FAST-WRITE, DEEP-READ: ECPHORYRAG AS DYNAMIC ASSOCIATIVE MEMORY FOR LIFELONG AGENTS

Zirui Liao & Zhengxian Wu & Zhuohong Chen & Xiaoyu Liu & Yifan Xu & Yunyao Yu
& Haoqian Wang
Shenzhen International Graduate School
Tsinghua University
Shenzhen, Guangdong 518055, China
{liao-zr24}@mails.tsinghua.edu.cn

ABSTRACT

Effective long-term memory is the cornerstone of autonomous agents capable of complex reasoning over extended horizons. However, current retrieval-augmented generation (RAG) systems face a critical trade-off: they are either computationally efficient but logically shallow (dense retrieval), or structurally rich but prohibitively expensive to update (knowledge graphs). Inspired by the cognitive neuroscience of memory *ecphory*—where specific cues trigger the reconstruction of associative traces—we introduce **EcphoryRAG**, a neuro-symbolic memory architecture for agents. Unlike static graph methods that require heavy pre-computation, EcphoryRAG employs a "fast-write" mechanism, abstracting raw experiences into lightweight *engrams* (entity-centric traces) with minimal latency. During retrieval, it utilizes a dynamic, centroid-based spreading activation algorithm to traverse implicit associations in vector space, simulating the brain's ability to recall multi-hop narratives from sparse signals. Extensive evaluations on the 2WikiMultiHop, HotpotQA, and MuSiQue benchmarks demonstrate that EcphoryRAG establishes a new state-of-the-art, improving the average Exact Match (EM) score to 0.475 while reducing memory consolidation costs by **18x** compared to leading graph-based baselines like HippoRAG2. These results validate EcphoryRAG as a scalable, high-fidelity cognitive substrate for the next generation of lifelong learning agents.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation. However, their reliance on static parametric knowledge remains a fundamental limitation. These models often struggle with information obsolescence and are prone to hallucinations when queried about facts outside their training data. Consequently, Retrieval-Augmented Generation (RAG) has become the standard paradigm for grounding AI agents in external, verifiable sources (Gao et al., 2024). By decoupling knowledge storage from reasoning, RAG allows models to access the most up-to-date information without the need for constant retraining.

Despite its success, standard RAG faces a critical bottleneck when handling complex, multi-hop reasoning. Most existing systems treat external memory as a flat collection of isolated vector embeddings. This lack of relational structure makes it difficult to connect disparate facts across multiple documents. While Knowledge Graph (KG) augmented methods (Edge et al., 2024; Jimenez Gutierrez et al., 2024) introduce explicit structure to solve this, they incur prohibitive computational costs. Constructing a comprehensive global graph is a slow, resource-intensive process that requires massive token consumption. This "memory consolidation latency" prevents agents from rapidly encoding new experiences in real-time. There is a clear need for a memory substrate that supports "fast-write" indexing while enabling "deep-read" relational reasoning.

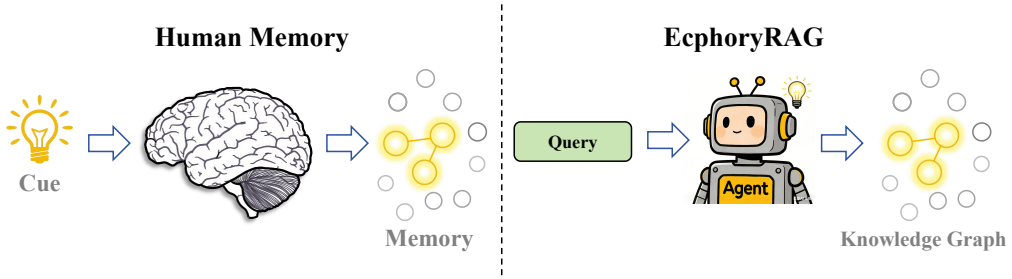

Figure 1: Neuro-symbolic alignment between biological and artificial memory. **(Left) Biological Ecphory:** A sparse cue (e.g., a sensation or keyword) triggers the spreading activation of a specific memory trace (engram) within the neural network, reconstructing a complete memory. **(Right) EcphoryRAG:** We operationalize this mechanism for agents. A user query acts as a stimulus to dynamically activate a targeted subgraph within the agent's long-term associative memory, enabling precise recall without exhaustive search.

Cognitive science provides a compelling blueprint for resolving this trade-off. Human memory does not function like a brute-force database. Instead, it operates through a reconstructive process known as **ecphory** (Tulving, 1984). In this mechanism, a sparse retrieval cue—such as a specific keyword or concept—triggers the activation of a localized network of associated memory traces, or *engrams* (Poo et al., 2016). This allows the brain to retrieve complex, multi-step narratives by activating only the relevant subset of its vast knowledge base, rather than traversing every possible connection.

Inspired by this biological principle, we introduce **EcphoryRAG**, a neuro-symbolic memory architecture designed for autonomous agents. As shown in Figure 1, EcphoryRAG operationalizes memory as a dynamic, cue-driven process. During the indexing phase, the system abstracts raw text into lightweight entity-centric engrams. This approach is highly efficient, reducing token consumption by up to 94% compared to traditional structured RAG systems. During retrieval, the agent uses the query as a stimulus to activate a targeted subgraph, propagating signals to uncover latent associations.

Our contributions are threefold:

1. We propose EcphoryRAG, a framework that mimics the cognitive process of cued recall to enable precise multi-hop reasoning.
2. We introduce a "fast-write, deep-read" architecture that reduces memory consolidation costs by 18x while maintaining high retrieval fidelity.
3. We demonstrate that EcphoryRAG sets a new state-of-the-art on 2WikiMultiHop, HotpotQA, and MuSiQue benchmarks, outperforming both iterative prompting and static graph methods.

2 Related Work

Our work sits at the intersection of retrieval-augmented generation, autonomous agent architectures, and neuro-symbolic memory systems. We review the progression from static indexing to dynamic, associative memory mechanisms.

2.1 From Static Retrieval to Agentic Reasoning Loops

Early augmentation strategies, such as Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) and standard RAG (Gao et al., 2024), treated memory as a static, flat collection of vectors. While efficient for single-hop factoid queries, these systems lack the relational structure necessary for complex reasoning. To address this, recent agentic frameworks have adopted iterative paradigms. Methods like ReAct (Yao et al., 2022), IRCoT (Trivedi et al., 2023), and FLARE (Jiang et al., 2023) simulate reasoning by decomposing complex goals into sequential retrieval steps. While these "loop-based" agents improve multi-hop performance, they suffer from high latency and error propagation, as they

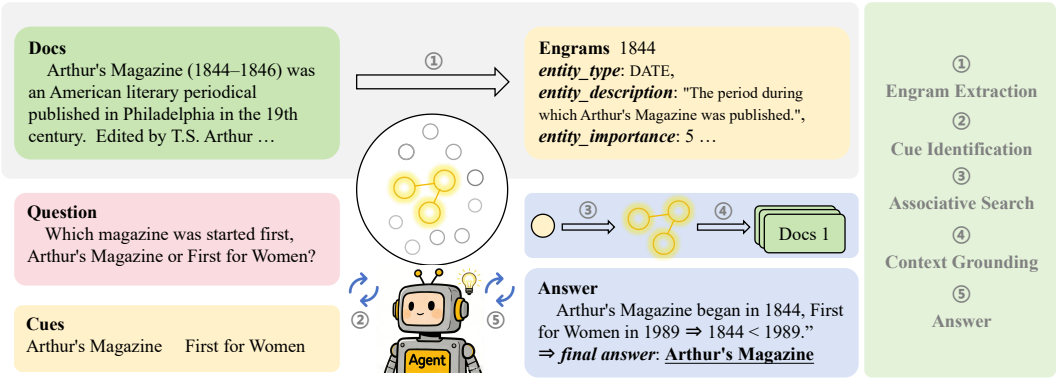

Figure 2: The cognitive lifecycle of an EcphoryRAG agent, comprised of two distinct phases: **(Top) Memory Consolidation (Fast Write):** Raw unstructured experiences (Docs) are rapidly abstracted into structured *Engrams* (entities with metadata) and assimilated into the long-term memory graph. **(Bottom) Associative Recall (Deep Read):** When formulated with a goal (Question), the agent ② identifies retrieval cues to ③ trigger a multi-hop spreading activation process. This reconstructs a reasoning path ④ grounded in specific source chunks, allowing the agent to ⑤ synthesize a faithful answer. This architecture balances low-latency encoding with high-fidelity reasoning.

lack a persistent, structured memory state to ground their intermediate reasoning. They effectively "think" without a "map," relying solely on the LLM's transient context window.

2.2 Structured Memory and Knowledge Graphs

To provide a more robust cognitive substrate, Knowledge Graph-Augmented RAG (KG-RAG) systems explicitly model entities and relationships. Approaches such as QA-GNN (Yasunaga et al., 2021) and GraphRAG (Edge et al., 2024) leverage pre-computed structures to enhance context. However, these systems often face a rigid trade-off between indexing cost and retrieval flexibility. Static graph methods require computationally expensive pre-processing (clustering, summarization), making them ill-suited for lifelong learning agents that must rapidly consolidate new experiences. Conversely, dynamic traversal methods like Think-on-Graph (Sun et al., 2024) allow for flexible exploration but incur prohibitive token costs during the online reasoning phase. There remains a critical need for a memory architecture that combines the low-latency consolidation of vector stores with the deep reasoning capabilities of knowledge graphs.

2.3 Neuro-Symbolic and Associative Memory

A growing body of research draws inspiration from cognitive neuroscience to resolve the plasticity-stability dilemma in AI agents. Human memory is not a flat database but a reconstructive process known as *ecphory*, where sparse cues trigger the activation of complex memory traces (engrams) (Tulving, 1984; Poo et al., 2016). Recent works like HippoRAG (Gutiérrez et al., 2025) have begun to operationalize these concepts using Personalized PageRank to simulate the spreading activation of the hippocampus. EcphoryRAG advances this neuro-symbolic paradigm by introducing a more dynamic, cue-centric retrieval mechanism. Unlike global graph algorithms, our approach focuses on the localized, associative reconstruction of memory, enabling agents to "recall" complex narratives from minimal cues with significantly higher computational efficiency.

3 The EcphoryRAG Framework

We propose **EcphoryRAG**, a retrieval architecture designed to simulate the cognitive process of *ecphory*—where a retrieval cue triggers the reconstruction of associated memory traces (Tulving, 1984). Unlike traditional Knowledge Graph RAG (KG-RAG) systems that rely on explicit edge traversal over hard-coded relations, EcphoryRAG operates on an **implicit co-occurrence graph**

constructed within a high-dimensional vector space. As shown in Figure 2, the framework comprises two phases: **Memory Consolidation** (Indexing) and **Ecphoric Retrieval** (Querying).

3.1 Phase I: Memory Consolidation via Engram Abstraction

The consolidation phase maps unstructured documents into a structured memory system $\mathcal{M}$. We formally define this structure not as a traditional semantic KG, but as a **Bipartite Entity-Chunk Graph** $\mathcal{G} = (\mathcal{V}_E \cup \mathcal{V}_C, \mathcal{E}_{prov})$, where entities $\mathcal{V}_E$ are linked to their source text chunks $\mathcal{V}_C$ via provenance edges $\mathcal{E}_{prov}$.

Episodic Window Definition. We first partition the corpus $\mathcal{D}$ into discrete "episodic windows" (text chunks) $\mathcal{C}$. To maintain narrative coherence, we employ a sliding window strategy with a fixed size of $L = 1200$ characters and an overlap of $O = 200$ characters. These windows define the boundaries for entity co-occurrence.

Metadata-Rich Engram Encoding. For each window $c \in \mathcal{C}$, we extract a set of entities (engrams) $\mathcal{E}_c$. To address the ambiguity of bare entity names, we enrich each engram e with metadata before embedding:

$$\text{text}_e = \text{name}_e \oplus \text{[TYPE]type}_e \oplus \text{[DESC]desc}_e \tag{1}$$

where $\oplus$ denotes concatenation. The metadata (e.g., importance score, type) is thus encoded directly into the semantic vector $\mathbf{v}_e = \text{Embed}(\text{text}_e)$. This ensures that the vector space topology reflects not just lexical similarity, but functional and descriptive proximity.

Dual-Store Indexing. We index the memory using two parallel structures:

1. **Entity Trace Index ($\mathcal{I}_E$):** A global ANN index storing $\{\mathbf{v}_e \mid e \in \mathcal{V}_E\}$. This allows for $O(1)$ access to any concept in the memory space.
2. **Chunk Index ($\mathcal{I}_C$):** A parallel index storing embeddings of the raw text windows.

Crucially, while we do not explicitly store edges between entities (e.g., $e_i \leftrightarrow e_j$), their relationship is **implicitly modeled** by their proximity in the vector space and their shared provenance to the same episodic windows.

3.2 Phase II: Ecphoric Retrieval via Centroid-Based Spreading

Retrieval in EcphoryRAG is not a graph walk in the traditional sense (traversing adjacency lists), but rather an iterative navigation of the semantic vector space. We term this "Centroid-Based Spreading Activation."

Step 1: Cue Extraction & Primary Ecphory (Cue Activation). Given the user query q, we first extract a concise *cue* $\mathbf{c}_q$ that captures the core retrieval intent using $\text{ExtractCue}(q)$. This cue is then embedded into a vector $\mathbf{v}_c$, and we perform a global ANN search on $\mathcal{I}_E$ to retrieve the top-k_{init} engrams. This set, $\mathcal{S}_0$, represents the "direct hits"—concepts explicitly mentioned or strictly synonymous with the extracted cue rather than the raw query.

Step 2: Associative Spreading via Dynamic Centroids. To discover latent multi-hop connections without relying on sparse explicit edges, we perform an iterative expansion. In each iteration t:

1. **Focus Shift:** We calculate the **Weighted Centroid** $\mathbf{c}_t$ of the currently active memory traces $\mathcal{S}_{t-1}$. The weights are determined by the relevance of each trace to the *original query*:

$$\mathbf{c}_t = \sum_{e \in \mathcal{S}_{t-1}} \frac{\exp(\mathbf{v}_e \cdot \mathbf{v}_q)}{\sum_{e'} \exp(\mathbf{v}_{e'} \cdot \mathbf{v}_q)} \mathbf{v}_e \tag{2}$$

2. **Global Associative Search:** We use this centroid $\mathbf{c}_t$ as a *new search query* to perform a global retrieval on $\mathcal{I}_E$:

$$\mathcal{S}_{new} = \text{Search}(\mathcal{I}_E, \mathbf{c}_t, k_{expand}) \tag{3}$$

Clarification on Graph Usage: This step functionally simulates a graph traversal. By moving the query vector to the centroid of retrieved entities (e.g., the centroid of "Apple" and "1984" might be near "Macintosh"), we "hop" to semantically related regions of the vector space. This effectively

Algorithm 1 EcphoryRAG Retrieval Process

Require: Query q, Indices $\mathcal{I}_E, \mathcal{I}_C$, Params k_{init}, D
Ensure: Answer a
$\mathbf{c}_q \leftarrow \text{ExtractCue}(q)$ {Cue Extraction from Query}
$\mathbf{v}_c \leftarrow \text{Embed}(\mathbf{c}_q)$
$\mathcal{S}_0 \leftarrow \text{Search}(\mathcal{I}_E, \mathbf{v}_c, k_{init})$ {Global ANN Search with Cue}
$\mathcal{S}_{all} \leftarrow \mathcal{S}_0$
for $t = 1$ to D **do**
  {Iterative Vector Space Navigation}
  $\mathcal{S}_{seed} \leftarrow \text{TopK}(\mathcal{S}_{all}, 10, \text{by} = \mathbf{v}_e \cdot \mathbf{v}_c)$
  $\mathbf{c}_t \leftarrow \text{WeightedCentroid}(\mathcal{S}_{seed}, \text{weights} \propto \text{Sim}(\cdot, \mathbf{c}_q))$
  $\mathcal{S}_{new} \leftarrow \text{Search}(\mathcal{I}_E, \mathbf{c}_t, 3 \times k_{init})$ {Global Search with Shifted Focus}
  $\mathcal{S}_{new} \leftarrow \mathcal{S}_{new} \setminus \mathcal{S}_{all}$ {Deduplicate}
  $\mathcal{S}_{all} \leftarrow \mathcal{S}_{all} \cup \mathcal{S}_{new}$
end for
{Final Re-ranking & Grounding}
for $e \in \mathcal{S}_{all}$ **do**
  $e.\text{score} \leftarrow \text{CosineSim}(\mathbf{v}_e, \mathbf{v}_c)$ {Rank by cue intent}
end for
$\mathcal{S}_{final} \leftarrow \text{TopK}(\mathcal{S}_{all}, k_{final})$
$\mathcal{C}_{final} \leftarrow \text{GetSourceChunks}(\mathcal{S}_{final}) \cup \text{Search}(\mathcal{I}_C, \mathbf{v}_c, 5)$
$a \leftarrow \text{LLM}(\text{Prompt}(q, \mathcal{S}_{final}, \mathcal{C}_{final}))$
return a

traverses the **implicit correlations** encoded in the embeddings, avoiding the noise and sparsity issues of hard-coded knowledge graph edges.

Step 3: Context Grounding and Re-ranking. After D iterations, the accumulated set of entities $\mathcal{S}_{all}$ is re-ranked. The final score for an entity e is strictly its cosine similarity to the *original* query: $\text{score}(e) = \mathbf{v}_e \cdot \mathbf{v}_q$. Finally, we utilize the explicit bipartite graph structure ($\mathcal{E}_{prov}$) to retrieve the source text chunks for the top-k_{final} entities. These chunks form the context for the LLM generation.

4 EXPERIMENTS

Our empirical evaluation focuses on validating EcphoryRAG as a robust cognitive substrate for autonomous agents. We assess the framework's capacity for complex multi-hop reasoning (Memory Fidelity) and analyze its operational efficiency (Cognitive Load), specifically isolating the costs of memory consolidation versus active recall.

4.1 EXPERIMENTAL SETUP

To ensure a rigorous assessment, we standardize the backbone models where applicable. EcphoryRAG utilizes **Phi-4** (Abdin et al., 2024) as the reasoning core for both engram extraction and answer generation, and **bge-m3** (Xiao et al., 2023) for semantic embedding. Comprehensive hyperparameter settings, system constraints, and the exact prompt templates used for these tasks are detailed in **Appendix A** and **Appendix B**.

Benchmarks. We employ three datasets representing a spectrum of reasoning complexity: **2WikiMultiHopQA** (Ho et al., 2020) for bridging concepts across distinct entities; **HotpotQA** (Yang et al., 2018) (distractor setting) for testing noise resilience; and **MuSiQue** (Trivedi et al., 2022) for assessing deep, long-range dependency resolution (2-4 hops).

Baselines. We compare EcphoryRAG against a diverse suite of retrieval paradigms: (1) **Standard RAG** (Gao et al., 2024), representing static, unstructured retrieval; (2) **Iterative & Agentic Methods**, including **FLARE** (Jiang et al., 2023), **Self-RAG** (Asai et al., 2024), and **IRCoT** (Trivedi et al., 2023), which simulate agentic loops through iterative generation and retrieval; (3) **Graph-Based Memory Systems**, specifically **LightRAG** (Guo et al., 2025) and the neuro-symbolic method **Hip-**

Table 1: Performance comparison on multi-hop QA benchmarks. **Avg.** denotes the average score across all three datasets. Best results are **bolded**, and second-best are underlined. All metrics are reported on a 0-1 scale.

Method	2WikiMultiHop		HotpotQA		MuSiQue		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
Standard RAG (Gao et al., 2024)	0.185	0.210	0.284	0.353	0.121	0.158	0.197	0.240
Iterative & Agentic Methods								
FLARE (Jiang et al., 2023)	0.225	0.339	0.280	0.558	0.134	0.207	0.213	0.368
Self-RAG (Asai et al., 2024)	0.213	0.251	0.296	0.382	0.145	0.194	0.218	0.276
IRCoT (Trivedi et al., 2023)	0.284	0.324	0.333	0.415	0.182	0.246	0.266	0.328
Graph-Based Memory								
LightRAG (Guo et al., 2025)	0.130	0.141	0.210	0.233	0.045	0.090	0.128	0.155
HippoRAG (Jimenez Gutierrez et al., 2024)	0.404	**0.520**	0.580	0.716	0.186	0.362	0.390	0.533
EcphoryRAG (Ours)	**0.406**	0.454	**0.722**	**0.814**	**0.295**	**0.369**	**0.475**	**0.547**

poRAG (Jimenez Gutierrez et al., 2024). Baseline results are derived from their official repositories using optimal configurations.

Metrics Definition. To address potential ambiguity in scaling:

- **Performance:** Exact Match (EM) and F1 scores are reported as decimal fractions in the range $[0.0, 1.0]$.
- **Efficiency:** *Indexing Tokens (IT)* is the sum total of tokens processed to construct the memory for the entire corpus. *Querying Tokens (QT)* is the average number of tokens processed per query (including prompts and retrieved contexts) during inference.

4.2 Main Results: Reasoning Capabilities

Table 1 presents the comparative performance across all benchmarks. EcphoryRAG establishes a new state-of-the-art, achieving the highest Average Exact Match (0.475) and F1 Score (0.547), significantly outperforming both iterative agentic frameworks and existing graph-based methods.

Superiority over Iterative Agents. A key finding is the limitation of iterative retrieval methods (e.g., FLARE, IRCoT) in complex reasoning scenarios. While these methods improve over standard RAG by dynamically querying external sources, they often suffer from error propagation where an incorrect intermediate step derails the entire reasoning chain. For instance, on HotpotQA, EcphoryRAG surpasses IRCoT by a substantial margin (+0.399 F1). This indicates that our "associative memory" approach—which retrieves the entire reasoning path via signal propagation in the engram graph—is more robust than the step-by-step query generation used in current agentic workflows.

Advantage over Static Graphs. Compared to HippoRAG, which relies on Personalized PageRank for retrieval, EcphoryRAG demonstrates superior precision, particularly on the highly challenging HotpotQA and MuSiQue datasets. By utilizing the query to trigger a specific, localized activation of engrams (Ecphory) rather than a global graph traversal, our method effectively filters noise. The performance gap is most evident in HotpotQA (+14.2% EM), validating the hypothesis that cue-driven retrieval is critical for handling distractor-heavy environments.

4.3 Efficiency Analysis: The Cost of Memory Consolidation

For autonomous agents operating in dynamic environments, the cost of updating memory (Consolidation) is as critical as the accuracy of retrieval. We analyze the computational overhead in Table 2.

EcphoryRAG exhibits a highly efficient memory consolidation process, requiring only 2.0M tokens for indexing—**3.3x more efficient than HippoRAG** and **18x more efficient than LightRAG**. This efficiency is achieved by our streamlined engram extraction pipeline, which avoids the redundant, multi-pass processing common in other graph-based approaches. While our query-time computational cost (QT) is higher (1.3M tokens), this reflects a strategic architectural trade-off: we shift the

Table 2: Efficiency Analysis. **Indexing Tokens (IT)** measures the one-time cost of memory consolidation. **Querying Tokens (QT)** measures the operational cost per interaction. **Total Tokens (TT)** is the sum of IT and QT.

Method	Indexing Tokens (IT)	Querying Tokens (QT)	Total Tokens (TT)
Standard RAG	11.2k	848.1k	859.3k
LightRAG	36.4M	462.9k	36.9M
HippoRAG	6.6M	832.5k	7.4M
EcphoryRAG	**2.0M**	**1.3M**	**3.3M**

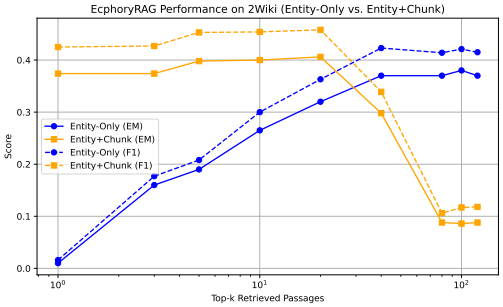

Figure 3: **Impact of Context Grounding (2Wiki).** The "Entity+Chunk" configuration (orange) significantly outperforms the "Entity-Only" baseline (blue), demonstrating that structured engrams alone are insufficient for evidence-based reasoning.

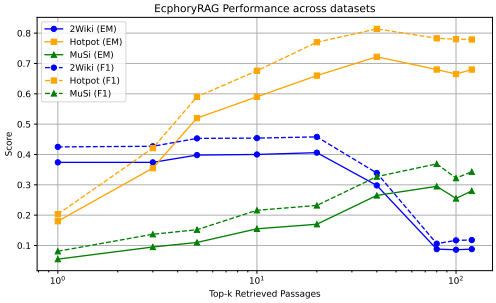

Figure 4: **Sensitivity to Context Size (k).** Performance varies by dataset information density. Noise-sensitive tasks (2Wiki) peak at lower k, while complex multi-hop tasks (HotpotQA/MuSiQue) require broader context windows.

computational burden from rigid offline indexing to flexible online associative reasoning. This design allows the agent to update its long-term memory rapidly ("fast write") while investing cognitive resources ("slow think") only when necessary to solve complex queries.

4.4 Ablation Studies

To rigorously validate the architectural decisions underpinning EcphoryRAG, we conducted a series of ablation studies. These experiments isolate the contributions of specific cognitive mechanisms—context grounding, associative depth, and memory capacity—to the overall reasoning performance.

4.4.1 The Necessity of Episodic Grounding

A central hypothesis of our framework is that entities serve as *navigational cues*, while text chunks provide the necessary *evidence*. To test this, we evaluated a variant of the model ("Entity-Only") that feeds only the structured engrams to the LLM, excluding the source text chunks.

As illustrated in Figure 3, the performance gap is substantial. The "Entity-Only" approach struggles to reconstruct the full reasoning chain, confirming that while the associative graph is highly effective for *locating* relevant information, it functions primarily as a semantic scaffold. The LLM requires the nuanced, unstructured details contained within the "Entity+Chunk" configuration to synthesize accurate answers. This validates our dual-store design: the graph provides the **address**, but the episodic store provides the **content**.

4.4.2 DEPTH OF ASSOCIATIVE PROCESSING

We analyzed the impact of the associative search depth (spreading activation hops) on reasoning accuracy and latency, using HotpotQA as a representative benchmark. Table 3 reveals that performance peaks at **Depth=2**.

- **Reasoning vs. Noise:** A depth of 0 or 1 is insufficient to bridge the "reasoning gaps" inherent in multi-hop questions, leading to lower recall. However, extending beyond Depth 2 (e.g., Depth 3) introduces semantic drift, where the retrieval focus wanders too far from the original query intent.
- **Efficiency:** Crucially, the computational cost of deeper retrieval is marginal. The average processing time remains stable (approx. 6.7s) even as depth increases, demonstrating that our centroid-based vector navigation allows for deep associative reasoning without the combinatorial explosion typically seen in explicit graph traversal.

Table 3: Effect of associative retrieval depth on HotpotQA. **Depth=2** offers the optimal trade-off between connecting multi-hop evidence and minimizing semantic drift, with negligible impact on latency.

Retrieval Depth	**EM** (↑)	**F1-Score** (↑)	**Avg Time (s)** (↓)
Depth 0 (Direct)	0.714	0.8085	**6.612**
Depth 1	0.702	0.8032	6.690
Depth 2	**0.722**	**0.8143**	6.734
Depth 3	0.698	0.7943	6.816

4.4.3 OPTIMAL MEMORY ACTIVATION (k)

Figure 4 investigates the system's sensitivity to the number of retrieved passages (k). The optimal k value correlates with the information density of the dataset. For **2WikiMultiHop**, which requires precise bridging of specific facts, a focused context ($k = 20$) is optimal; larger contexts introduce distraction ("lost-in-the-middle" phenomenon). Conversely, **HotpotQA** and **MuSiQue**, which involve gathering dispersed evidence amidst distractors, benefit from a broader activation window ($k = 40$ to 80). This suggests that the "working memory" capacity of the agent should be dynamically tuned based on the complexity of the environment.

5 CONCLUSION

In this work, we presented **EcphoryRAG**, a framework that operationalizes the cognitive principle of memory ecphory to address the plasticity-stability dilemma in agentic memory systems. By moving beyond static indexing to a dynamic, cue-driven associative search, our architecture reconciles the need for rapid memory consolidation ("fast-write") with the demand for deep, multi-hop reasoning ("deep-read"). Empirical results across three challenging benchmarks confirm that EcphoryRAG not only outperforms existing neuro-symbolic baselines in reasoning fidelity but does so with orders-of-magnitude greater efficiency. This structural advantage suggests that mimicking biological memory mechanisms—specifically the interplay between engram abstraction and spreading activation—is a viable path toward scalable, autonomous intelligence.

5.1 LIMITATIONS AND FUTURE WORK

While our results are promising, the current system relies on the fidelity of the initial engram extraction, which can be sensitive to the extraction model's capabilities. Looking forward, we envision three key directions to extend EcphoryRAG into a fully autonomous cognitive architecture:

1. **True Continual Learning via Sleep Consolidation:** We aim to implement an offline "sleep" phase where the agent autonomously refines and prunes its engram graph, abstracting generalized schemas from episodic experiences to prevent memory saturation.

2. **Goal-Conditioned Working Memory:** Extending the retrieval mechanism to populate a dynamic "working memory" buffer that persists across multi-turn interactions, allowing the agent to maintain state and context over long task horizons.
3. **Active Memory Construction:** Transitioning from passive text ingestion to active inquiry, where the agent can identify gaps in its associative graph and proactively seek information to complete its internal world model.

Ultimately, EcphoryRAG represents a step towards building more cognitively plausible AI. By grounding our engineering solutions in the time-tested principles of human memory, we aim to contribute to a new generation of systems capable of robust, scalable, and continuous learning.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. 2024.

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, 2024. URL https://arxiv.org/abs/2404.16130.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2410.05779.

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. URL https://arxiv.org/abs/2502.14802.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569, 2024.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pp. 6769–6781, 2020.

Mu-ming Poo, Michele Pignatelli, Tomás J Ryan, Susumu Tonegawa, Tobias Bonhoeffer, Kelsey C Martin, Andrii Rudenko, Li-Huei Tsai, Richard W Tsien, Gord Fishell, et al. What is memory? the present state of the engram. *BMC biology*, 14(1):40, 2016.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 05 2022. ISSN 2307-387X. doi: 10.1162/tacl_a_00475. URL https://doi.org/10.1162/tacl_a_00475.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Endel Tulving. Précis of elements of episodic memory. *Behavioral and Brain Sciences*, 7(2): 223–238, 1984. doi: 10.1017/S0140525X0004440X.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

A Implementation Details

To ensure the reproducibility of our results and provide transparency regarding the computational resources required for EcphoryRAG, we detail the experimental environment, model specifications, and hyperparameter settings below.

A.1 Computational Environment

All experiments were orchestrated using Python 3.10 on a Linux-based server (Ubuntu 22.04) equipped with 4×NVIDIA RTX 4090 GPUs (96GB VRAM total). The core retrieval and memory operations rely on optimized vector and graph libraries to ensure low latency. The software stack is detailed in Table 4.

Table 4: Software and Framework Configuration

Component	**Specification**
Programming Language	Python 3.10
Inference Engine	Ollama (Local Deployment)
Vector Indexing	FAISS CPU ($\geq$ 1.7.4)
Graph Processing	NetworkX
Text Processing	LangChain (RecursiveCharacterTextSplitter)

A.2 Model Architectures

We standardize the backbone models across the memory lifecycle to maintain consistency.

- **Embedding Model:** We utilize **bge-m3** (Xiao et al., 2023) with an embedding dimension of 1024. This model was selected for its multi-granularity capability, essential for encoding both short entities and longer text chunks.
- **Backbone LLM:** We employ **Phi-4:14b** (Abdin et al., 2024) for both *Entity Extraction* (during memory consolidation) and *Answer Generation* (during retrieval). Phi-4 offers a compelling balance of reasoning capability and computational efficiency, making it suitable for agentic workflows.

A.3 Data Preprocessing and Representation

Raw documents are pre-processed into a standardized JSON format before being ingested into the memory system. We employ a chunking strategy with a size of 1200 characters and an overlap of 200 characters to preserve context at boundaries.

```
{
    "id": "sample_id",
    "question": "multi-hop question text",
    "answer": "ground truth answer",
    "answer_aliases": ["alias1", "alias2"],
    "chunks": [
        {
            "id": "chunk_id",
            "title": "Document Title",
            "chunk": "document text content"
        }
    ],
    "supporting_facts": [
        {
            "id": "sf_id",
            "title": "Title",
            "chunk": "supporting fact text"
        }
    ]
}
```

A.4 Hyperparameters and System Configuration

Table 5 comprehensively lists the hyperparameters used for retrieval dynamics and system constraints. Note that we strictly enforce memory limits (e.g., STM Capacity) to simulate realistic resource-constrained agent environments.

Compression Strategy: To optimize the context window usage, retrieved text chunks exceeding a token threshold (200-300 words) are dynamically compressed by the LLM to under 100 words during the generation phase, ensuring that the reasoning core receives high-density information.

A.5 Evaluation Metrics

We report performance using standard multi-hop QA metrics:

- **Exact Match (EM):** Measures whether the predicted answer exactly matches the ground truth after normalization. Normalization includes lowercasing, article removal (a, an, the), and punctuation stripping.
- **Token-level F1:** Calculates the harmonic mean of precision and recall at the token level. For questions with multiple valid answers (aliases), we report the maximum F1 score achieved against any valid reference.

Table 5: Hyperparameters and System Configuration for EcphoryRAG.

Category	Parameter	Value / Description
Retrieval Dynamics	`top_k_initial`	10 (Initial engrams activated by query)
	`top_k_final`	Tuned per dataset: $\{1, 3, 5, \ldots, 120\}$
	`retrieval_depth`	2 (Hops of spreading activation)
	Expansion Factor	$3\times$ (Search scope per hop: $k_{init} \times 3$)
	Hybrid Retrieval	Enabled (Combines Vector + Graph)
System & Memory	STM Capacity	20 (Max active traces in Short-Term Memory)
	Embedding Cache	1000 entries (LRU Strategy)
	Emb. Truncation	2048 chars (Max input for embedding)
	Extract. Truncation	4000 chars (Max input for entity extraction)
	Index Type	FAISS `IndexIDMap2(IndexFlatL2)`
	Incremental Indexing	Enabled (via MD5 content hashing)

A.6 Baseline Implementation

For all baseline methods (Standard RAG, LightRAG, HippoRAG, etc.), we utilized the official codebases and adhered to the default best configurations recommended by the respective authors to ensure a fair comparison.

B Prompt Templates

To facilitate reproducibility and further research into neuro-symbolic agent memory, we provide the exact prompt templates used in EcphoryRAG. These prompts are fed into the **Phi-4** model. In the templates below, terms enclosed in double curly braces (e.g., {{`input`}}) denote dynamic slots filled by the system during runtime.

B.1 Memory Consolidation: Engram Extraction

This prompt is used during the *Indexing Phase* to abstract raw text chunks into structured entity engrams (Entity + Type + Description).

```
You are an expert knowledge engineer building a memory system for an
AI agent.  Your task is to extract key entities (engrams) from the
following text chunk.
Text Chunk:  text_chunk
Requirements:  1.  Identify important entities (Person, Location,
Organization, Event, Date, Concept).
2.  For each entity, provide a concise description based ONLY on the
text provided.
3.  Assign an importance score (1-5) based on its relevance to the
core topic.
4.  Focus on entities that are crucial for understanding the context.
Respond in the following JSON format:  [  "text":  "entity1", "type":
"PERSON/LOCATION/ORGANIZATION/DATE/CONCEPT/EVENT/...", "description":
"brief contextual information", "importance_score" : 1 - 5, ...]
Only return the JSON array, nothing else.
Output:
```

Figure 5: Prompt template for Engram Extraction (Memory Consolidation).

B.2 ASSOCIATIVE RECALL: CUE EXTRACTION

This prompt is used at the beginning of the *Retrieval Phase* to identify ”retrieval cues” from the user’s query, simulating the initial activation of memory traces.

```
You are the memory retrieval mechanism of an intelligent agent.
Analyze the user's query and extract "Retrieval Cues". These are
the key entities or concepts that will be used to start a search in
the long-term memory.
User Query: user_query
Instructions: - Extract specific named entities. - Extract temporal
or locational constraints. - Do not answer the question, just list
the cues.
Output Format: Cue 1, Cue 2, Cue 3...
Output:
```

Figure 6: Prompt template for Cue Extraction (Initial Activation).

B.3 REASONING AND ANSWER GENERATION

This prompt is used in the final stage to synthesize the retrieved memory traces (engrams + chunks) into a coherent answer.

```
You are an expert research assistant specializing in multi-hop
reasoning and connecting information across different sources. Your
task is to answer a complex question by carefully analyzing the
provided information.
QUESTION: query_text
AVAILABLE INFORMATION:
KEY ENTITIES IDENTIFIED: entity_context
SUPPORTING TEXT PASSAGES: chunk_context
INSTRUCTIONS: 1. This question requires multi-hop reasoning - you
need to connect information from different passages.
2. Think step-by-step through the reasoning process before giving
your final answer.
3. Your answer MUST be derived from the provided information only.
Do not use external knowledge.
4. If the information provided is insufficient, clearly state what's
missing.
5. After your reasoning, provide the final concise answer on a new
line, prefixed with 'FINAL_ANSWER_TEXT:'.
6. Your answer MUST be a single word or short phrase that directly
answers the question, with no explanation or extra text.
7. If the answer is not present in the provided information, output:
"NOT FOUND".
REASONING STEPS: (Work through your reasoning process here,
connecting information across different passages)
FINAL_ANSWER_TEXT:
```

Figure 7: Prompt template for Final Reasoning and Generation.