HIPPOCAMPUS–NEOCORTEX-INSPIRED SPARSE–DENSE MEMORY SELF-ORGANIZATION FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory is essential for LLM agents to retain and reuse information, but reliable associative access remains challenging over extended interactions. Relevant evidence may span temporally distant interactions, yet existing methods often decouple associative memory organization from fine-grained evidence retrieval and therefore miss related information. Inspired by hippocampus–neocortex interactions in human memory, we propose LUMI-MEM, a sparse–dense memory self-organization framework for LLM agents. Within the resulting memory bank, Sparse Attractor (SA) performs associative indexing over topic basins, while Dense Representation (DR) supports fine-grained memory consolidation over candidate chunks. Sparse-to-dense activation and dense-to-sparse calibration close the bidirectional interaction loop, forming a brain-inspired mechanism for memory organization. Experiments on LongMemEval, 2WikiMultiHopQA, and MuSiQue show that LUMI-MEM achieves the best R@5 on all three benchmarks and the best average QA performance with GPT-4.1-mini. Ablations with different retrievers show that SA, DR, and the sparse–dense interaction modules each contribute to performance, supporting self-organization as a practical mechanism for memory retrieval.

1 INTRODUCTION

Large language model (LLM) agents increasingly operate over long-horizon interactions, where useful information accumulates across conversations, tasks, and temporally distant experiences (Izacard et al., 2022; Lewis et al., 2020). However, effective memory retrieval remains challenging because the information needed to answer a query rarely has a simple surface-level match (Song et al., 2025; Wu et al., 2024). In long-term conversational memory, useful evidence may be scattered across temporally distant interactions, expressed in different surface forms, and informative only when multiple related memories are combined (Tulving, 2002). Memory therefore depends not only on storage but also on self-organization: associative cues index relevant memory regions, and the retrieved evidence in turn refines the associative index.

Existing memory systems have explored different strategies for retrieval. Methods such as Mem0 (Chhikara et al., 2025) and EverMemOS (Hu et al., 2026) introduce factual updating and hierarchical organization, but memory organization and fine-grained evidence retrieval remain only loosely coupled. Graph-based retrieval methods, such as HippoRAG, introduce knowledge graphs for associative retrieval, but graph traversal often provides only coarse-grained evidence (Edge et al., 2024; Gutiérrez et al., 2024). As memory grows, graph updating and maintenance also introduce substantial overhead (Wu et al., 2026). While hierarchical methods capture broader contextual information (Sarthi et al., 2024; Zhao et al., 2024), they may dilute fine-grained evidence, especially when answers depend on local cues distributed across multiple dialogue episodes. LLM-augmented approaches, such as ThinkQE, PBR, and ComoRAG, use query expansion (Lei et al., 2025; Zhang et al., 2026) or iterative refinement (Wang et al., 2026b) to bridge the query–evidence gap, but they typically require multiple inference rounds and additional tokens. Taken together, existing systems consider both associative indexing and fine-grained retrieval, but lack a self-organization mechanism through which the two mutually shape memory storage and access.

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

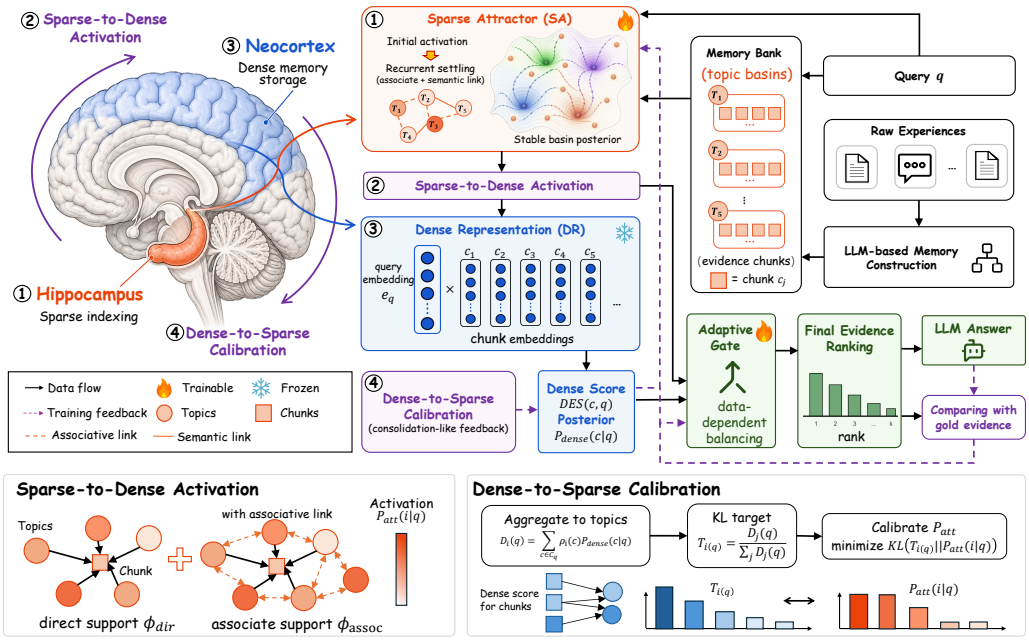

Figure 1: **Overview of the LUMI-MEM framework.** Inspired by hippocampus–neocortex interactions, LUMI-MEM forms a sparse–dense memory organization. The Sparse Attractor (SA) indexes relevant topic basins, sparse-to-dense activation maps topic-level activations to chunk-level retrieval signals, and the Dense Representation module (DR) scores candidate chunks within the activated context. Dense-to-sparse calibration further aligns the SA posterior with dense evidence feedback during training, while an adaptive gate combines sparse and dense signals for final retrieval.

In contrast, human memory systems exhibit continuous self-organization, largely relying on the interactions between the hippocampus and the neocortex. The hippocampus provides fast, sparse indexing of cortical representations (Scoville & Milner, 1957; Teyler & Rudy, 2007), whereas the neocortex stores fine-grained memories (McClelland et al., 1995). During retrieval, the hippocampus activates distributed cortical representations within relevant memory regions, and cortical recognition signals can in turn modulate hippocampal dynamics to support memory updating (Preston & Eichenbaum, 2013). This bidirectional sparse–dense interaction forms a core mechanism of memory self-organization, offering a concrete algorithmic principle for artificial memory.

Inspired by hippocampus–neocortex interactions in human memory, we propose LUMI-MEM, a sparse–dense memory self-organization framework for LLM agents. As shown in Figure 1, LUMI-MEM organizes raw dialogue turns into sparse topic-level memory traces and merges related traces into topic basins. Given a query, the Sparse Attractor (SA), a hippocampal-style network, propagates activation over the topic basins to index relevant memory regions, yielding a topic posterior. This posterior is projected to the chunk level to expand the candidate set. Dense Representation (DR) then performs fine-grained memory consolidation via embedding similarity over the candidate chunks, while an adaptive gate learns to balance SA signals with DR scores for final retrieval. During training, topic-level DR feedback calibrates SA, forming a consolidation-like mechanism that closes the sparse–dense interaction loop.

In summary, our contributions are threefold:

- We propose LUMI-MEM, which translates the self-organizing principles of the human hippocampal–cortical memory system into a brain-inspired framework.
- We introduce a bidirectional sparse–dense interaction mechanism that aligns sparse memory activation with dense evidence recognition.
- We achieve the best QA performance on 2WikiMultiHopQA and MuSiQue, improving average QA F1 by 11.7% over the strongest baseline. Particularly in long-term memory retrieval, LUMI-MEM achieves relative gains of 8.7% in N@5 on LongMemEval.

2 Related Work

Existing work on memory for LLM agents spans both memory organization and memory access. Memory systems such as Mem0, A-Mem, and MemoryOS maintain compact, evolving memories through LLM-augmented extraction and updating (Chhikara et al., 2025; Xu et al., 2026; Kang et al., 2025), while EverMemOS further organizes episodic traces into higher-level memory structures. Memory-R1 learns memory-management policies through reinforcement learning (Yan et al., 2026), and HeLa-Mem models associative memory through a dynamic Hebbian graph (Zhu et al., 2026). These methods improve how experiences are stored and organized, but memory organization and fine-grained evidence retrieval remain only loosely coupled.

A complementary line of work focuses on retrieving stored evidence. Similarity-based retrievers such as BM25 and Contriever rank chunks by lexical or embedding similarity (Robertson & Walker, 1994; Izacard et al., 2022; Ding & Zhai, 2025). Structure-augmented methods range from GraphRAG and RAPTOR (Edge et al., 2024; Sarthi et al., 2024) to recent hypergraph formulations such as HGRAG, HyperGraphRAG, and HyperRAG (Wang et al., 2026a; Luo et al., 2026; Feng et al., 2025), although graph noise and maintenance can limit scalability. Query- and feedback-enhanced methods, including ThinkQE and PBR, improve retrieval through query expansion, personalization, or iterative refinement (Lei et al., 2025; Zhang et al., 2026; Tan et al., 2025), while Self-RAG allows the model itself to decide when to retrieve or reflect (Asai et al., 2024). Recent work has also explored biologically or cognitively inspired memory (Fountas et al., 2025; Liao, 2025). HippoRAG uses hippocampal indexing theory with knowledge graphs and Personalized PageRank, while HippoRAG-v2 extends this view toward non-parametric continual memory (Gutiérrez et al., 2024; 2025). MemoRAG uses a dual-system memory to generate retrieval clues, whereas ComoRAG employs a dynamic memory workspace for iterative evidence refinement (Qian et al., 2025; Wang et al., 2026b). Overall, these RAG methods primarily focus on improving memory retrieval, whereas memory self-organization remains less explicitly modeled.

3 Problem Statement

Our objective is to develop a memory architecture that supports retrieval across conversations with personal AI assistants. Let $\mathcal{D} = \{d_1, \ldots, d_S\}$ be a corpus of episodic documents, such as personal conversation sessions or knowledge articles. A memory bank $\mathcal{M} = \{m_1, \ldots, m_K\}$ is constructed from $\mathcal{D}$. For personalized-memory data, each user has an independent memory bank, and a query accesses only its associated user's bank. Each memory unit $m_i = \{\mathrm{summary}_i, e_i, \mathcal{C}_i, \mathrm{conf}_i\}$ represents a semantically coherent topic with a summary, an embedding e_i, a set of evidence chunks $\mathcal{C}_i$, and a confidence score $\mathrm{conf}_i \in [0, 1]$ indicating how confidently the summary is supported by $\mathcal{C}_i$. The full candidate chunk pool is defined as $\mathcal{C} = \bigcup_{i=1}^{K} \mathcal{C}_i$; each chunk can be attributed to multiple source topics.

Given a query q, let $\mathcal{G}_q \subseteq \mathcal{C}$ denote the set of gold evidence chunks for q. The retrieval task is to return a ranked list $\pi_q = (c_1, c_2, \ldots)$ over $\mathcal{C}$ such that chunks in $\mathcal{G}_q$ appear at high ranks. We evaluate retrieval performance using $\mathrm{Recall@}k$ ($\mathrm{R@}k$) and $\mathrm{NDCG@}k$ ($\mathrm{N@}k$), and further assess LLM answer-generation quality using Exact Match (EM) and token-level F1 scores.

4 Lumi-Mem Framework

4.1 Memory Bank Construction

Raw sessions $\mathcal{D}$ are converted into structured memory units through an LLM-based module as a preprocessing step before training. For each session, the LLM receives the raw episodic document and extracts topics of the form m_i introduced in Section 3, where the embedding e_i is obtained by encoding the topic summary into the shared embedding space. For each new topic, embedding similarity retrieves a shortlist of existing memory units. A second LLM call then decides whether to add a new unit or merge the topic into one shortlisted unit. Merging appends the new evidence and updates the summary while preserving corrections and temporal changes. Appendix D provides the complete preprocessing procedure.

4.2 SPARSE ATTRACTOR DYNAMICS

Here, Sparse Attractor (SA) denotes query-conditioned recurrent settling toward a stable topic posterior. SA maps the query to an initial distribution over topic basins, providing the cue for associative retrieval. For a query embedding $e_q \in \mathbb{R}^d$, let $\ell_i^0 = e_q^\top e_i/\sqrt{d}$ be the cue logit for basin i, let $\boldsymbol{\ell}^0 = (\ell_i^0)_{i=1}^K$, and initialize $\mathbf{p}^0 = \mathrm{softmax}(\boldsymbol{\ell}^0/\tau_{\mathrm{topic}})$, where $\tau_{\mathrm{topic}} > 0$ is the topic temperature.

To provide fixed pathways for propagation, SA builds a semantic graph from positive inter-basin embedding similarities and an associative graph from shared episodic metadata (e.g., temporal and event links). Both graphs are then symmetrized and normalized by node degree. We denote the resulting adjacency matrices by $\widetilde{\mathbf{W}}_{\mathrm{sem}}$ and $\widetilde{\mathbf{W}}_{\mathrm{assoc}}$.

To adapt these pathways to each query, a shared basin-conditioned adapter f_ϕ converts the query-specific state of each basin into four controls: reliability $r_i(q)$, semantic gain $g_i^{\mathrm{sem}}(q)$, associative gain $g_i^{\mathrm{assoc}}(q)$, and activation threshold $\theta_i(q)$. Together, these controls determine how strongly a basin participates in semantic and associative propagation and how readily it is activated. We define the effective bilateral gains $g_{i,\mathrm{eff}}^{\bullet}(q) = r_i(q) g_i^{\bullet}(q)$ for $\bullet \in \{\mathrm{sem}, \mathrm{assoc}\}$ and collect them in $\mathbf{g}_{\mathrm{eff}}^{\bullet}(q)$.

To make propagation query dependent while retaining symmetry, SA applies the controls to both endpoints and forms the excitation matrix

$$\begin{aligned}\mathbf{B}(q) = \eta_{\mathrm{sem}} \,\mathrm{Diag}(\mathbf{g}_{\mathrm{eff}}^{\mathrm{sem}}(q))\widetilde{\mathbf{W}}_{\mathrm{sem}}\,\mathrm{Diag}(\mathbf{g}_{\mathrm{eff}}^{\mathrm{sem}}(q)) \\ + \eta_{\mathrm{assoc}}\,\mathrm{Diag}(\mathbf{g}_{\mathrm{eff}}^{\mathrm{assoc}}(q))\widetilde{\mathbf{W}}_{\mathrm{assoc}}\,\mathrm{Diag}(\mathbf{g}_{\mathrm{eff}}^{\mathrm{assoc}}(q)),\end{aligned} \tag{1}$$

where $\eta_{\mathrm{sem}}, \eta_{\mathrm{assoc}} \geq 0$ are base interaction strengths. Bilateral control preserves $\mathbf{B}(q) = \mathbf{B}(q)^\top$.

To stabilize recurrent propagation, SA combines query-conditioned excitation with residual query evidence and query-wise adaptive inhibition $\kappa_0(q)$, then applies a damped softmax refinement for T steps. This limits abrupt posterior changes while allowing activation to spread through the topic graphs. Denoting the resulting state by $\mathbf{p}^T$, SA outputs $P_{\mathrm{att}}(i \mid q) = p_i^T$. Appendix A.4 details the graph construction, the adapter features and parameterization, and the exact recurrent update.

To characterize the fixed points and their uniqueness, we use the entropy-regularized energy in Eq. 21, whose stationary points coincide with the update's fixed points. For each fixed query q, if $\kappa_0(q) > \lambda_{\max}(\mathbf{B}(q))$, with $\lambda_{\max}$ denoting the largest eigenvalue, the energy is strictly convex and the corresponding equilibrium is unique. Appendix C.1 gives the corresponding derivation.

4.3 DENSE REPRESENTATION

The DR module provides a fine-grained scorer over individual evidence chunks. For a chunk embedding $h_c \in \mathbb{R}^d$, DR uses an alignment matrix $\mathbf{R} \in \mathbb{R}^{d\times d}$ to compute $\mathrm{DR}_{\mathrm{raw}}(c, q) = e_q^\top \mathbf{R} h_c/\sqrt{d}$. $\mathbf{R}$ is pretrained with a listwise objective that maximizes the posterior mass of gold chunks under $\mathrm{softmax}(\mathrm{DR}_{\mathrm{raw}}/\tau_{\mathrm{pre}})$. After pretraining, DR is kept fixed during LUMI-MEM training.

All chunk-level scoring is performed over a query-specific candidate set that combines a dense retrieval surface with the chunks attached to SA-activated basins:

$$\mathcal{C}_q = \mathrm{TopK}(\mathrm{DR}_{\mathrm{raw}}(\cdot, q), k_{\mathrm{DR}}) \cup \bigcup_{i \in \mathrm{TopK}(P_{\mathrm{att}}(\cdot|q), k_{\mathrm{basin}})} \mathcal{C}_i. \tag{2}$$

In this way, SA finds associative candidates that may not be retrieved by dense similarity alone.

Over the candidate pool, we normalize the raw score and form the corresponding dense posterior as

$$\mathrm{DR}(c, q) = \mathrm{zscore}_{c\in\mathcal{C}_q}(\mathrm{DR}_{\mathrm{raw}}(c, q)), \qquad P_{\mathrm{dense}}(c \mid q) = \mathrm{softmax}(\mathrm{DR}(\cdot, q)/\tau_{\mathrm{dense}})_c. \tag{3}$$

Here, $\mathrm{zscore}_{c\in\mathcal{C}_q}$ denotes standardization across the candidate set to zero mean and unit variance.

4.4 SPARSE-TO-DENSE ACTIVATION

Sparse-to-dense activation converts the topic posterior $P_{\mathrm{att}}(\cdot \mid q)$ into a retrieval signal for each chunk $c \in \mathcal{C}_q$. Its inputs are the posterior, the chunk's source-basin index set $\mathrm{src}(c) = \{i : c \in \mathcal{C}_i\}$,

Algorithm 1 Lumi-Mem Inference

Require: Query q; memory bank $\mathcal{M} = \{m_i\}_{i=1}^{K}$ with e_i and $\mathcal{C}_i$; topic graph; trained shared adapter f_ϕ; pretrained fixed DR module; gate MLP
Ensure: Ranked evidence chunks
1: Initialize $\mathbf{p}^0$, compute the adapter controls, construct $\mathbf{B}(q)$ using Eq. 1, and run the damped SA update for T steps (Appendix A.4) to obtain $P_{\mathrm{att}}(i \mid q) = p_i^T$
2: $\mathcal{C}_q \leftarrow \mathrm{TopK}(\mathrm{DR}_{\mathrm{raw}}(\cdot, q), k_{\mathrm{DR}}) \cup \bigcup_{i \in \mathrm{TopK}(P_{\mathrm{att}}(\cdot|q), k_{\mathrm{basin}})} \mathcal{C}_i$
3: Compute $\mathrm{DR}(c, q)$ for all $c \in \mathcal{C}_q$ using Eq. 3
4: **for** each chunk $c \in \mathcal{C}_q$ **do**
5: Compute $s^{\mathrm{s2d}}(c, q)$ using Eq. 4
6: Compute $\mathrm{Gate}(\mathbf{f}_{qc})$ and $\mathrm{score}(c \mid q)$ using Eq. 8
7: **end for**
8: **return** chunks in $\mathcal{C}_q$ sorted by $\mathrm{score}(c \mid q)$ in descending order

and the fixed associative topic graph. A chunk receives direct support from its own activated basins and associative support from neighboring activated basins.

To produce the chunk-level output, let $\boldsymbol{\psi}(c, q) = [\psi_{\mathrm{dir}}(c, q), \psi_{\mathrm{assoc}}(c, q)]^\top$ collect the direct and associative support. Their learned mixture is

$$s^{\mathrm{s2d}}(c, q) = \mathrm{zscore}_{c \in \mathcal{C}_q} \left(\mathbf{w}^\top \boldsymbol{\psi}(c, q) \right). \tag{4}$$

Here, $\mathbf{w} = \mathrm{softmax}(\mathbf{w}^{\mathrm{raw}})$ gives non-negative mixture weights, and $s^{\mathrm{s2d}}(c, q)$ is the normalized sparse support passed to adaptive fusion. Appendix A.4 gives the exact support calculations.

4.5 Dense-to-Sparse Calibration and Adaptive Fusion

The previous two modules move information in distinct directions. SA produces a topic-level posterior $P_{\mathrm{att}}(i \mid q)$ and a chunk-level sparse-to-dense activation $s^{\mathrm{s2d}}(c, q)$, while DR produces a chunk-level evidence posterior $P_{\mathrm{dense}}(c \mid q)$. This subsection closes the loop: dense evidence is aggregated back to topics to calibrate the attractor, and the calibrated sparse-to-dense activation is then adaptively fused with scores from DR.

Dense evidence as topic feedback. The dense posterior is aggregated from chunks back to topics as

$$D_i(q) = \sum_{c \in \mathcal{C}_q} \rho_i(c) P_{\mathrm{dense}}(c \mid q), \tag{5}$$

where $\rho_i(c) = \mathbb{I}[i \in \mathrm{src}(c)] / (\sum_{j=1}^{K} \mathbb{I}[j \in \mathrm{src}(c)] + \varepsilon)$ redistributes each chunk's dense score equally across its source basins. Thus, $D_i(q)$ gives the dense support assigned to each topic basin.

Attractor calibration. When gold evidence is available during training, we define the topic-level supervision signal as $G_i(q) = \mathbb{I}[\exists\, c \in \mathcal{G}_q \text{ s.t. } i \in \mathrm{src}(c)]$. The calibration target combines a normalized gold-topic signal with dense feedback:

$$\widehat{P}_{\mathrm{cal}}(i \mid q) = \alpha \frac{G_i(q)}{\sum_{j=1}^{K} G_j(q) + \varepsilon} + (1 - \alpha) D_i(q), \qquad P_{\mathrm{cal}}(\cdot \mid q) = \mathrm{Norm}\Big(\widehat{P}_{\mathrm{cal}}(\cdot \mid q) \Big). \tag{6}$$

Here, Norm rescales a non-negative vector to unit sum. The mixing coefficient $\alpha \in [0, 1]$ balances gold-topic supervision against dense feedback. Let $P_{\mathrm{cal}}(\cdot \mid q)$ be the resulting calibration-target distribution over topic basins. The scalar $\gamma(q) \in [0, 1]$ adapts calibration strength to dense uncertainty, attractor uncertainty, and task demand. We train SA with

$$\mathcal{L}_{\mathrm{d2s}} = \mathrm{sg}(\gamma(q))\, D_{\mathrm{KL}} \left(\mathrm{sg}(P_{\mathrm{cal}}(\cdot \mid q)) \parallel P_{\mathrm{att}}(\cdot \mid q) \right). \tag{7}$$

The stop-gradient operator $\mathrm{sg}(\cdot)$ keeps both $\gamma(q)$ and $P_{\mathrm{cal}}(\cdot \mid q)$ fixed during backpropagation, so calibration updates $P_{\mathrm{att}}(\cdot \mid q)$ only through the KL term.

Adaptive fusion. LUMI-MEM combines dense recognition with sparse-to-dense activation through an adaptive non-negative gate. Here, $u_{\mathrm{att}}(q)$ is the normalized attractor entropy defined in Appendix A.5, and $\mathbf{f}_{qc} = [\mathrm{DR}(c,q), u_{\mathrm{att}}(q), \psi_{\mathrm{dir}}(c,q), \psi_{\mathrm{assoc}}(c,q)]$ is the gate feature vector. The gate and final evidence score are

$$\mathrm{Gate}(\mathbf{f}_{qc}) = \mathrm{softplus}(\mathrm{MLP}(\mathbf{f}_{qc}))\,, \qquad \mathrm{score}(c \mid q) = \mathrm{DR}(c,q) + \mathrm{Gate}(\mathbf{f}_{qc})\, s^{\mathrm{s2d}}(c,q). \tag{8}$$

The softplus activation keeps the gate non-negative, so it scales the sparse index correction. Thus, DR remains the base scorer, while sparse memory contributes a learned, query-dependent correction.

4.6 TRAINING OBJECTIVE AND RETRIEVAL PIPELINE

LUMI-MEM is optimized with a combination of evidence ranking, dense-to-sparse attractor calibration, and gate regularization:

$$\mathcal{L} = \mathcal{L}_{\mathrm{rank}} + \eta_{\mathrm{d2s}}\mathcal{L}_{\mathrm{d2s}} + \eta_{\mathrm{gate}}\mathcal{L}_{\mathrm{gate}}. \tag{9}$$

Here, $\eta_{\mathrm{d2s}}, \eta_{\mathrm{gate}} \geq 0$ are the loss weights for dense-to-sparse calibration and gate regularization, respectively. $\mathcal{L}_{\mathrm{rank}}$ is a listwise evidence-ranking loss, $\mathcal{L}_{\mathrm{d2s}}$ calibrates SA using dense evidence feedback, and $\mathcal{L}_{\mathrm{gate}}$ regularizes the gate magnitude.

Training updates only lightweight parameters in the sparse and fusion components: the shared adapter parameters ϕ, sparse-to-dense activation weights, and the gate MLP. The pretrained DR module remains frozen and provides two fixed signals: DR scores for ranking and a stop-gradient dense posterior for calibrating SA. Algorithm 1 summarizes the inference pass. During training, we run the same forward pass, aggregate the dense posterior back to topic basins to form the calibration target $P_{\mathrm{cal}}(\cdot \mid q)$, and update the trainable components with the joint loss in Eq. 9.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets. LUMI-MEM is evaluated on three memory retrieval benchmarks spanning personal-memory QA and multi-hop QA. **LongMemEval** (Wu et al., 2024) focuses on long-term interactions between humans and AI assistants and contains 500 distinct user queries. Each query is paired with timestamped chat histories spanning roughly 40 sessions, with an average context length of 115K tokens. **2WikiMultiHopQA** (Ho et al., 2020) is evaluated on 1,000 sampled questions with 6,119 associated passages. **MuSiQue** (Trivedi et al., 2022) is evaluated on 1,000 sampled questions with 11,656 associated passages.

Baselines. We compare four groups of baselines. **(1) Full-context LLM.** The LLM agent directly processes the available full context. **(2) Direct Retrieval.** This group includes standard lexical and dense retrievers, including BM25 (Robertson & Walker, 1994), Contriever (Izacard et al., 2022), and Stella (Zhang et al., 2024). **(3) Enhanced Retrieval.** BKRAG reranks candidate chunks with BGE (Guo et al., 2024), PBR incorporates user-specific features into query expansion (Zhang et al., 2026), ThinkQE expands queries through iterative LLM reasoning (Lei et al., 2025), and GFM-RAG performs graph-based retrieval (Luo et al., 2025). **(4) Memory-based Systems.** HippoRAG-v1 and HippoRAG-v2 organize external knowledge into bio-inspired graphs (Gutiérrez et al., 2024; 2025), while ComoRAG uses an iterative memory workspace to retrieve and consolidate evidence (Wang et al., 2026b). For end-to-end QA, we include Mem0 (Chhikara et al., 2025), EverMemOS (Hu et al., 2026), and HeLa-Mem (Zhu et al., 2026) as recent long-term memory baselines.

Implementation Details. All methods use GPT-4.1-mini for answer generation; auxiliary LLM components use GPT-4o-mini with temperature 0. Enhanced-retrieval and memory-based retrieval methods use Contriever as the retriever. We use a query-level 80%/20% train–test split; all methods share the candidate pool and test queries. BKRAG, GFM-RAG, and LUMI-MEM are all trained on the same split, while the remaining baselines are evaluated in their original settings. DR is pretrained and frozen; LUMI-MEM updates only the shared adapter parameters ϕ, sparse-to-dense weights, and gate MLP. At test time, ϕ remains frozen, and f_ϕ generates the four controls for each basin in the test memory bank from $\mathbf{z}_i(q)$ without basin-specific fitting. We use $T = 3$, with $(k_{\mathrm{DR}}, k_{\mathrm{basin}}) =$

Table 1: Retrieval performance across three datasets. Enhanced and memory-based retrieval methods use Contriever as the retriever. The **best** and second-best results are highlighted.

Method	LongMemEval				2Wiki				MuSiQue			
	R@3	R@5	N@3	N@5	R@3	R@5	N@3	N@5	R@3	R@5	N@3	N@5
Direct Retrieval												
BM25	64.08	71.50	57.03	57.18	59.75	65.25	67.36	67.89	38.83	44.96	44.16	46.46
Contriever	75.08	79.33	64.38	63.93	52.63	56.50	57.33	57.54	40.54	47.38	46.50	49.13
Stella	68.17	81.75	59.02	62.77	51.88	56.25	59.77	59.83	26.54	31.92	28.54	30.83
Enhanced Retrieval												
BKRAG	73.75	82.92	64.08	65.71	71.38	73.50	80.78	78.83	50.46	56.96	57.69	59.91
PBR	76.58	86.58	66.28	68.09	53.50	57.75	57.15	58.00	41.21	48.17	44.74	47.55
ThinkQE	71.58	77.67	61.02	62.67	66.00	74.25	70.99	73.24	48.67	55.88	52.00	54.93
GFM-RAG	75.17	83.83	63.11	65.55	73.75	80.63	80.40	81.12	42.54	46.42	45.89	46.71
Memory-based Retrieval												
HippoRAG-v1	41.75	52.17	36.94	40.23	56.75	72.00	55.48	62.22	47.83	53.13	54.59	56.01
HippoRAG-v2	73.75	80.58	66.85	67.99	79.63	85.25	85.24	85.21	**55.38**	62.63	59.82	62.31
ComoRAG	76.92	84.25	66.52	68.13	57.88	65.25	63.01	64.81	45.79	52.79	50.88	53.71
Lumi-Mem (ours)	**79.83**	**90.58**	**72.31**	**74.08**	**85.88**	**93.38**	**92.42**	**93.18**	54.75	**65.54**	**60.44**	**63.90**

$(60, 50)$ for LongMemEval and $(512, 256)$ for 2Wiki and MuSiQue. Reported results are averaged over five runs. Hyperparameter settings are provided in Appendix B. Experiments were conducted on a single NVIDIA RTX 3090 GPU.

5.2 Overall Performance

Retrieval performance across benchmarks is summarized in Table 1. Among the compared methods, Lumi-Mem ranks first or second across all retrieval metrics and achieves the best R@5 on all three benchmarks. Although it does not use per-query LLM reflection, it outperforms ComoRAG on all metrics. It also outperforms HippoRAG-v2 on most metrics, with the largest margins on LongMemEval. These gains also persist when Stella replaces Contriever as the retriever (Appendix C.5). Under zero-shot cross-corpus transfer, Lumi-Mem retains much of the R@5 performance achieved with target-domain supervision in the evaluated settings (Appendix C.6).

We further evaluate answer generation performance using the top-5 retrieved chunks, as shown in Table 2. Lumi-Mem ranks first on seven of the eight QA metrics and second on LongMemEval EM. Its average F1 exceeds that of the strongest baseline, ThinkQE, by 5.32 points, corresponding to an 11.7% relative gain. Lumi-Mem also leads on MuSiQue while using fewer inference-time tokens than ThinkQE (1,818.4 vs. 4,010.4), and it outperforms the full-context LLM on all six dataset-level QA metrics.

5.3 Ablation Study

Complementary Memory Sources. As shown in Table 3, both SA and DR contribute to the final retrieval performance, serving as complementary memory sources. Notably, removing SA leads to a substantial performance drop, with declines exceeding 10% on the multi-hop QA datasets. This result suggests that sparse associative indexing plays a central role in identifying the topic basins needed for accurate evidence retrieval. A matched SA–PPR comparison further indicates the benefit of SA propagation (Appendix C.2).

Sparse–Dense Interaction. With both SA and DR retained, Figure 2 examines three components of sparse–dense interaction: sparse-to-dense activation $s^{\mathrm{s2d}}(c, q)$, the adaptive gate $\mathrm{Gate}(\mathbf{f}_{qc})$, and dense-to-sparse calibration. Removing the adaptive gate disables dynamic cortical–hippocampal gain control and reduces performance on 2Wiki by more than 20%. Removing dense-to-sparse calibration eliminates the topic-level signal used to train SA, thereby weakening the posterior used by sparse-to-dense activation. Removing sparse-to-dense activation eliminates sparse associative

Table 2: QA performance across three datasets. GPT-4.1-mini is used for answer generation. For retrieval-based methods, the top-5 retrieved evidence chunks are provided as context. The **best** and second-best results are highlighted.

Method	LongMemEval		2Wiki		MuSiQue		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
Full-context LLM								
GPT-4.1-mini	23.00	38.82	22.00	26.81	6.00	14.86	17.00	26.83
Direct Retrieval								
BM25 (Robertson & Walker, 1994)	29.00	36.55	39.50	46.11	19.00	29.56	29.17	37.41
Contriever (Izacard et al., 2022)	30.00	38.49	37.50	44.44	19.00	27.56	28.83	36.83
Stella (Zhang et al., 2024)	32.00	41.69	33.00	39.82	17.00	23.03	27.33	34.85
Enhanced Retrieval								
BKRAG (Guo et al., 2024)	**37.00**	43.36	48.00	56.87	20.00	32.74	35.00	44.32
PBR (Zhang et al., 2026)	29.00	39.98	32.00	40.01	25.50	34.72	28.83	38.24
ThinkQE (Lei et al., 2025)	30.00	38.22	49.50	59.71	27.50	38.40	35.67	45.44
GFM-RAG (Luo et al., 2025)	28.00	36.20	55.00	63.60	19.50	27.67	34.17	42.49
Memory-based Systems								
Mem0 (Chhikara et al., 2025)	27.00	35.98	38.50	45.91	18.50	28.01	28.00	36.63
EverMemOS (Hu et al., 2026)	22.00	32.28	49.50	58.67	22.50	33.45	31.33	41.47
HeLa-Mem (Zhu et al., 2026)	30.00	38.98	40.00	46.60	18.00	27.23	29.33	37.60
HippoRAG-v1 (Gutiérrez et al., 2024)	18.00	24.27	45.00	52.92	21.00	27.77	28.00	34.99
HippoRAG-v2 (Gutiérrez et al., 2025)	29.00	36.23	50.50	61.34	27.00	38.20	35.50	45.26
ComoRAG (Wang et al., 2026b)	30.00	38.34	43.00	51.01	21.50	32.06	31.50	40.47
Lumi-Mem (ours)	36.00	**43.61**	**60.00**	**69.62**	**29.00**	**39.04**	**41.67**	**50.76**

evidence from the final score, reducing the ability to retrieve evidence through associative activation. Overall, all three mechanisms contribute to the retrieval performance of LUMI-MEM.

Table 3: Ablation study of LUMI-MEM components across datasets and retrievers. Full denotes the complete model; w/o SA removes SA; w/o DR removes DR. Best results within each dataset–retriever group are highlighted in bold.

Dataset	Full		w/o SA		w/o DR	
	R@5	N@5	R@5	N@5	R@5	N@5
			Contriever			
LongMemEval	**90.58**	**74.08**	86.42	70.37	87.75	69.37
2Wiki	**93.38**	**93.18**	69.38	71.71	92.38	92.07
MuSiQue	**65.54**	**63.90**	51.96	54.58	60.75	60.83
			Stella			
LongMemEval	**84.70**	**69.20**	82.75	66.84	82.33	64.74
2Wiki	**89.90**	90.20	65.75	71.13	89.88	**90.21**
MuSiQue	**64.00**	**65.90**	43.92	43.93	60.79	62.24

5.4 VISUALIZATION CASE STUDY

Figure 3 shows the topic activation produced by SA. Given a query, different topic basins are stimulated and connected through semantic and associative edges. The size of each circle represents the attractor-based topic posterior, with larger circles indicating higher activation. In this case, the question asks about the total money earned from selling products at different markets. Topics T1, T2, T3, T4, and T5 are connected by semantic links because they all contain information related to products and markets. In addition, T1, T3, and T4 are connected by associative links because they encode related personal experiences not captured by lexical overlap. Overall, T1, T2, and T3 receive higher activation scores, suggesting that they provide more useful retrieval cues for answering the query.

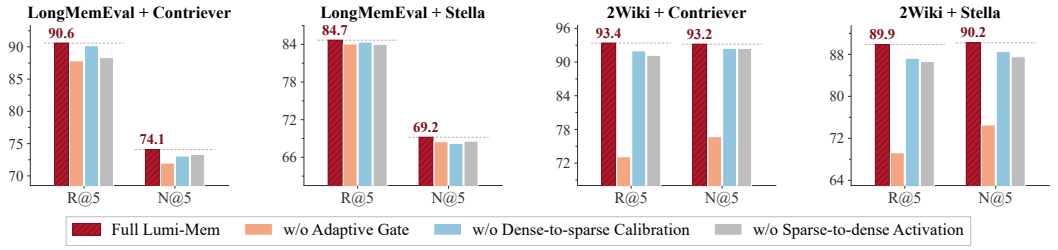

Figure 2: Ablation study of sparse–dense mechanisms with both memory backbones retained. Results are reported on LongMemEval and 2Wiki with two retrievers.

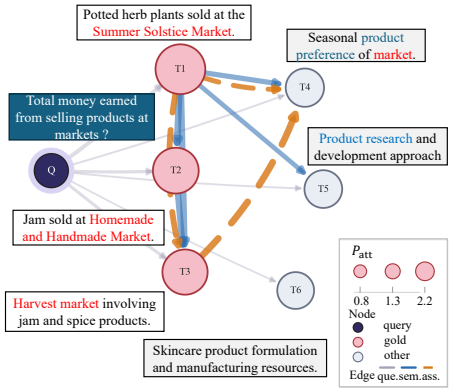

Figure 3: Case illustration of topic activation. Node size indicates the attractor-based topic posterior, and edges encode semantic and associative links.

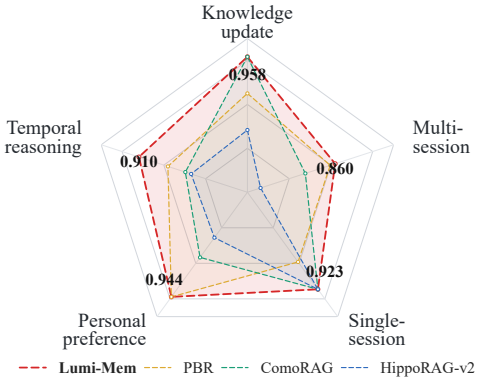

Figure 4: R@5 across LongMemEval question types, including single-session, preference-based, multi-session, temporal reasoning, and knowledge-update questions.

5.5 Analysis by Question Type

We evaluate the retrieval performance of LUMI-MEM across different LongMemEval question types. LUMI-MEM performs best among the compared methods on temporal reasoning, multi-session reasoning, and knowledge updates. ComoRAG obtains comparable performance on knowledge-update questions but performs worse on multi-session and temporal-reasoning questions. For single-session questions, LUMI-MEM, ComoRAG, and HippoRAG-v2 achieve comparable performance, while PBR shows less stable results. Overall, these results suggest that sparse–dense coordination enables LUMI-MEM to preserve retrieval accuracy on simpler questions while supporting memory updating and associative reasoning on more complex questions.

6 Conclusion

We propose LUMI-MEM, a sparse–dense memory self-organization framework for LLM agents that addresses memory retrieval when queries require multi-hop associative reasoning. LUMI-MEM coordinates sparse indexing and dense evidence representation through a bio-inspired interaction: SA indexes topic-level memory regions, while DR identifies fine-grained evidence within these regions. Experiments on LongMemEval, 2WikiMultiHopQA, and MuSiQue show consistently strong retrieval and answer-generation performance and indicate that each sparse–dense interaction module contributes to the final results. Despite these results, several limitations remain: evaluation is limited to three text-based QA benchmarks, and the memory bank is constructed offline. Future work should validate LUMI-MEM in broader, continuously updated assistant settings and extend it with online memory-bank maintenance and adaptive candidate budgeting to improve deployment robustness.

AI USE STATEMENT

Generative AI tools were used for language editing and limited assistance with code development and debugging. They were also occasionally used to organize and clarify discussions of experimental design and the interpretation of results. All AI-assisted outputs were independently reviewed and verified by the authors. The authors made all research, methodological, experimental, and interpretive decisions and take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

Sections 3–5 describe the problem formulation, the complete LUMI-MEM architecture, the training objective and inference procedure, and the benchmarks, data splits, baselines, metrics, and evaluation protocol. Appendix A specifies the model notation, component definitions, loss functions, DR pretraining, topic-graph construction, and calibration mechanism. Appendix B lists all numerical training and inference settings. Appendix C reports settling diagnostics, matched comparisons, sensitivity analyses, retriever robustness, cross-corpus transfer, and retrieval efficiency, while Appendix D documents topic-bank preprocessing. All experiments use publicly available benchmarks, and the reported effectiveness results are averaged over five runs under the shared evaluation protocol described in Section 5.

REFERENCES

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. Fast incremental and personalized pagerank. *arXiv preprint arXiv:1006.2880*, 2010.

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.

Bailu Ding and Jiaqi Zhai. Retrieval with learned similarities. In *Proceedings of the ACM on Web Conference 2025*, pp. 1626–1637, 2025.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Yifan Feng, Hao Hu, Xingliang Hou, Shiquan Liu, Shihui Ying, Shaoyi Du, Han Hu, and Yue Gao. Hyper-rag: Combating llm hallucinations using hypergraph-driven retrieval-augmented generation. *arXiv preprint arXiv:2504.08758*, 2025.

Zafeirios Fountas, Martin A Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou Ammar, and Jun Wang. Human-inspired episodic memory for infinite context llms. In *International Conference on Learning Representations*, volume 2025, pp. 77230–77268, 2025.

Jun Guo, Bojian Chen, Zhichao Zhao, Jindong He, Shichun Chen, Donglan Hu, and Hao Pan. Bkrag: A bge reranker rag for similarity analysis of power project requirements. In *Proceedings of the 2024 6th International Conference on Pattern Recognition and Intelligent Systems*, pp. 14–20, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models, 2024. NeurIPS 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models, 2025. ICML 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, et al. Evermemos: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853, 2026.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

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.

Yibin Lei, Tao Shen, and Andrew Yates. Thinkqe: Query expansion via an evolving thinking process. *arXiv preprint arXiv:2506.09260*, 2025.

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.

Zirui Liao. Ecphoryrag: Re-imagining knowledge-graph rag via human associative memory. *arXiv preprint arXiv:2510.08958*, 2025.

Haoran Luo, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, et al. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. *Advances in Neural Information Processing Systems*, 38:152206–152234, 2026.

Linhao Luo, Zicheng Zhao, Reza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. GFM-RAG: Graph foundation model for retrieval augmented generation. *Advances in Neural Information Processing Systems*, 38:36371–36405, 2025. doi: 10.52202/085713-1222.

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.

Alison R Preston and Howard Eichenbaum. Interplay of hippocampus and prefrontal cortex in memory. *Current biology*, 23(17):R764–R773, 2013.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pp. 2366–2377, 2025.

Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 232–241, Dublin, Ireland, 1994. Springer. doi: 10.1007/978-1-4471-2099-5_24.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024.

William Beecher Scoville and Brenda Milner. Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, 20(1):11, 1957.

Peiyang Song, Pengrui Han, and Noah Goodman. A survey on large language model reasoning failures. In *2nd AI for Math Workshop@ ICML 2025*, 2025.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long T. Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.413.

Timothy J Teyler and Jerry W Rudy. The hippocampal indexing theory and episodic memory: updating the index. *Hippocampus*, 17(12):1158–1169, 2007.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Endel Tulving. Episodic memory: From mind to brain. *Annual review of psychology*, 53(1):1–25, 2002.

Changjian Wang, Weihong Deng, Weili Guan, Quan Lu, and Ning Jiang. Cross-granularity hypergraph retrieval-augmented generation for multi-hop question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33368–33376, 2026a.

Juyuan Wang, Rongchen Zhao, Wei Wei, Yufeng Wang, Mo Yu, Jie Zhou, Jin Xu, and Liyan Xu. ComoRAG: A cognitive-inspired memory-organized RAG for stateful long narrative reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):33557–33565, 2026b. doi: 10.1609/aaai.v40i39.40644.

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

Yuchao Wu, Junqin Li, XingCheng Liang, Yongjie Chen, Yinghao Liang, Linyuan Mo, and Guanxian Li. Sag: Sql-retrieval augmented generation with query-time dynamic hyperedges. *arXiv preprint arXiv:2606.15971*, 2026.

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, 2026.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. Jasper and Stella: Distillation of SOTA embedding models, 2024.

Yingyi Zhang, Pengyue Jia, Derong Xu, Yi Wen, Xianneng Li, Yichao Wang, Wenlin Zhang, Xiaopeng Li, Weinan Gan, Huifeng Guo, Yong Liu, and Xiangyu Zhao. Personalize before retrieve: LLM-based personalized query expansion for user-centric retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(19):16406–16414, 2026. doi: 10.1609/aaai.v40i19.38679.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22600–22632, 2024.

Jinchang Zhu, Jindong Li, Cheng Zhang, Jiahong Liu, and Menglin Yang. Hela-mem: Hebbian learning and associative memory for llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13757–13769, 2026.

A Model and Training Details

A.1 Overview and Notation

Lumi-Mem is a sparse–dense memory self-organization framework for LLM agents. It contains a Sparse Attractor (SA) for topic-level associative indexing and a Dense Representation (DR) module for chunk-level evidence scoring. Sparse-to-dense activation transfers topic support to evidence chunks, while dense-to-sparse calibration aligns SA with dense evidence during training.

Let $\mathcal{M} = \{m_i\}_{i=1}^{K}$ be a memory bank of K topic basins. For personalized-memory data, each user has an independent memory bank, and a query accesses only its associated user's bank. Each basin m_i has an embedding e_i, a set of evidence chunks $\mathcal{C}_i$, and a confidence score $\mathrm{conf}_i \in [0, 1]$ that indicates how confidently its summary is supported by $\mathcal{C}_i$. For a query q with embedding e_q, $\mathcal{G}_q$ denotes the gold evidence set and $\mathrm{src}(c) = \{i : c \in \mathcal{C}_i\}$ denotes the source-basin index set of chunk c.

Sparse Attractor. SA uses the initialization and query-conditioned excitation in Section 4.2, followed by the damped update in Eq. 15. The resulting topic posterior drives candidate expansion and sparse-to-dense activation. Appendix C.1 gives the complete energy derivation, equilibrium condition, and numerical settling protocol.

Dense Representation. DR uses the fixed alignment matrix $\mathbf{R}$ to score candidate chunks. Eq. 3 defines the normalized score and dense posterior used by the remaining modules.

Candidate construction and fusion. Eq. 2 combines dense candidates with chunks attached to activated basins. Eqs. 16 and 17 define direct and associative support, while Eq. 4 gives their learned mixture. Eq. 8 then combines this sparse signal with DR.

A.2 Training Loss Details

For each training query, the final evidence scores define the listwise posterior $P_{\mathrm{rank}}(c \mid q) = \mathrm{softmax}(\mathrm{score}(\cdot \mid q)/\tau_{\mathrm{rank}})_c$. The evidence-ranking loss maximizes its posterior mass on gold chunks:

$$\mathcal{L}_{\mathrm{rank}} = -\log \sum_{c \in \mathcal{G}_q \cap \mathcal{C}_q} P_{\mathrm{rank}}(c \mid q). \tag{10}$$

Queries without a gold chunk in $\mathcal{C}_q$ do not contribute to this term. We regularize the adaptive gate with $\mathcal{L}_{\mathrm{gate}} = |\mathcal{C}_q|^{-1} \sum_{c \in \mathcal{C}_q} \mathrm{Gate}(\mathbf{f}_{qc})^2$.

Dense-to-sparse calibration uses the dense topic signal in Eq. 5 and the gold-topic indicator defined in Section 4.5. The normalized target formed from these signals and the corresponding KL objective are given in Eqs. 6 and 7. The complete joint objective is Eq. 9.

Algorithm 2 Dense-to-Sparse Calibration and Joint Training

Require: Query q; gold evidence $\mathcal{G}_q$; memory bank $\mathcal{M}$; pretrained DR; trainable Lumi-Mem parameters
Ensure: Updated shared adapter parameters ϕ, sparse-to-dense weights, and gate MLP
1: Compute $P_{\mathrm{att}}(\cdot \mid q)$ using the SA definitions in Section A.1
2: Construct $\mathcal{C}_q$ using Eq. 2
3: Compute DR, sparse-to-dense activation, and final evidence scores using Eqs. 3, 4, and 8
4: Compute $D_i(q)$, $G_i(q)$, and $P_{\mathrm{cal}}(i \mid q)$ using Eqs. 5 and 6
5: Compute $\gamma(q)$ using Eq. 19
6: Compute the ranking, gate-regularization, and dense-to-sparse losses
7: Backpropagate the loss in Eq. 9 and update the trainable parameters

A.3 DR Pretraining

Before joint training, DR is pretrained as a standalone dense ranker. This pretraining uses only queries and gold evidence from the training split; no held-out query is used. For each query, $\mathcal{C}_q^{\mathrm{pre}}$

is a sampled or retrieved candidate pool containing available gold chunks. Using the raw score $\mathrm{DR}_{\mathrm{raw}}(c, q)$, DR defines

$$P_{\mathrm{DR}}(c \mid q) = \mathrm{softmax}\left(\mathrm{DR}_{\mathrm{raw}}(\cdot, q)/\tau_{\mathrm{pre}}\right)_c . \tag{11}$$

The alignment matrix $\mathbf{R}$ is trained with

$$\mathcal{L}_{\mathrm{DR}} = -\log \sum_{c \in \mathcal{G}_q \cap \mathcal{C}_q^{\mathrm{pre}}} P_{\mathrm{DR}}(c \mid q). \tag{12}$$

DR is then frozen and supplies dense candidates, ranking scores, and dense-to-sparse feedback.

A.4 Topic Graph and Trainable Sparse Components

Topic graph construction. The semantic and associative graphs used in Eq. 1 are built once over the fixed topic basins within each memory bank. Both start from relation-score matrices whose zero diagonals prevent self-reinforcement. For $i \neq j$, the semantic score is $S_{\mathrm{sem},ij} = \max(e_i^\top e_j, 0)$. Associative edges link topics that co-occur within the same session or day, or that explicitly refer to one another. These relations use observable metadata inherited from the evidence chunks in each basin: $\mathrm{sess}(i)$ is the set of session identifiers represented in $\mathcal{C}_i$, $\mathrm{cre}(i)$ is the set of calendar-day identifiers on which those chunks were created, and $\mathrm{upd}(i)$ is the set of update-link identifiers assigned during memory-bank construction to explicit references or corrections involving those chunks. The associative score is

$$\begin{aligned} S_{\mathrm{assoc},ij} = {} & \omega_s \mathbb{I}[\mathrm{sess}(i) \cap \mathrm{sess}(j) \neq \varnothing] \\ & + \omega_c \mathbb{I}[\mathrm{cre}(i) \cap \mathrm{cre}(j) \neq \varnothing] \\ & + \omega_u \mathbb{I}[\mathrm{upd}(i) \cap \mathrm{upd}(j) \neq \varnothing], \qquad i \neq j. \end{aligned} \tag{13}$$

Here, $\mathbb{I}[\cdot]$ is the indicator function, while ω_s, ω_c, and ω_u weight shared-session, shared-creation-day, and shared-update links, respectively. When a metadata type is unavailable, its set is empty and the corresponding term contributes no edge. For relation $\bullet \in \{\mathrm{sem}, \mathrm{assoc}\}$, $\mathrm{TopK}_{k_\bullet}$ denotes row-wise top-k sparsification. It is the identity for the semantic graph, whereas the associative graph uses $k_{\mathrm{assoc}} = 8$. We set $\mathbf{W}_{\mathrm{raw},\bullet} = \mathrm{Sym}(\mathrm{TopK}_{k_\bullet}(\mathbf{S}_\bullet))$, with $\mathrm{Sym}(\mathbf{M}) = \max(\mathbf{M}, \mathbf{M}^\top)$ elementwise. For $\mathbf{D}_\bullet = \mathrm{Diag}(\mathbf{W}_{\mathrm{raw},\bullet}\mathbf{1})$, degree normalization gives $\widetilde{\mathbf{W}}_\bullet = \mathbf{D}_\bullet^{-1/2}\mathbf{W}_{\mathrm{raw},\bullet}\mathbf{D}_\bullet^{-1/2}$. Here, $\mathbf{1}$ is the all-ones vector, and inverse degree is set to zero for zero-degree basins.

Shared basin-conditioned adapter. For basin i and query q, the adapter input is $\mathbf{z}_i(q) = [\ell_i^0, d_i^{\mathrm{graph}}, \mathrm{conf}_i, a_i^0]^\top \in \mathbb{R}^4$. The cue logit $\ell_i^0 = e_q^\top e_i/\sqrt{d}$ measures direct query–basin similarity, while d_i^{graph} records query-independent connectivity in the fixed topic graph. The confidence score $\mathrm{conf}_i \in [0, 1]$ is assigned during memory-bank construction and records how confidently the basin summary is supported by $\mathcal{C}_i$. The bounded cue $a_i^0 = \sigma(\ell_i^0) \in (0, 1)$ represents initial activation strength. The shared affine map $f_\phi : \mathbb{R}^4 \to \mathbb{R}^4$ generates four raw adapter values:

$$[\xi_i^r(q), \xi_i^{\mathrm{sem}}(q), \xi_i^{\mathrm{assoc}}(q), \xi_i^\theta(q)]^\top = f_\phi(\mathbf{z}_i(q)) = \mathbf{W}_\phi \mathbf{z}_i(q) + \boldsymbol{\delta}_\phi, \quad \mathbf{W}_\phi \in \mathbb{R}^{4\times 4}, \quad \boldsymbol{\delta}_\phi \in \mathbb{R}^4. \tag{14}$$

Here, $\phi = (\mathbf{W}_\phi, \boldsymbol{\delta}_\phi)$ comprises the shared trainable weight matrix and offset. The outputs define $r_i(q) = 1 + \epsilon_r \tanh(\xi_i^r(q))$, $g_i^{\mathrm{sem}}(q) = 1 + \epsilon_g \tanh(\xi_i^{\mathrm{sem}}(q))$, $g_i^{\mathrm{assoc}}(q) = 1 + \epsilon_g \tanh(\xi_i^{\mathrm{assoc}}(q))$, and $\theta_i(q) = \theta_0 + \epsilon_\theta \tanh(\xi_i^\theta(q))$. η_{sem}, η_{assoc}, and θ_0 are global base values; ϵ_r, ϵ_g, and ϵ_θ bound feature-conditioned deviations. Because f_ϕ is shared, i indexes a basin input rather than a parameter slot; at test time, ϕ remains frozen and generates target-basin controls without requiring a shared K, basin alignment, or basin-specific fitting.

Damped recurrent update. To combine graph propagation with query evidence, we set $\mathbf{A}(q) = \mathbf{B}(q) - \kappa_0(q)\mathbf{I}$ and $\mathbf{b}(q) = \beta\boldsymbol{\ell}^0 - \boldsymbol{\theta}(q)$. Here, $\mathbf{I}$ is the identity matrix, β weights the residual cue, $\boldsymbol{\theta}(q) = (\theta_i(q))_{i=1}^K$, and $\kappa_0(q)$ is query-wise adaptive inhibition. Starting from $\mathbf{p}^0$, SA applies

$$\mathbf{p}^{t+1} = (1-\lambda)\mathbf{p}^t + \lambda\, \mathrm{softmax}\left(\frac{\mathbf{A}(q)\mathbf{p}^t + \mathbf{b}(q)}{\tau_{\mathrm{topic}}}\right), \tag{15}$$

where $\lambda \in (0, 1]$ is the damping coefficient, $\mathbf{p}^t$ is the topic posterior after step t, and T is the total number of recurrent steps. The output is $P_{\mathrm{att}}(i \mid q) = p_i^T$.

Sparse-to-dense activation. For a candidate chunk c, direct support aggregates attractor mass over its source basins:

$$\psi_{\mathrm{dir}}(c, q) = \log\left(\varepsilon + \sum_{i \in \mathrm{src}(c)} P_{\mathrm{att}}(i \mid q)\right), \tag{16}$$

where $\varepsilon > 0$ is a numerical-stability constant. Associative support additionally aggregates mass from basins connected to the source basins:

$$\psi_{\mathrm{assoc}}(c, q) = \log\left(\varepsilon + \sum_{i \in \mathrm{src}(c)} \sum_{j=1}^{K} (\mathbf{W}_{\mathrm{raw,assoc}})_{ij} P_{\mathrm{att}}(j \mid q)\right). \tag{17}$$

The two signals are mixed by the trainable vector $\mathbf{w}^{\mathrm{raw}}$. Softmax normalization makes the resulting weights non-negative and ensures that they sum to one, while candidate-level z-score normalization places sparse support on the same scale as DR.

A.5 Adaptive Calibration Weight

The calibration weight $\gamma(q)$ combines dense uncertainty, attractor uncertainty, and task demand. Let s_k^{DR} be the k-th highest raw DR score in $\mathcal{C}_q$ and define the top-five margin $\Delta_5^{\mathrm{DR}}(q) = s_1^{\mathrm{DR}} - s_5^{\mathrm{DR}}$. The two uncertainty measures are

$$\begin{aligned} u_{\mathrm{DR}}(q) &= \sigma\left(1 - \Delta_5^{\mathrm{DR}}(q)\right), \\ u_{\mathrm{att}}(q) &= \frac{-\sum_{i=1}^{K} P_{\mathrm{att}}(i \mid q) \log P_{\mathrm{att}}(i \mid q)}{\log K}. \end{aligned} \tag{18}$$

For each training query, the task indicator $\chi_{\mathrm{task}}(q)$ is read from dataset-provided query-type metadata. It equals one for associative or temporal queries and zero otherwise. The indicator is used only to weight $\mathcal{L}_{\mathrm{d2s}}$ during training and is not used by the inference pipeline. It does not use test query types, answers, gold evidence, or retrieval results. When query-type metadata is unavailable during training, we set $\chi_{\mathrm{task}}(q) = 0$. We compute

$$\gamma(q) = \mathrm{clip}\left(0.5 u_{\mathrm{DR}}(q) + 0.3 u_{\mathrm{att}}(q) + 0.2 \chi_{\mathrm{task}}(q), 0, 1\right). \tag{19}$$

In Eq. 7, $\mathrm{sg}(\gamma(q))$ keeps this calibration weight fixed during backpropagation.

B Training Configuration

Table 4 reports all numerical training settings. The candidate budgets k_{DR} and k_{basin} are defined in Eq. 2.

Table 4: Training configuration for Lumi-Mem. Candidate budgets are dataset specific; all attractor, adapter, and optimization settings are shared across datasets.

Setting	LongMemEval	2Wiki/ MuSiQue
Candidate construction		
DR candidates k_{DR}	60	512
Topic basins k_{basin}	50	256
Attractor dynamics		
Recurrent steps T	3	
Damping λ	0.7	
Residual cue β	0.65	
Adaptive inhibition $\kappa_0(q)$	Eq. 20	
Semantic strength η_{sem}	0.25	
Associative strength η_{assoc}	0.35	
Topic temperature τ_{topic}	1.47	
Adapter controls		
Threshold θ_0	0.0	
Adapter bound ϵ_r	0.15	
Adapter bound ϵ_g	0.10	
Adapter bound ϵ_θ	0.05	

Setting	LongMemEval	2Wiki/ MuSiQue
Optimization		
Optimizer	AdamW	
Learning rate	3×10^{-3}	
Batch size	16	
Weight decay	10^{-3}	
Maximum epochs	80	
DR pretraining τ_{pre}	0.1	
Dense temperature τ_{dense}	1.0	
Ranking temperature τ_{rank}	1.0	
Gold-topic weight α	0.8	
Calibration weight η_{d2s}	0.05	
Gate weight η_{gate}	0.01	

C Additional Results

C.1 Energy Formulation and Settling Diagnostics

Energy-compatible construction. The symmetric graph normalization and bilateral controls in Eq. 1 make $\mathbf{B}(q)$ symmetric. We use

$$\kappa_0(q) = 1.05 \max\{\lambda_{\max}(\mathbf{B}(q)), 0\} + 10^{-6}. \tag{20}$$

This gives $\mathbf{A}(q) = \mathbf{B}(q) - \kappa_0(q)\mathbf{I}$ and $\mathbf{b}(q) = \beta \boldsymbol{\ell}^0 - \boldsymbol{\theta}(q)$ for the damped update in Eq. 15.

Unique-equilibrium condition. The fixed points of the undamped map are stationary points of

$$\mathcal{E}_q(\mathbf{p}) = -\frac{1}{2}\mathbf{p}^\top \mathbf{A}(q)\mathbf{p} - \mathbf{b}(q)^\top \mathbf{p} + \tau_{\mathrm{topic}} \sum_i p_i \log p_i, \qquad \mathbf{p} \in \Delta^{K-1}. \tag{21}$$

For the energy in Eq. 21, the Lagrangian stationarity condition is $-(\mathbf{A}(q)\mathbf{p})_i - b_i(q) + \tau_{\mathrm{topic}}(\log p_i + 1) + \mu = 0$, where μ enforces the simplex constraint. This is equivalent to $\mathbf{p} = \mathrm{softmax}((\mathbf{A}(q)\mathbf{p} + \mathbf{b}(q))/\tau_{\mathrm{topic}})$, and damping does not change the fixed points. If $\kappa_0(q) > \lambda_{\max}(\mathbf{B}(q))$, then $\mathbf{A}(q)$ is negative definite and $\nabla^2 \mathcal{E}_q = -\mathbf{A}(q) + \tau_{\mathrm{topic}} \mathrm{Diag}(1/p_i)$ is positive definite. Thus, for each fixed query q, the energy is strictly convex on the simplex interior and the corresponding equilibrium is unique.

Validation-only dynamics selection. We selected λ and T on an 80-query subset of the LongMemEval training partition; the 100 test queries were not used. The grid was $\lambda \in \{1.0, 0.7, 0.5, 0.3, 0.1\}$ and $T \in \{1, 2, 3, 5, 10, 20\}$. The topic temperature followed $\tau_{\mathrm{topic}}(e) = 2.0(1 - e/80) + 0.3(e/80)$ during training and was not tuned separately. The selected checkpoint was at epoch $e = 25$, giving $\tau_{\mathrm{topic}} = 1.47$ at inference. For each validation query, we solved a reference equilibrium $\mathbf{p}^\star$ for at most 200 iterations, stopping when $\|\mathbf{p}^{t+1} - \mathbf{p}^t\|_1 < 10^{-7}$. A setting was retained only if, for every query, $\|\mathbf{p}^T - \mathbf{p}^\star\|_1 < 10^{-3}$ and the finite-step energy was non-increasing within 10^{-7}. We ranked the retained settings by R@5 and then by N@5, breaking any remaining ties by smaller T and larger λ. The selected setting was fixed for test evaluation.

Equilibrium-distance criterion. The step change in Eq. 15 scales with λ, so a small λ can satisfy a step threshold while $\mathbf{p}^T$ remains far from equilibrium. We therefore used $\|\mathbf{p}^T - \mathbf{p}^\star\|_1$ for selection and report $\|\mathbf{p}^T - \mathbf{p}^{T-1}\|_1$ only as a diagnostic.

Multi-start test. For each query in the held-out LongMemEval set, we fixed the cue, graph, adapter controls, and resulting energy system, and changed only $\mathbf{p}^0$. We used one default softmax initialization, 20 draws from $\mathrm{Dirichlet}(\mathbf{1})$, and 20 structured initializations obtained by reassigning the default top-10 probability masses. The solver ran for at most 200 iterations and stopped when $\|\mathbf{p}^{t+1} - \mathbf{p}^t\|_1 < 10^{-7}$. All $4{,}100$ runs met this tolerance, and the maximum discrepancy from the default endpoint was 1.75×10^{-6}. These finite multi-start results numerically check initialization consistency for the evaluated queries; they do not establish global convergence.

Table 5: Numerical diagnostics for the validation-selected configuration $(\lambda, T) = (0.7, 3)$ on LongMemEval with Contriever. The reference equilibrium $\mathbf{p}^\star$ was solved independently to a step tolerance of 10^{-7}.

Diagnostic	Validation (80)	Held-out test (100)
$\|\mathbf{p}^T - \mathbf{p}^\star\|_1 < 10^{-3}$	80/80	100/100
Mean distance to $\mathbf{p}^\star$	1.90×10^{-4}	1.79×10^{-4}
Maximum distance to $\mathbf{p}^\star$	7.40×10^{-4}	7.55×10^{-4}
Non-increasing finite-step energy	80/80	100/100
Final step-update magnitude (mean)	–	6.52×10^{-4}
Final step-update magnitude (maximum)	–	8.97×10^{-4}
Multi-start termination	–	4,100/4,100
Maximum endpoint discrepancy	–	1.75×10^{-6}

C.2 Matched Comparison of SA and PPR Propagation

To isolate graph propagation, we compare SA with Personalized PageRank (PPR), the propagation mechanism used in HippoRAG (Bahmani et al., 2010; Gutiérrez et al., 2024). Table 6 compares the two variants with the same candidate chunks, graph inputs, downstream scores, sparse-to-dense projection, and fusion architecture.

Table 6: Matched comparison of SA and PPR propagation on LongMemEval. Supervised values are means across five seeds, whereas label-free variants are deterministic. Differences are SA minus PPR; positive values favor SA.

Setting	Method	R@5	ΔR@5	N@5	ΔN@5
Label-free	PPR	83.33	–	66.37	–
	SA	84.75	+1.42	68.06	+1.69
Supervised	PPR	85.28	–	72.02	–
	SA	**90.58**	**+5.30**	**74.08**	**+2.06**

In the supervised setting, SA improved R@5 from 85.28 to 90.58 and N@5 from 72.02 to 74.08. Using 1,000 paired bootstrap resamples over 100 test questions, the 95% CIs for these gains were $[1.40, 9.35]$ and $[0.25, 3.25]$; predictions were averaged across seeds before resampling. In the label-free setting, the R@5 and N@5 gains were 1.42 and 1.69 points, with CIs of $[0.83, 4.25]$ and $[0.23, 3.92]$.

C.3 Sensitivity to the Target Mixing Coefficient

Table 7: Sensitivity to the target mixing coefficient α on LongMemEval, 2Wiki, and MuSiQue with Contriever.

Dataset	Metric	α					
		0.0	0.2	0.4	0.6	0.8	1.0
LongMemEval	R@3	78.27	78.45	78.91	79.65	79.83	**79.90**
	N@3	71.21	71.56	71.82	72.14	72.31	**72.45**
2Wiki	R@3	83.50	84.62	85.31	85.71	85.88	**86.12**
	N@3	90.10	91.02	91.65	92.08	92.42	**92.68**
MuSiQue	R@3	53.50	54.14	54.48	54.65	54.75	**54.92**
	N@3	59.21	59.72	60.03	60.28	60.44	**60.58**

R@3 and N@3 increase monotonically with α on all three datasets. Over $\alpha \in [0.6, 1.0]$, the maximum change is 0.41 R@3 and 0.60 N@3 points. With a dense-only calibration target ($\alpha = 0$), scores are 0.63–2.62 points below those at $\alpha = 1$.

C.4 Sparse-to-Dense Illustration

Figure 5 illustrates how sparse topic activation complements direct query–chunk matching in two representative retrieval cases. For visualization, principal component analysis (PCA) projects the embeddings of the query, activated topics, gold chunks, and distractor chunks into two dimensions (Abdi & Williams, 2010). The projection provides a qualitative view of their relative geometry but does not determine retrieval order. All SA, DR, and fusion scores are computed in the original embedding space.

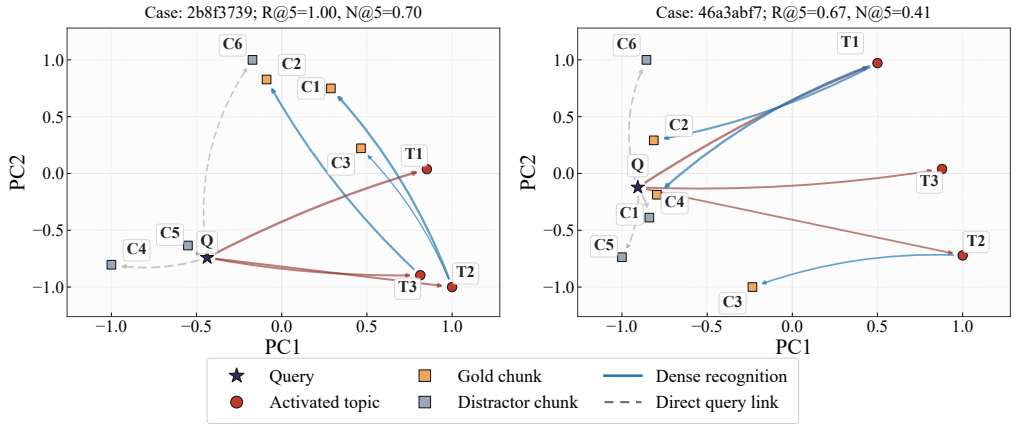

Figure 5: **Case illustration of sparse-to-dense activation for two retrieval queries.** Stars, red circles, orange squares, and gray squares denote queries, activated topics, gold chunks, and distractor chunks, respectively. Solid red arrows trace query-to-topic activation, solid blue arrows trace topic-to-chunk recognition links, and dashed gray arrows show direct query-to-chunk links. Each panel uses a two-dimensional PCA projection for visualization only; retrieval scores are computed in the original embedding space. The left and right cases achieve R@5/N@5 of 1.00/0.70 and 0.67/0.41, respectively.

In both panels, some distractors lie closer to the query than gold chunks in the PCA projection, while activated topics connect to gold chunks. The fused ranker retrieves three of three gold chunks in the left case and two of three in the right. These examples illustrate how topic activation can supplement direct dense similarity; they do not establish a general performance effect.

C.5 Robustness to the Retriever Backbone

We evaluate retriever robustness on LongMemEval, 2WikiMultiHopQA, and MuSiQue (Wu et al., 2024; Ho et al., 2020; Trivedi et al., 2022). Stella replaces the default Contriever backbone, while all settings in Table 4 remain fixed (Zhang et al., 2024; Izacard et al., 2022). Retrieval is measured by Recall@5 (R@5) and NDCG@5 (N@5); answer quality is measured by Exact Match (EM) and token-level F1. All methods use GPT-4.1-mini for answer generation from the top five retrieved chunks.

The comparison includes PBR, ThinkQE, and RMM as enhanced RAG baselines (Zhang et al., 2026; Lei et al., 2025; Tan et al., 2025), and HippoRAG-v1, HippoRAG-v2, and ComoRAG as bio-inspired baselines (Gutiérrez et al., 2024; 2025; Wang et al., 2026b).

Table 8: Retrieval and QA performance with Stella as the dense retriever. The best and second-best results are bold and underlined, respectively.

Method	LongMemEval				2Wiki				MuSiQue			
	R@5	N@5	EM	F1	R@5	N@5	EM	F1	R@5	N@5	EM	F1
Enhanced Retrieval												
PBR	<u>84.30</u>	**69.50**	31.00	40.90	67.80	66.80	48.00	57.10	38.40	37.60	24.50	34.20
ThinkQE	80.80	64.00	<u>32.00</u>	40.20	64.00	64.90	40.00	47.80	42.10	41.20	<u>26.50</u>	<u>36.10</u>
RMM	83.40	68.80	**35.00**	<u>42.80</u>	65.80	71.10	43.50	50.40	43.90	43.90	24.50	34.00
Memory-based Retrieval												
HippoRAG-v1	48.90	36.70	19.00	25.20	68.30	59.10	41.50	49.20	<u>52.80</u>	<u>55.90</u>	20.00	27.70
HippoRAG-v2	77.40	65.80	28.00	38.20	<u>81.90</u>	<u>83.10</u>	<u>48.50</u>	<u>58.20</u>	47.40	46.90	22.00	32.50
ComoRAG	81.70	64.50	29.00	37.90	68.50	69.60	43.50	52.10	38.90	37.50	20.00	28.20
Lumi-Mem (ours)	**84.70**	<u>69.20</u>	**35.00**	**45.60**	**89.90**	**90.20**	**56.00**	**65.90**	**64.00**	**65.90**	**27.00**	**38.00**

With Stella, Lumi-Mem ranks first on 11 of 12 metrics and second on LongMemEval N@5.

C.6 Zero-Shot Cross-Corpus Transfer

We test zero-shot (ZS) transfer in both directions between 2WikiMultiHopQA (2Wiki) and MuSiQue using Contriever and Stella. *Shared-Source-ZS* transfers the source-trained ϕ and gate MLP while keeping the target DR module; the shared adapter acts on each target basin without requiring basin correspondence. *Strict-Full-Source-ZS* also transfers the source DR module. Neither setting uses target evidence labels. We define retention as R@5 relative to the target-domain *Full* result in Table 3 for the same retriever.

Table 9: Zero-shot cross-corpus transfer across retrieval backbones. R@5 and N@5 are reported as percentages. Retention is computed from the displayed R@5 relative to the target-domain *Full* R@5 in Table 3 for the same retriever.

Direction	Retriever	Shared-Source-ZS		Retention (%)	Strict-Full-Source-ZS		Retention (%)
		R@5	N@5		R@5	N@5	
MuSiQue → 2Wiki	Contriever	88.00	84.85	94.24	85.25	83.17	91.29
	Stella	83.75	85.66	93.16	82.25	83.64	91.49
2Wiki → MuSiQue	Contriever	62.88	62.46	95.94	60.54	60.84	92.37
	Stella	62.29	62.69	97.33	56.08	58.23	87.63

Across the four settings, *Shared-Source-ZS* retained 93.16–97.33% of the target-domain *Full* R@5, compared with 87.63–92.37% for *Strict-Full-Source-ZS*.

C.7 RETRIEVAL EFFICIENCY

We measure online retrieval with corpus, topic, and query representations cached, excluding offline memory construction and answer generation. Tests use one CPU thread on an Intel Core Ultra 7 155H with PyTorch 2.10. Candidate-construction latency is the median of three full passes. Timings for SA with fusion and for top-5 selection are averaged over 10 post-warm-up runs; p95 is shown in parentheses.

Table 10: Single-thread CPU retrieval latency with cached representations. Setup times are in seconds; all other values are milliseconds per query.

Dataset	Queries	Corpus chunks	Setup (s)	Candidate + structure	SA + fusion mean (p95)	Top-5 selection	Total
LongMemEval	100	–	2.63	253.98	2.25 (3.06)	0.013	256.25
2WikiMultiHopQA	200	6,120	11.26	103.28	5.07 (6.10)	0.018	108.37
MuSiQue	200	11,656	27.57	167.01	5.11 (6.14)	0.018	172.14

Total retrieval latency is 108.37–256.25 ms per query. SA + fusion takes 2.25–5.11 ms (p95 $\leq$ 6.14 ms), accounting for 0.88–4.68% of the total. Candidate construction and structural processing dominate runtime. LongMemEval uses per-user memory banks and has 124.94 candidates on average (p95: 132); therefore, no global corpus size is reported.

D LLM-BASED TOPIC-BANK PREPROCESSING

For temporally ordered episodic collections, LUMI-MEM builds one topic bank per user or collection before training. Records retain natural boundaries such as dialogue sessions or document segments. Bank construction uses record content, order, and timestamps, but not downstream queries, answers, or gold evidence.

Session-level extraction and evidence chunks. Each record is serialized chronologically with local source identifiers. An instruction-following LLM, using deterministic decoding where supported, returns memory items with a summary, source identifiers, an evidence excerpt, and a confidence score. Empty, invalid, generic, or unsupported items are discarded. Each retained item creates an evidence chunk; adjacent source units are grouped only when they describe the same topic (Prompt A).

Incremental topic consolidation. Items are processed sequentially: the first creates a basin, and each later summary retrieves a shortlist of similar basins. A second LLM call adds a basin or merges the item into an existing one. A merge keeps the basin identity, appends the evidence chunk, and updates the summary and confidence score conf_i. Dates, corrections, and preference changes are retained; invalid outputs create a new basin (Prompt B).

Vectorization and reuse. After consolidation, the retriever encoder embeds topic summaries and evidence chunks. The banks are frozen and cached for training and evaluation. A different encoder can re-embed the same content without repeating extraction or changing topic membership.

Prompt A: session-level memory extraction

Role. Build a long-term topic memory bank from one temporally bounded record.
Task. Extract concise, semantically coherent memory items that could help answer future questions about record-specific facts, preferences, events, plans, relationships, or updates.
Required fields. For each item, return a retrieval summary, valid source identifiers, a minimal supporting excerpt, and a confidence value in $[0, 1]$ indicating how confidently the summary is supported by the cited evidence.
Constraints. Group adjacent units only when they concern the same topic; prefer specific grounded information over generic advice; do not introduce information absent from the record.

Output. Return one valid JSON object under `extracted_memories`; return an empty list when no useful memory is present.

Prompt B: add-or-merge topic update

Role. Update an existing topic memory bank with one newly extracted memory item.
Input. A shortlist of existing topic summaries and confidence values, and one new summary and confidence value from the same memory scope.
Decision. Return `add` when no existing summary describes the same underlying fact, preference, event, plan, relationship, or update. Otherwise, return `merge`, the selected shortlist index, and a revised summary.
Constraints. Preserve earlier information while incorporating the new evidence. Retain temporal qualifiers, corrections, and changed states or preferences. Do not merge items solely because they share broad vocabulary.
Output. Return a single valid JSON object containing the decision, the selected shortlist index when merging, the resulting summary, and a confidence value in $[0, 1]$ for that summary, with no surrounding prose.