Reactivation statistics by question category on the held-out LoCoMo questions using Qwen3-30B-Instruct. #Q is the evaluated category size. Time and Rounds are the mean end-to-end online seconds and executed reactivation rounds per query. Awakened counts distinct events evaluated across all rounds, whereas Supporting and Contrastive count the final $|\mathcal{E}_q^+|$ and $|\mathcal{E}_q^-|$ before evidence serialization. Readback is the percentage of queries that dereference at least one source pointer. All means include unsuccessful queries.

Category	#Q	DREAM Reactivation Statistics					
		Time (s)	Rounds	Awakened	Supporting	Contrastive	Readback (%)
Single-Hop	757	12.56	4.00	40.28	4.58	1.92	18.27
Multi-Hop	254	18.20	4.00	59.36	7.95	3.42	37.96
Temporal	289	17.09	4.00	52.28	6.04	4.98	40.71
Open Domain	86	13.92	4.00	45.02	5.23	2.50	25.23
Overall	1386	14.62	4.00	46.57	5.54	2.87	26.99

Table 19: Effect of the repulsive and attractive gate thresholds on LoCoMo using Qwen3-30B-Instruct. Results report mean F1 and BLEU-1 (%) over three runs. Superscript 0 denotes the deployed default, and δ is a prespecified perturbation within the valid score range. The evidence budget and maximum reactivation rounds remain at 4k and four. Tokens counts total online LLM input and output tokens in thousands per query; Calls and Time report LLM calls and end-to-end seconds per query, respectively.

Gate Thresholds		LoCoMo		Online Inference		
ϵ_{rep}	ϵ_{att}	F1	BLEU-1	Tokens (k)	Calls	Time (s)
$\epsilon^0_{\text{rep}} - \delta$	$\epsilon^0_{\text{att}} - \delta$	50.91	42.61	17.84	17.32	14.91
$\epsilon^0_{\text{rep}} - \delta$	ϵ^0_{att}	51.62	43.24	17.03	17.28	14.36
$\epsilon^0_{\text{rep}} - \delta$	$\epsilon^0_{\text{att}} + \delta$	50.47	42.02	15.98	17.19	13.61
ϵ^0_{rep}	$\epsilon^0_{\text{att}} - \delta$	51.48	43.01	18.21	17.35	15.22
ϵ^0_{rep}	ϵ^0_{att}	53.56	44.83	17.36	17.29	14.58
ϵ^0_{rep}	$\epsilon^0_{\text{att}} + \delta$	51.08	42.75	16.19	17.21	13.84
$\epsilon^0_{\text{rep}} + \delta$	$\epsilon^0_{\text{att}} - \delta$	49.96	41.67	18.75	17.41	15.81
$\epsilon^0_{\text{rep}} + \delta$	ϵ^0_{att}	50.83	42.49	17.91	17.34	15.03
$\epsilon^0_{\text{rep}} + \delta$	$\epsilon^0_{\text{att}} + \delta$	50.31	42.04	16.83	17.25	14.31

C.5 Sensitivity Analysis

Gate Thresholds. Table 19 examines threshold changes with the evidence budget fixed at 4k and a maximum of four rounds. At each fixed repulsive threshold, increasing the attractive threshold reduces recorded online token use. With the repulsive threshold at its default, token use decreases from 18.21k to 16.19k as the attractive threshold moves from its lower to its upper setting. This is consistent with admitting fewer candidates as positive support. Conversely, raising the repulsive threshold leaves more candidates available to the attractive gate. The resulting cost variation shows how the two thresholds control the amount of evidence processed online.

Table 20: Effect of the initial awakening budget on LoCoMo using Qwen3-30B-Instruct. Top-k selects the highest-scoring events under ω_i before relation-based expansion. Results report mean F1 and BLEU-1 (%) over three runs. The expansion rule, gate settings, evidence budget of 4k, and maximum of four reactivation rounds are unchanged.

Top-k	Single-Hop		Multi-Hop		Temporal		Open Domain		Overall	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
10	55.94	48.16	41.23	30.87	49.82	41.21	35.72	28.64	50.71	42.33
20	57.36	49.41	44.18	33.29	52.62	43.87	37.14	30.09	52.70	44.10
30	58.12	50.04	45.36	34.28	53.47	44.62	37.91	30.88	53.56	44.83
40	58.01	49.95	45.61	34.47	53.32	44.49	37.83	30.76	53.51	44.78
50	57.68	49.63	45.24	34.14	52.94	44.08	37.42	30.31	53.15	44.44

Table 21: Performance comparison on LongDialQA across three answer backbones. Results report mean F1 and BLEU-1 (%) over three runs for each subset. Avg. is the unweighted mean of the three subset scores. All methods use the same evaluation questions within each backbone.

Model	Method	Big Bang Theory		Friends		The Office		Avg.	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
GPT-4o-mini	Full Context	10.14	9.36	6.58	5.94	6.91	6.02	7.88	7.11
	RAG	8.72	7.85	5.91	5.27	5.84	4.96	6.82	6.03
	A-Mem	6.22	5.60	6.80	7.29	6.44	5.01	6.49	5.97
	Mem0	13.73	12.77	9.42	8.28	9.54	8.46	10.90	9.84
	MemoryOS	13.36	11.63	9.56	8.16	8.92	8.13	10.61	9.31
	LightMem	12.68	11.75	8.37	7.71	8.51	7.36	9.85	8.94
	SimpleMem	13.45	12.57	9.28	8.14	8.86	7.83	10.53	9.51
	GAM	14.23	13.06	9.51	8.32	9.36	8.41	11.03	9.93
	DREAM	15.08	14.16	9.87	8.61	9.68	8.73	11.54	10.50
Qwen3-8B-Instruct	Full Context	13.28	12.67	5.82	5.36	6.74	5.93	8.61	7.99
	RAG	10.46	9.81	4.93	4.58	5.71	5.02	7.03	6.47
	A-Mem	8.36	7.93	5.21	5.64	4.16	3.77	5.91	5.78
	Mem0	15.64	15.12	6.81	6.37	9.25	8.74	10.57	10.08
	MemoryOS	10.38	9.14	6.02	4.17	5.14	4.08	7.18	5.80
	LightMem	14.92	14.41	6.39	6.02	8.64	7.91	9.98	9.45
	SimpleMem	16.27	15.63	6.74	6.48	9.16	8.52	10.72	10.21
	GAM	18.04	17.36	7.16	7.49	10.78	9.92	11.99	11.59
	DREAM	19.12	18.47	7.48	8.13	11.76	10.88	12.79	12.49
Qwen3-30B-Instruct	Full Context	16.43	15.86	6.48	6.79	7.86	7.18	10.26	9.94
	RAG	12.94	12.26	5.38	5.16	6.72	6.08	8.35	7.83
	A-Mem	15.37	14.68	6.79	6.14	4.91	4.36	9.02	8.39
	Mem0	19.06	18.34	7.53	8.01	9.68	8.43	12.09	11.59
	MemoryOS	17.12	16.39	6.73	6.19	5.26	4.48	9.70	9.02
	LightMem	18.37	17.89	7.24	7.66	9.31	8.71	11.64	11.42
	SimpleMem	19.21	18.56	7.47	7.92	10.04	9.17	12.24	11.88
	GAM	20.08	19.41	7.81	8.26	11.12	10.29	13.00	12.65
	DREAM	21.26	20.58	8.06	8.67	11.63	10.76	13.65	13.34

Initial Awakening Budget. Table 20 varies the initial top-k before relation-based expansion. Increasing k from 10 to 30 raises overall F1 from 50.71 to 53.56 and BLEU-1 from 42.33 to 44.83. Multi-hop and temporal F1 improve by 4.13 and 3.65 points, respectively, indicating the value of broader initial evidence coverage. Increasing k to 40 produces a near-plateau at 53.51 F1: multi-hop performance improves slightly, while the other categories decline. At $k = 50$, overall F1 falls to 53.15. These results favor a moderate candidate budget, since additional candidates offer limited benefit under a fixed evidence budget.

C.6 Results on LongDialQA

LongDialQA (Kim et al., 2024) is a long-term multi-party dialogue QA benchmark introduced in the DialSim framework. It is constructed from multi-season TV-show dialogues, including *Friends*, *The Big Bang Theory*, and *The Office*. Each source contains several hundred thousand tokens and many dialogue sessions, requiring agents to track long-range context, multiple speakers, temporal relations, and multi-hop dependencies. The questions are derived from fan quizzes and temporal knowledge graphs. We filter fan quiz questions to ensure they are answerable from dialogue evidence, while temporal knowledge graphs provide dated relational facts for constructing multi-hop questions. Character names are anonymized to reduce reliance on pretrained show knowledge. In

Table 22: Comparison with GAM on LoCoMo across Qwen2.5-Instruct model scales. Each method uses the listed backbone for memory construction and online LLM modules. F1 and BLEU-1 are mean percentages over three runs. Δ (%) is the relative change, $100(\mathrm{DREAM} - \mathrm{GAM})/\mathrm{GAM}$; Overall uses the LoCoMo question-count weights. Tokens counts all online LLM input and output in thousands per query; Rounds counts each method's executed retrieval-integration or reactivation cycles.

Model	Method	Single-Hop		Multi-Hop		Temporal		Open Domain		Overall		Online Inference	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	Tokens (k)	Rounds
Qwen2.5-3B	GAM	35.42	29.14	21.83	14.61	21.37	17.46	18.12	15.02	28.93	23.17	25.40	5.31
	DREAM	38.60	33.20	24.80	16.80	24.60	20.20	17.80	15.30	31.86	26.37	16.90	4.00
	Δ (%)	+8.98	+13.93	+13.61	+14.99	+15.11	+15.69	-1.77	+1.86	+10.13	+13.81	-33.46	-24.67
Qwen2.5-7B	GAM	46.23	43.58	28.41	19.76	30.94	25.53	22.68	18.44	38.31	33.89	26.75	5.48
	DREAM	50.10	48.60	31.80	22.40	34.70	28.90	23.60	19.20	41.89	37.87	17.36	4.00
	Δ (%)	+8.37	+11.52	+11.93	+13.36	+12.15	+13.20	+4.06	+4.12	+9.34	+11.74	-35.10	-27.01
Qwen2.5-14B	GAM	56.38	50.94	37.86	28.27	42.43	34.81	28.67	23.52	48.36	41.72	27.61	5.62
	DREAM	60.60	55.00	41.70	31.80	46.50	38.70	28.40	23.80	52.20	45.41	17.82	4.00
	Δ (%)	+7.48	+7.97	+10.14	+12.49	+9.59	+11.17	-0.94	+1.19	+7.94	+8.84	-35.46	-28.83

our experiments, we evaluate DREAM on the three show-specific subsets and report F1 and BLEU-1 over three runs. LongDialQA tests whether DREAM generalizes from two-party long-term memory to long-term multi-party dialogue understanding.

Table 21 shows that DREAM achieves the highest F1 and BLEU-1 on every subset within each evaluated backbone. Its average F1 reaches 11.54, 12.79, and 13.65 with GPT-4o-mini, Qwen3-8B-Instruct, and Qwen3-30B-Instruct, respectively, exceeding GAM by 0.51, 0.80, and 0.65 points. The corresponding average BLEU-1 scores are 10.50, 12.49, and 13.34. The largest absolute F1 gains over GAM occur on *The Big Bang Theory*, while improvements also extend to the other two shows. These results support the applicability of controlled reactivation to histories containing multiple speakers and recurring events. The low absolute scores also indicate that evidence retrieval and integration over such extended dialogues remain challenging.

C.7 Comparison across Model Scales

Table 22 compares DREAM with GAM using Qwen2.5-3B, 7B, and 14B, with the listed backbone used for both memory construction and online inference. DREAM achieves overall F1 scores of 31.86, 41.89, and 52.20, compared with GAM's 28.93, 38.31, and 48.36. These correspond to relative improvements of 10.13%, 9.34%, and 7.94%, respectively. The gains cover single-hop, multi-hop, and temporal questions at every scale. In particular, multi-hop F1 improves by 2.97, 3.39, and 3.84 points. Open-domain results are less uniform, with small F1 deficits at 3B and 14B, indicating that the benefits are strongest on questions that depend directly on event recall and integration.

DREAM also reduces total online LLM input and output tokens by 33.46%–35.46%. It uses four reactivation rounds, compared with GAM's 5.31–5.62 retrieval-integration rounds. Thus, the token savings include memory processing as well as answer generation. The joint improvement in overall accuracy and token use supports organizing recalled events into explicit supporting and contrastive roles within a bounded inference process.

C.8 Module-Level Cost

Table 23 decomposes DREAM's cost using GPT-4o-mini, separating offline construction per history from online inference per query. Including embedding and indexing, offline construction takes 76.37 seconds per LoCoMo history and 352.90 seconds per LongMemEval history, with event parsing and consolidation accounting for most of this time. Online inference consumes 21.26k and 29.63k LLM input and output tokens, respectively, and takes 16.694 and 21.43 seconds end-to-end. The corresponding mean call counts are 17.29 and 17.38 per query.

The two gates account for 56.73% and 60.61% of online token use on the two benchmarks. Source reconstruction adds a smaller average overhead, at 0.29 and 0.38 LLM calls per query and 0.38 and 0.62 seconds. These call averages are distinct from the fraction of queries that read a source. On LoCoMo, offline construction takes 76.37 seconds per history, totaling 763.7 seconds over the

Table 23: Module-level cost of DREAM using GPT-4o-mini. Calls denote LLM calls, Tokens count LLM input and output tokens, and Time is wall-clock time in seconds, including non-LLM operations. Offline costs are averaged per history and online costs per query over the evaluated workload. Total Online Inference is measured end-to-end, including source reads and all reactivation rounds.

Stage	LoCoMo			LongMemEval		
	Calls	Tokens	Time	Calls	Tokens	Time
▼ *Offline construction: per dialogue history*						
Event Parsing and Consolidation	12.00	24560	68.40	54.00	112480	318.20
Embedding and Indexing	0.00	0	7.97	0.00	0	34.70
▼ *Online inference: per query*						
Cue Extraction and Event Awakening	4.00	2140	2.28	4.00	2310	2.46
Repulsive Encoding Gate	4.00	6940	4.12	4.00	10320	5.73
Attractive Decision Gate	4.00	5120	3.47	4.00	7640	4.86
Source Reconstruction	0.29	460	0.38	0.38	790	0.62
Evidence Transformation	4.00	4380	3.16	4.00	5960	4.09
Answer Generation	1.00	2220	3.284	1.00	2610	3.67
Total Online Inference	17.29	21260	16.694	17.38	29630	21.43

ten histories, as reported in the Construction column of Table 4. Final answer generation takes 3.284 seconds per query and is included in the 16.694-second end-to-end online time. The latter also includes awakening, both gates, source reconstruction, and evidence transformation. The 1,984 tokens in Table 4 count only the memory portion of the final answer input, whereas the 2,220 tokens in the Answer Generation row include the full input and output of that call.

D Case Study

We use four illustrative cases to show how DREAM assigns query-dependent roles to awakened memories. The cases focus on state revision, event identity, source reconstruction, and incomplete evidence. Each case presents the source dialogue, relevant memory states, evidence roles, and an expected answer.

Reading the examples. Source identifiers refer to turns within each illustration, and event identifiers refer to the corresponding compact records. Green **[Support]** labels indicate evidence eligible to support the specific query; red **[Constraint]** labels identify limits on its use; gray **[Silent]** labels indicate records that need not enter the answer-facing state. The labels remain interpretable without color. Event roles are query-dependent: a historically valid record can support a retrospective question while constraining a current-state answer.

D.1 State Revision without Erasing History

Case 1: Partial revision with retained evidence

Source dialogue

S1, 3 March, user: "For my Berlin conference trip, please book Hotel Linden. My budget is EUR 180 per night."
S2, 8 March, user: "For that Berlin trip, cancel Hotel Linden and use Hotel Spree instead."

Reactivation-ready event records

E1: schema = travel accommodation; slots = {trip: Berlin conference, hotel: Linden, budget: EUR 180/night}; slot sources = S1; state = superseded hotel preference; accessibility = historical; $p_1 = \{S1\}$, with a state-update link to E2.
E2: the revised record for the same trip; slots = {trip: Berlin conference, hotel: Spree, budget: EUR 180/night}; trip and budget retain S1 as their source, while the revised hotel cites S2; state = revised hotel preference; accessibility = active; $p_2 = \{S1, S2\}$.

Query A: "Which hotel should you book for my Berlin conference now, and what is my nightly budget?"

[Constraint] E1 constrains the obsolete hotel preference: do not use Hotel Linden as the current hotel (S1–S2). This restriction does not invalidate the budget.

Case 1: Partial revision with retained evidence (continued)

[Support] E2 supplies Hotel Spree (S2) and the retained EUR 180/night budget (S1). Both supporting claims enter through E2.
Transformed evidence: $\mathcal{S}_q$: Spree is the current hotel preference (E2, S2) and the budget is EUR 180/night (E2, S1); $\mathcal{N}_q$: Linden is not the current hotel preference; $\mathcal{H}_q$: Linden $\rightarrow$ Spree; $\mathcal{P}_q$: hotel $\mapsto$ S2, budget $\mapsto$ S1; $\mathcal{G}_q$: answer for the current trip and retain the budget's nightly scope.
Expected answer: Hotel Spree, with a budget of EUR 180 per night.

Query B: "Which hotel did I choose before changing to Hotel Spree?"

[Support] E1 supplies the historical hotel, Hotel Linden; S2 establishes the change boundary. **Expected answer:** Hotel Linden.

Interpretation. The hotel update does not repeat the budget. E2 retains the budget and its original source, so the budget can enter positive evidence even when E1 constrains the old hotel. The records have disjoint roles for Query A but share S1 as a source. Query B still has access to the historical hotel.

D.2 Event Identity beyond Semantic Similarity

Case 2: Similar activities do not imply the same event

Source dialogue

S1, 10 May, user: "Our school fundraiser on 20 May will support the Riverside Animal Shelter."
S2, 2 June, user: "The neighborhood fundraiser on 15 June will support the Children's Hospital."
S3, 12 June, user: "For the neighborhood fundraiser on 15 June, the venue has changed from the library to the community hall. The beneficiary is unchanged."

Query

"Which organization will receive the proceeds from the school fundraiser on 20 May?"

Awakening and role assignment

Structured cues: schema = fundraiser; slots = {organizer: school, date: 20 May, requested attribute: beneficiary}. A beneficiary is requested, rather than supplied as a retrieval filter.
[Support] E1 matches the organizer and date and supplies Riverside Animal Shelter (S1).
[Constraint] If E2 is awakened through its shared schema, it is event-mismatched: Children's Hospital belongs to the neighborhood fundraiser on 15 June, not the school fundraiser on 20 May (S1–S2).
[Silent] E3 revises the venue of the neighborhood event. It provides no additional support for the beneficiary of the school event.
Transformed constraint: Do not transfer the neighborhood event's beneficiary to the school event. The more recent statement does not supersede S1 because the two statements concern different events.
Expected answer: Riverside Animal Shelter.

Interpretation. Recency alone is insufficient for validity checking. The relevant comparison includes the entity, event scope, attribute, and time. Here, an event-mismatched record is not false; it is unsuitable support for the requested event. Global-local fields make this distinction explicit, while the repulsive gate prevents cross-event attribute transfer.

D.3 Source Reconstruction for an Exact Historical Detail

Case 3: A pointer recovers a distinction omitted by compression

Source dialogue

S1, 4 September, user: "I can attend the workshop on either 12 or 19 October."
S2, 6 September, assistant: "The organizer offers the 12 October session at 09:00 or the 19 October session at 14:00."
S3, 7 September, user: "I confirm my attendance on 19 October at 14:00. I cannot attend on 12 October."

Compressed event and query

The compact event records a confirmed October workshop, but omits the exact date and time; its source pointers retain S1–S3. The query asks: "What date and time did I confirm for the workshop?"

Case 3: A pointer recovers a distinction omitted by compression (continued)

Source-grounded reconstruction

The summary is topically relevant but insufficient for the requested precision. Dereferencing S1–S3 distinguishes initial availability, the assistant's offered alternatives, and the user's final confirmation.
[Support] S3 explicitly confirms 19 October at 14:00.
[Constraint] The 12 October option was explicitly declined in S3; S1 and S2 must not be presented as the final confirmation.
Transformed evidence: $\mathcal{H}_q$: alternatives $\rightarrow$ confirmation; $\mathcal{P}_q$: S1–S3; $\mathcal{G}_q$: preserve the distinction between an offered option and an accepted booking. Do not invent a year or time zone.
Expected answer: 19 October at 14:00.

Interpretation. A source pointer provides access to original dialogue, rather than merely to another generated summary. Readback recovers the evidence needed to separate a proposal from a decision. The answer's precision is limited to what the source states; compression recovery does not license unsupported date normalization.

D.4 Boundary Case: No Confirmed Replacement

Case 4: Excluding an old plan does not establish a new one

Source dialogue

S1, user: "I accepted the job offer from Cedar Labs."
S2, later user turn: "I have withdrawn my acceptance of Cedar Labs. I am still considering Pine Research, but I have not decided."
Query: "Which company have I now accepted an offer from?"
[Constraint] The Cedar Labs acceptance has been withdrawn. It cannot support a current accepted-employer answer (S1–S2).
[Support] S2 supports a limited status statement: Pine Research is under consideration, with no acceptance established by the record.
Evidence gap: The source establishes neither an accepted offer from Pine Research nor another confirmed replacement. Reusing an already read record cannot resolve this gap.
Expected answer: "The conversation does not establish a current accepted offer. You withdrew from Cedar Labs and had not decided about Pine Research."
Boundary: If cancellation evidence is absent from both the stored event and accessible sources, the gates may fail to detect that an earlier acceptance has become outdated. Reactivation cannot reconstruct evidence that was never retained.

Interpretation. The repulsive gate limits the use of an old claim; it does not prove the opposite claim or identify a replacement. The attractive gate must retain the distinction between a possibility and a confirmed decision. A calibrated answer therefore reports the supported status and the remaining gap instead of turning uncertainty into a new fact.

Table 24: Overview of the four illustrative cases and the DREAM components they examine.

Illustration	Risky use of memory	Required distinction	Relevant component
Case 1	Old preference used as current support	Current versus historical query	State, accessibility, dual gates
Case 2	Attribute transferred between similar events	Event identity versus topic overlap	Global-local encoding, cues, repulsive gate
Case 3	Proposed option treated as confirmed detail	Summary versus original source evidence	Pointers, reconstruction, transformation
Case 4	Unconfirmed replacement promoted to an answer	Exclusion versus positive evidence	Attractive gate, generation constraints

E Impact and Future Work

E.1 Broader Impact

DREAM contributes to the development of reliable long-horizon LLM agents by showing that awakened memories should not always be treated as positive evidence. In extended interactions, a memory can be relevant to the current query while still being outdated, conflicting, or associated with a different event state. By separating memory awakening from evidential support, DREAM provides a more controlled way to use long-term memory during generation.

More broadly, this perspective is important for persistent agents in applications such as personal assistants, educational systems, healthcare support, customer service, and collaborative tools. In these settings, agents must not only recall past information, but also determine whether the recalled information remains valid for the current user state, task, or environment. DREAM suggests that modeling the evidential role of memory is a useful design principle for building more trustworthy memory-augmented agents.

At the same time, stronger long-term memory systems also raise privacy and safety concerns. Persistent memories may contain sensitive personal information, preferences, behavioral patterns, and historical decisions. If such memories are stored, retrieved, or reactivated without appropriate safeguards, they may cause privacy leakage, over-personalization, or inappropriate use of outdated information. We believe that progress in long-term agent memory should be accompanied by user consent, transparent memory inspection, controllable editing and deletion, secure storage, and clear documentation of memory usage.

E.2 Limitations

Although DREAM improves the reliability of long-term memory use, several limitations remain.

First, experiments are conducted on representative long-term memory and long-context QA benchmarks, including LoCoMo, LongMemEval, HotpotQA, and NarrativeQA. These datasets cover multi-session memory, temporal reasoning, knowledge updates, multi-hop retrieval, and long-document understanding. However, they do not fully capture open-ended agentic environments where user goals, tool states, and external conditions evolve continuously over longer time spans. The question-disjoint LoCoMo split evaluates unseen questions over shared histories; generalization to unseen users and conversation histories is not separately assessed.

Second, DREAM focuses on controlling how awakened memories influence generation, but it still depends on the quality of the underlying memory store. If important events are missing, incorrectly summarized, or poorly indexed, role-aware reactivation may not recover the necessary evidence. Therefore, DREAM should be viewed as complementary to memory writing, consolidation, and retrieval methods.

Finally, the current framework relies on LLM-based reasoning to identify memory roles and distinguish supportive evidence from outdated or mismatched memories. This design is flexible but may introduce additional inference cost and model-dependent uncertainty. Extending DREAM to more efficient, multimodal, and tool-grounded memory settings remains an important challenge.

E.3 Future Work

Future work will focus on improving the generality, efficiency, and scope of memory reactivation.

A first direction is to evaluate DREAM in more interactive long-horizon agent environments, where agents must maintain goals, revise plans, use tools, and adapt to changing user or world states. Such settings would better test whether role-aware reactivation can reduce memory-induced errors during extended decision-making.

A second direction is to integrate DREAM with stronger memory construction and consolidation mechanisms. Future systems could jointly optimize memory writing, updating, retrieval, and reactivation, preserving outdated memories as historical contrastive evidence while explicitly linking them to the newer states that supersede them.

Finally, future work may explore more efficient and broader forms of reactivation. Lightweight verifier models, caching, distillation, or hybrid symbolic-neural mechanisms could reduce inference cost. Extending DREAM to multimodal memories, structured databases, and tool execution traces would further support reliable long-term memory use in practical LLM agents.