EVOLINKBENCH: A BENCHMARK AND FRAMEWORK FOR KNOWLEDGE GRAPH SELF-EVOLUTION IN MEMORY-AUGMENTED AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent memory is evolving from transient context into long-term memory that can be continually updated. Existing research on self-evolving memory primarily investigates how systems distill and update experience, reflections, and strategies from interaction trajectories. In contrast, graph memory that encodes factual and semantic knowledge generally remains relatively static after its initial construction, and its capacity to proactively reorganize its internal structure using existing evidence remains underexplored. In particular, such systems struggle to discover hidden relations that can only be inferred by combining multiple text chunks located far apart in a document. These structural gaps disrupt multi-hop evidence chains, prevent complete retrieval of relevant evidence, and ultimately lead to question-answering failures. We argue that knowledge graphs should also support self-evolution by reorganizing existing static structures and persistently writing back verified relations inferred from evidence across text chunks, thereby improving relation coverage and evidence connectivity. To study this problem, we introduce EvoLinkBench, a benchmark for graph self-evolution. Using a latent-graph-driven construction approach, the benchmark generates source texts and several categories of questions requiring long-range reasoning, each grounded in an evidence chain, to evaluate relation discovery and its effects on downstream question answering. Building on this benchmark, we propose Evidence-Chain-Linked Relation Recovery (ECLRR). ECLRR searches the initial static graph for candidate paths that satisfy type constraints, assembles the source-text evidence supporting each hop, and writes hidden relations back to the graph after relation generation and independent verification. Experimental results show that Q-ECLRR effectively discovers hidden relations supported by long-range evidence chains and improves the performance of graph-augmented systems on subsequent multi-hop reasoning and hidden-relation inference tasks.

1 INTRODUCTION

Retrieval-augmented generation and agent memory systems typically treat indexed factual and semantic knowledge as static memory. Recent work on self-evolving memory distills experience, reflections, and strategies from interaction trajectories, primarily updating experiential knowledge about how to solve problems. Whether a graph can proactively reorganize existing factual knowledge and its relations using documents already available remains underexplored. Can graph-based factual memory discover previously unestablished relations without new input and consolidate them into persistent structures reusable in future tasks?

This problem is pronounced in graph construction from text chunks. In Figure 1, chunks c_1 through c_5 encode $A \to B$, $B \to C$, $C \to D$, $D \to E$, and $E \to F$, respectively. Together, these facts support a relation r_{AF} that no individual chunk states. Conventional GraphRAG systems extract explicit entities and relations within chunks, retaining these local facts without consolidating the full evidence chain into an edge $A \xrightarrow{r_{AF}} F$ between distant endpoints.

For a question about A and F, text and graph retrieval may prioritize the endpoint chunks c_1 and c_5, omitting bridging evidence c_2–c_4 because of retrieval budgets or relevance rankings (Figure 1(a)).

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.

Increasing Top-k, using iterative retrieval, or extending context may recover the complete chain for one query, but leaves the persistent graph unchanged. Subsequent questions require the same intermediate evidence and inference again. Missing hidden relations therefore reflect both incomplete retrieval and a limited capacity for background consolidation of existing knowledge.

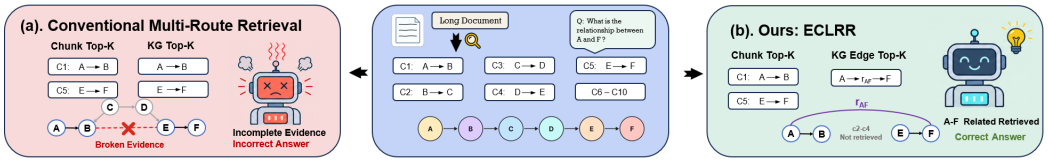


Figure 1: Hidden relations and missing bridging evidence across text chunks.

Inspired by the reorganization of human memory associations during sleep or rest, we propose offline reorganization of graph-based factual memory. We define *evidence-based graph self-evolution* as discovering hidden relations supported by cross-chunk evidence chains in G_0, verifying their semantics, direction, and evidence completeness, and writing them back to form a structurally more complete graph G_1. Paths identify candidates only: every new edge requires complete source-text support and a traceable evidence chain.

Unlike knowledge graph completion, which primarily predicts missing triples from graph structure without requiring complete documentary evidence, this process requires relations jointly inferred from multiple chunks. Unlike query-time multi-hop reasoning, whose conclusions typically leave persistent memory unchanged, it consolidates verified relations into reusable graph structures.

Existing benchmarks provide limited visibility into this transition. Long-term memory evaluations test retention, updating, and retrieval of observed facts; self-evolving memory evaluations test the future value of experience; and multi-hop QA and cross-document relation extraction evaluate answers, evidence, or relation labels. They rarely check pre-query discovery, reliable writeback, or future utility without error propagation. Correctly answering one question therefore does not establish effective graph-memory evolution.

We introduce EvoLinkBench for long-range hidden-relation discovery and graph self-evolution. Its latent-graph-driven construction specifies target relations and necessary evidence chains before distributing local facts across long documents. No individual chunk establishes the target, but the complete chain does. Single-chunk checks, evidence removal, and counterfactual interventions evaluate relation discovery, chain recovery, relation use, and error propagation.

We also propose ECLRR: candidate fact-chain search, hop-level source recovery, relation generation, independent verification, and deterministic admission write reliable relations and evidence back to the graph. Its query-conditioned variant, Q-ECLRR, reduces graph-wide search costs by recovering chains in query-relevant subgraphs and persistently writing verified relations and evidence back to the graph for retrieval and reuse.

EvoLinkBench challenges existing retrieval and memory frameworks: most baselines cannot consistently recover hidden cross-chunk relations. Q-ECLRR improves single-hop and standard multi-hop evidence recovery and reasoning, while Safe Routing improves selective-forgetting performance. These preliminary findings motivate further study of persistent writeback, cross-query reuse, and graph contamination control.

1.1 Contributions

We introduce evidence-based graph self-evolution as verifiable, query-independent, persistent structural reorganization using existing static evidence, distinct from graph completion, query-time reasoning, and unconstrained graph expansion.

We construct EvoLinkBench, using latent graphs, controlled long documents, ground-truth hidden relations, and necessary evidence chains to evaluate discovery, evidence recovery, persistent writeback, future utility, and graph contamination.

We propose Q-ECLRR to consolidate distributed facts into persistent relations through type-constrained path search, hop-level evidence tracing, independent verification, and provenance-

preserving writeback. Relations can be expanded into supporting evidence, audited, and revoked. Experiments demonstrate the method's effectiveness.

2 Related Work

2.1 Benchmarks for Long-Term Memory and Long-Range Reasoning

Long-term memory benchmarks assess retention and retrieval across conversational sessions. LongMemEval evaluates information extraction, cross-session and temporal reasoning, knowledge updates, and abstention; LoCoMo examines factual, temporal, and event understanding in very long multi-session conversations; and MemoryAgentBench tests retrieval, test-time learning, and conflict resolution in incremental interactions (Wu et al., 2025; Maharana et al., 2024; Hu et al., 2025). Their updates primarily concern adding, replacing, correcting, or temporally updating explicit facts already observed.

LongBench and LongBench v2 assess understanding of long inputs and multiple documents (Bai et al., 2024; 2025). HotpotQA, 2WikiMultiHopQA, MuSiQue, and IIRC require evidence composition, providing supporting sentences, reasoning paths, or unanswerable instances to varying degrees (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Ferguson et al., 2020). DocRED, Re-DocRED, and CodRED evaluate entity relations within or across documents (Yao et al., 2019; Tan et al., 2022; Yao et al., 2021).

These benchmarks establish that answers and relations can depend on distant evidence, but supervise final answers, evidence sets, or relation labels. They generally do not test whether an inferred relation enters a persistent graph and improves later retrieval. Our target transition is to discover relations jointly entailed by cross-chunk chains but unstated in individual chunks, then convert them into reusable graph edges.

2.2 Self-Evolving Memory and Learning from Experience

Self-evolving memory improves later agent performance through accumulated trajectories. EvoMemory reorganizes independent instances into continuous task streams; EvoMemBench assesses evolution within and across episodes and across knowledge-oriented and execution-oriented memory; and SEA-Eval emphasizes long-term evolution across tasks (Wei et al., 2025; Wang et al., 2026; Jiang et al., 2026). Evolution signals typically arise from new interactions, outcomes, or feedback.

Methods likewise focus on experience. ExpeL and Reflexion consolidate task experience or verbal feedback, while A-MEM creates and adjusts links as memories arrive (Zhao et al., 2024; Shinn et al., 2023; Xu et al., 2025). Experience, reflection, strategies, and skills support continual learning without parameter updates, but the evolving objects are usually experience repositories, prompts, strategies, skills, or task-conditioned notes; static memory itself does not evolve.

2.3 Graph Memory, Graph Retrieval, and Knowledge Graph Completion

GraphRAG uses entity graphs, communities, and hierarchical summaries for global questions; LightRAG combines entity-level and topic-level representations; and RAPTOR uses recursive clustering and summary trees for retrieval across granularities (Edge et al., 2024; Guo et al., 2025; Sarthi et al., 2024). HippoRAG, G-Retriever, and knowledge graph-guided RAG improve multi-hop evidence recall through graph propagation, compact subgraphs, and graph expansion, respectively (Gutiérrez et al., 2024; He et al., 2024; Zhu et al., 2025).

Graphs also organize and maintain long-term memory. A-MEM updates links as new memories arrive, HyperMem represents higher-order dependencies through hypergraphs, and ReGraphRAG reorganizes fragmented document graphs (Xu et al., 2025; Yue et al., 2026; Kim et al., 2025). Conversely, TransE, ComplEx, and RotatE predict missing triples from graph representations and relational patterns (Bordes et al., 2013; Trouillon et al., 2016; Sun et al., 2019). The former approaches improve existing memory organization and retrieval; the latter estimate edge plausibility in topological or representation space. We instead study revisions to the knowledge graph itself through hidden relations, beyond reordering existing elements or predicting links without textual support.

3 METHOD

3.1 EVOLINKBENCH CONSTRUCTION

3.1.1 CONSTRUCTION OVERVIEW

Multi-hop QA benchmarks such as HotpotQA commonly use public Wikipedia passages, while long-term memory benchmarks such as LongMemEval and LoCoMo use extended conversations. These materials may appear in LLM pretraining data, introducing contamination or memorized answering. Annotating answers and supporting facts after corpus collection also limits joint control over target relations, evidence distribution, and the necessity of each reasoning step.

EvoLinkBench instead starts with a latent graph specifying the complete fact structure, target hidden relations, and necessary evidence chains. Local facts are verbalized into long narratives; QA instances are then generated exclusively from the frozen graph and text. This reduces contamination risks from public-corpus reuse and supplies unified, auditable supervision for relations, gold answers, and hop-level evidence.

3.1.2 GOLD GRAPH AND NARRATIVE GENERATION

Construction comprises gold-graph design, controlled narrative generation, consistency verification, evidence alignment, and version freezing (Figure 2). The gold graph specifies node types, relation semantics and directions, temporal scopes, and epistemic states. Explicit fact edges must have directly locatable textual support. Hidden target edges require multi-hop inference and cannot be directly stated in any individual sentence or local passage.

Given the graph, entity specifications, and narrative constraints, an LLM generates five coherent chapters. Fact edges are expressed naturally and unambiguously; facts supporting hidden edges are distributed across chapters or chunks. Unrelated distractors prevent trivial evidence matching.

Agent-based and manual checks examine narrative consistency, relation direction, temporal order, and joint support for hidden relations. Invalid narratives or paths are revised or removed. Verified narratives are segmented with LightRAG's default chunking strategy. Each fact edge is aligned to a source span, and each hidden edge to its complete fact path and hop-level evidence locations. The graph, narrative, chunk mapping, and annotations are frozen together as the sole source for QA construction.

3.1.3 QA CONSTRUCTION AND VALIDATION

We select a target relation from the frozen graph, recover its minimally sufficient chain, and create an internal Question Card containing the relation, bridge entities, hop-level evidence, and gold answer (Figure 3). The generated question exposes no graph structure, path information, or chunk identifiers.

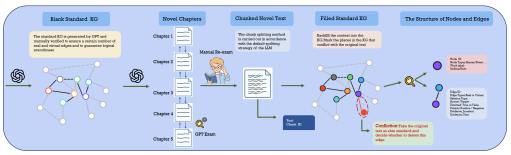


Figure 2: Gold-graph design and controlled narrative construction.

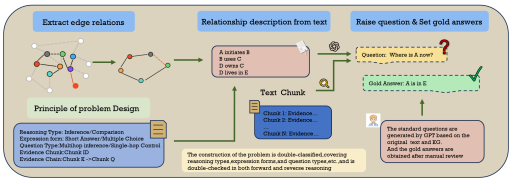


Figure 3: Question construction from the frozen graph and evidence chains.

Three question types are included (Figure 4). Multi-hop questions require three to six consecutive fact edges across at least two chunks. Selective Forgetting removes, negates, or reverses one fact and tests the updated conclusion. Single-hop questions control for explicit fact retrieval.

Before answer generation, we check path continuity, hop-level alignment, single-chunk unanswerability, alternative paths, and answer changes under leave-one-evidence-out testing. After generating the gold answer, we check bidirectional consistency among the question, answer, chain, and relation. We remove instances with a target stated in one chunk, multiple possible answers, redundant chain steps, or exposed implementation-specific information.

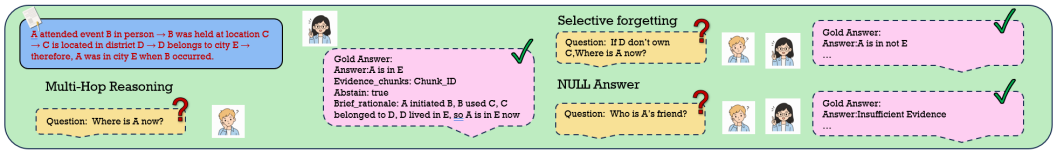


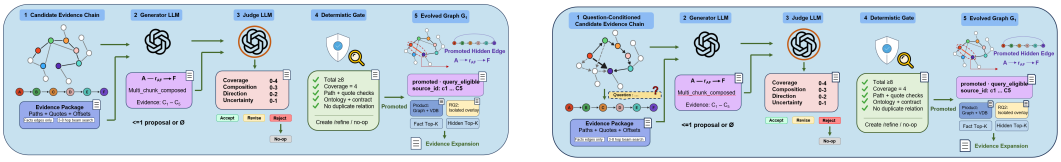
Figure 4: Illustrative question construction and evidence interventions.

3.1.4 Evaluation Dimensions

Evidence Retrieval measures retrieval of all necessary chunks; Chain Recovery tests complete chain reconstruction; Answer Correctness measures final-conclusion accuracy; and Selective Forgetting checks answer updates after evidence removal, negation, or reversal. Joint evaluation distinguishes retrieval failure, broken chains, incorrect reasoning, and failures to update conclusions after evidence changes.

3.2 ECLRR: Evidence-Chain-Linked Relation Recovery

ECLRR operates offline after indexing to discover relations unstated in individual chunks but inferable from cross-chunk chains. It comprises candidate-path search, source-evidence recovery, relation proposal, independent verification, deterministic admission, and persistent writeback (Figure 5(a)). Connectivity identifies candidate endpoints only; new relations require complete textual support and valid composition rules.



(a) Offline ECLRR (b) Query-conditioned Q-ECLRR

Figure 5: Evidence-chain-linked relation recovery: (a) offline ECLRR; (b) query-conditioned Q-ECLRR.

3.2.1 Candidate Evidence Chain Search

For graph $G_0 = (V, E_0)$ and chunk collection $\mathcal{C}$, ECLRR constructs a graph of source-traceable fact edges:

$$G_F = (V, E_F), \qquad E_F = \{e \in E_0 \mid e \text{ is an original fact edge}\}. \tag{1}$$

Ambiguous and generated edges are excluded so uncertain or derived relations cannot recursively support themselves. Beam search identifies simple paths between non-adjacent entities:

$$P = (v_0, e_1, v_1, \ldots, e_L, v_L), \qquad 3 \le L \le 8. \tag{2}$$

Search prioritizes reliable facts, complete provenance, and composable relations, penalizing excessive length, reverse traversal, high-degree intermediates, and conflicts. Two-hop paths are considered when a predefined deterministic composition rule applies.

For each path, source evidence is recovered in hop order. Missing explicit, traceable support for any hop disqualifies the path. The evidence package $\mathcal{E}_P$ contains endpoints, the fact path, and hop-level passages. A complete, traceable path does not itself prove the endpoint relation.

3.2.2 Relation Proposal

Given $\mathcal{E}_P$ and relation constraints, the Generator proposes

$$\pi = \text{Generator}(\mathcal{E}_P, \Omega), \qquad \pi \in \mathcal{R} \cup \{\varnothing\}, \tag{3}$$

where Ω specifies the ontology and composition rules and $\mathcal{R}$ the admissible relations. Each chain yields at most one proposal, or $\varnothing$ for insufficient evidence. Proposals specify type, direction, support mode, description, and evidence citations.

Support can be direct or compositional. For example, *A participated in E* and *E occurred at L* may support *A visited L* under an appropriate rule. Co-occurrence, shared locations, or connectivity alone

are insufficient. If all hops are supported but entail no well-defined endpoint relation, a retrieval-only evidence bridge may record the full path without asserting a new endpoint fact.

3.2.3 Independent Verification and Writeback

The Judge independently reads the original evidence and proposal, assesses coverage, semantic composability, direction, and uncertainty calibration, and returns accept, revise, or reject. Its score is

$$s_{\text{judge}} = s_{\text{cov}} + s_{\text{comp}} + s_{\text{dir}} + s_{\text{unc}}. \tag{4}$$

The LLM-based Judge cannot directly modify the graph. A deterministic Gate additionally requires

$$\begin{gathered} d \neq \text{reject}, \quad s_{\text{judge}} \geq 8, \quad s_{\text{cov}} = 4, \\ s_{\text{comp}} \geq 2, \quad s_{\text{dir}} \geq 1. \end{gathered} \tag{5}$$

It checks consecutive original fact edges, quotations matching source passages, an ontology-valid relation and direction, and valid composition. For retrieval-only bridges, it verifies complete, faithful hop-level evidence and prevents representation as definite endpoint facts.

The Gate returns create, refine, or no-op: a new hidden relation, refinement of an ambiguous relation, or rejection. Accepted relations retain complete chains and verification records for retrieval and provenance tracing.

3.2.4 Hidden Relation Retrieval

Original facts and accepted hidden relations use separate retrieval budgets:

$$\mathcal{R}(q) = \text{TopK}(E_{\text{src}}, q; k_s) \cup \text{TopK}(E^{+}_{\text{hid}}, q; k_h), \tag{6}$$

where E_{src} and E^{+}_{hid} denote source facts and admitted hidden relations. Separate budgets prevent either source evidence or useful hidden relations from being displaced.

Retrieved canonical relations expand into their original evidence chains. Retrieval-only bridges recall associated source chunks without asserting endpoint facts. Thus, hidden relations can shorten repeated multi-hop reasoning while preserving document provenance.

3.3 Q-ECLRR: Query-Conditioned ECLRR

Full ECLRR searches the entire fact graph before QA and persistently writes back relations, enabling cross-query reuse but incurring substantial search and verification costs. Q-ECLRR restricts discovery to a query-relevant local subgraph while retaining persistent writeback of verified relations and their supporting evidence.

From question q, it extracts explicit entities, target relation, expected answer type, direction, negation, and counterfactual constraints. Entity linking provides anchor nodes:

$$A_q = \text{Link}(\text{Entity}(q), V). \tag{7}$$

Search explores three- to eight-hop fact paths from A_q. Terminal nodes are candidate answers; gold answers are never used. Ranking considers query intent, answer type, fact quality, initial retrieval, and compatibility between endpoints and query clues. Paths missing source evidence are discarded.

After selecting the primary chain, Q-ECLRR uses the same Generator, Judge, and Gate. The methods differ in discovery scope: ECLRR performs graph-wide offline discovery, whereas Q-ECLRR recovers the chain most relevant to the current question. Both persistently write verified relations and their supporting evidence back to the graph. Q-ECLRR retrieves this evidence for the current answer and retains the written relations for subsequent queries (Figure 5(b)).

4 Experiments

4.1 Experimental Setup

Benchmark. The experiments evaluate 359 QA instances from EvoLinkBench: 25 Single-hop, 271 Multi-hop, and 63 Selective Forgetting questions. They test explicit fact retrieval, cross-chunk evidence composition, and conditional relation deletion, respectively.

Baselines. RQ2 compares Dense RAG, LightRAG, Mem0, MemOS, and GraphRAG with the answer model fixed within each block (Qwen-Plus or GPT-6-Luna) to isolate memory and retrieval architectures. RQ3 fixes LightRAG retrieval and compares Qwen-Plus, DeepSeek-Flash, GPT-5-mini, GPT-5.1, Gemini 2.5 Flash, Gemini 3 Flash, and GPT-6-Luna. RQ4 compares LightRAG, LightRAG + Q-ECLRR, and LightRAG + Q-ECLRR + Safe Routing on the same three question categories: 25 Single-hop, 271 Multi-hop, and 63 Selective Forgetting questions (359 in total).

Metrics. Answer Token F1 and LLM-judged Accuracy assess final answers; GPT-5.6-sol judges correctness against gold answers. Semantic Evidence Recall@K, Complete Chain@K, and Terminal Recall@K measure local evidence coverage, complete chain recovery, and terminal-evidence recall, respectively.

Implementation Details. All methods share retrieval and context budgets. We use text-embedding-v3 for embeddings and Qwen-Turbo for entity extraction. The graph entity/relation candidate budget is 40, the chunk retrieval budget is 20, and maximum context length is 30,000 tokens. Answers must use only retrieved context and abstain from definitive conclusions when evidence is insufficient.

4.2 RQ1: Performance on Established Multi-hop QA Benchmarks

Before evaluating EvoLinkBench, we evaluate Qwen-Plus and DeepSeek-Flash on HotpotQA and 2WikiMultihopQA using the same answer-generation and scoring procedure (Table 1). Both models achieve relatively high F1 scores, but these benchmarks do not systematically isolate long evidence-chain reasoning, intermediate bridging-evidence recovery, or selective forgetting.

Table 1: F1 scores (%) on HotpotQA and 2WikiMultihopQA.

Answer Model	HotpotQA	2WikiMultihopQA
Qwen-Plus	71.01	64.06
DeepSeek-Flash	**82.56**	**67.65**

4.3 RQ2: Performance across Memory Frameworks

We compare end-to-end QA across memory and retrieval frameworks with the answer model fixed within each block.

Table 2: End-to-end QA accuracy (%) of memory and retrieval frameworks with the answer model fixed within each block.

Answer Model	Method	Single-hop	Multi-hop	Selective Forgetting	Overall
Qwen-Plus	Dense RAG	68.00	22.88	55.56	28.64
	LightRAG	**72.00**	22.88	52.38	**30.91**
	Mem0	60.00	**23.25**	**66.67**	30.45
	MemOS	64.00	21.40	52.38	28.18
	GraphRAG	40.00	11.81	29.47	22.95
GPT-6-Luna	Dense RAG	10.00	14.02	**35.79**	21.59
	LightRAG	48.00	18.45	7.37	17.60
	Mem0	**76.00**	14.76	11.58	17.90
	MemOS	20.83	**36.60**	34.74	**29.63**
	GraphRAG	64.00	3.32	18.00	15.23

For Qwen-Plus, Table 2 shows substantially higher Single-hop than Multi-hop performance: frameworks retrieve explicit facts but struggle with hidden endpoint relations supported by multiple chunks. Dense RAG and LightRAG's similar Multi-hop performance and LightRAG's weaker Selective Forgetting performance indicate unresolved evidence-chain gaps and possible interference from outdated relations.

4.3.1 Evidence Retrieval Analysis

To locate QA failure sources, we compare evidence retrieval performance.

Increasing Top-k improves local recall but yields limited complete-chain recovery. MemOS achieves 62.81% Terminal Recall@20 but only 31.20% Complete Chain@20, indicating that terminal evidence is easier to retrieve than intermediate bridges. Dense RAG exceeds LightRAG on most recall

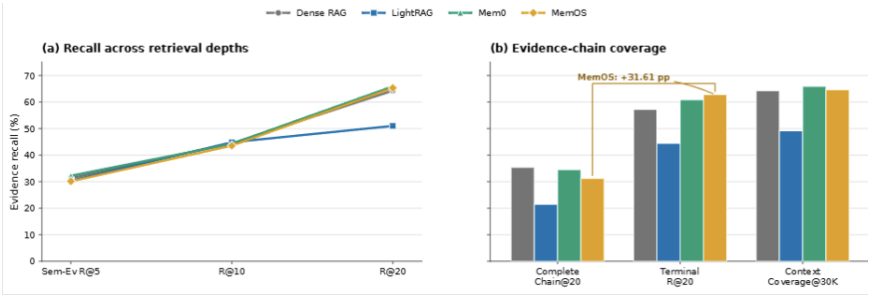


Figure 6: Evidence retrieval performance under different retrieval budgets. (a) Semantic evidence recall at different values of K. (b) Complete-chain, terminal-evidence, and context-coverage recall.

metrics yet trails by 2.27 percentage points in final accuracy: retrieving more isolated passages does not necessarily improve multi-hop reasoning.

4.4 RQ3: Impact of Answer Models

We fix LightRAG's retrieval results and vary the answer model to test whether model capabilities compensate for missing evidence.

Table 3: Performance of different answer models under the same LightRAG retrieval setting.

Answer Model	Single-hop	Multi-hop	Selective Forgetting	Overall
Qwen-Plus	**72.00**	22.88	**52.38**	**30.91**
DeepSeek-Flash	64.00	26.94	12.70	28.64
GPT-5-mini	**72.00**	25.46	19.05	28.41
GPT-5.1	64.00	23.25	25.40	30.23
Gemini 2.5 Flash	64.00	18.08	19.05	23.86
Gemini 3 Flash	64.00	**31.73**	17.46	30.00
GPT-6-Luna	48.00	18.45	7.37	17.60

Overall accuracy ranges from 17.60% to 30.91%, with a maximum Multi-hop accuracy of 31.73%. Some models combine retrieved evidence better, but none consistently compensates for missing intermediate evidence. Answer models influence context use; evidence retrieval and organization remain the main bottleneck for hidden cross-chunk relations.

4.5 RQ4: Effectiveness and Applicability of Q-ECLRR

We augment LightRAG with Q-ECLRR, which searches query-relevant fact paths, recovers hop-level source passages, and persistently writes verified hidden relations and evidence back to the graph for answer generation and subsequent reuse.

Table 4: Effectiveness of Q-ECLRR and safe routing across different question types. Values denote end-to-end QA accuracy (%).

System	Single-hop (25)	Multi-hop (271)	Selective Forgetting (63)	Overall (359)
LightRAG	72.00	22.98	52.38	30.91
Q-ECLRR	**72.22**	**28.40**	52.83	35.33
Q-ECLRR + Safe Routing	**72.22**	**28.40**	**64.15**	**38.00**

Q-ECLRR organizes distributed evidence into explicit relations for answer generation. Its overall accuracy is 35.33%, compared with 30.91% for LightRAG. Its accuracy is 72.22% for Single-hop, 28.40% for Multi-hop, and 52.83% for Selective Forgetting. Recovered relations may become invalid when a supporting fact is deleted or reversed.

4.5.1 Effect of Safe Routing

Safe Routing identifies deletion, reversal, negation, and insufficient-evidence cues from question text alone. It enables Q-ECLRR for standard questions and disables hidden-relation injection for high-risk questions, reverting to original LightRAG.

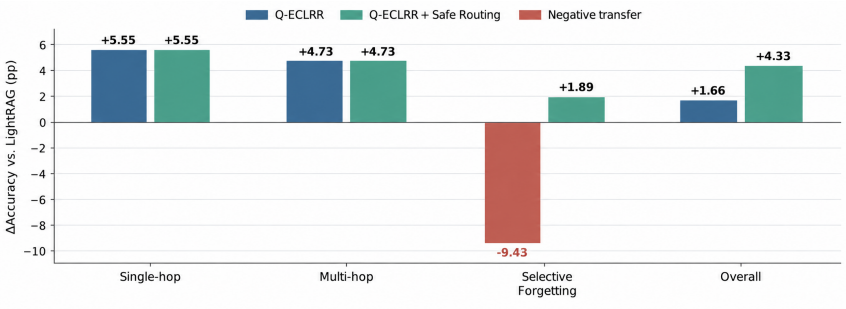


Figure 7: Accuracy changes of Q-ECLRR and safe routing relative to the LightRAG baseline across question types.

Safe Routing increases overall accuracy from 35.33% to 38.00% and Selective Forgetting accuracy from 52.83% to 64.15%, a gain of 11.32 percentage points on the latter. Single-hop and Multi-hop accuracy remain unchanged at 72.22% and 28.40%, respectively.

5 Limitations

EvoLinkBench has three limitations.

1. Our experiments cover a limited set of answer models. The results do not establish the performance ceiling of all frontier models; untested models may perform differently.
2. The benchmark contains only 10 texts and 359 QA instances, potentially limiting statistical stability and external validity. Larger datasets with broader topics and more complex text distributions require evaluation. The current release contains no dedicated insufficient-evidence question category, so these experiments do not establish abstention performance.
3. Systematic graph-wide experiments with offline ECLRR remain incomplete. Current experiments use Q-ECLRR to assess evidence-chain recovery for QA and provide preliminary support for hidden-relation recovery. Future work will evaluate the full graph self-evolution process more systematically.

6 Conclusion

We introduce EvoLinkBench for hidden-relation discovery across text chunks and multi-hop QA, with systematic data construction and evidence annotation. We formalize the difficulty of recovering relations unstated in individual chunks as evidence-based graph self-evolution. ECLRR provides explicit, traceable relational evidence through cross-chunk chain recovery, relation generation, independent verification, and controlled writeback.

EvoLinkBench is challenging: most baseline frameworks achieve overall accuracy below 40%. Q-ECLRR improves cross-chunk evidence recovery and composition for single-hop and standard multi-hop questions, but remains limited on Selective Forgetting. Safe Routing selectively uses hidden relations based on question semantics, improving performance when evidence changes make recovered relations inapplicable. These preliminary findings support the value of evidence-chain recovery for long-range reasoning and motivate further study of safe relation updates and hidden-relation applicability.

AI USE STATEMENT

Generative AI was used in the construction of EvoLinkBench, including synthetic narrative generation and consistency checks. This use is a core component of the benchmark construction methodology presented in this paper. AI tools also supported manuscript formatting and auxiliary work. The use of LLMs for entity extraction, question answering, relation proposal and verification, and answer evaluation is described in the method and experimental setup. The authors take responsibility for the final content of the paper, including its data, methods, results, and claims.

REFERENCES

Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou, Y. Dong, J. Tang, and J Li. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding, 2024. In Proceedings of ACL 2024, 3119–3137.

Y. Bai, S. Tu, J. Zhang, H. Peng, X. Wang, X. Lv, S. Cao, J. Xu, L. Hou, Y. Dong, J. Tang, and J Li. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks, 2025. In Proceedings of ACL 2025, 3639–3664.

A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O Yakhnenko. Translating Embeddings for Modeling Multi-relational Data, 2013. In Advances in Neural Information Processing Systems 26.

S. Diekelmann and J Born. The Memory Function of Sleep, 2010. Nature Reviews Neuroscience, 11: 114–126.

D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization, 2024. arXiv:2404.16130.

J. Ferguson, M. Gardner, H. Hajishirzi, T. Khot, and P Dasigi. IIRC: A Dataset of Incomplete Information Reading Comprehension Questions, 2020. In Proceedings of EMNLP 2020.

Z. Guo, L. Xia, Y. Yu, T. Ao, and C Huang. LightRAG: Simple and Fast Retrieval-Augmented Generation, 2025. In Findings of EMNLP 2025, 10746–10761.

B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, and Y Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models, 2024. In Advances in Neural Information Processing Systems 37.

X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B Hooi. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering, 2024. In Advances in Neural Information Processing Systems 37.

X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A Aizawa. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps, 2020. In Proceedings of COLING 2020, 6609–6625.

Y. Hu, Y. Wang, and J McAuley. MemoryAgentBench: Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions, 2025. arXiv:2507.05257.

S. Jiang, L. Ma, Z. Hong, K. Wang, Z. Lu, S. Chen, J. Zhang, T. Pan, W. Zhou, J. Liang, and Y Xiao. SEA-Eval: A Benchmark for Evaluating Self-Evolving Agents Beyond Episodic Assessment, 2026. arXiv:2604.08988.

S. Kim, S. J. Hwang, J. Kim, J. Park, and Y. S Choi. ReGraphRAG: Reorganizing Fragmented Knowledge Graphs for Multi-Perspective Retrieval-Augmented Generation, 2025. In Findings of EMNLP 2025, 5426–5443.

A. Maharana, D.-H. Lee, S. Tulyakov, M. Bansal, F. Barbieri, and Y Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents, 2024. In Proceedings of ACL 2024, 13851–13870.

P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D Manning. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval, 2024. In International Conference on Learning Representations.

N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S Yao. Reflexion: Language Agents with Verbal Reinforcement Learning, 2023. In Advances in Neural Information Processing Systems 36.

R. Stickgold and M. P Walker. Sleep-Dependent Memory Triage: Evolving Generalization through Selective Processing, 2013. Nature Neuroscience, 16: 139–145.

Z. Sun, Z.-H. Deng, J.-Y. Nie, and J Tang. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, 2019. In International Conference on Learning Representations.

Q. Tan, L. Xu, L. Bing, H. T. Ng, and S. M Aljunied. Revisiting DocRED—Addressing the False Negative Problem in Relation Extraction, 2022. In Proceedings of EMNLP 2022, 8472–8487.

H. Trivedi, N. Balasubramanian, T. Khot, and A Sabharwal. MuSiQue: Multihop Questions via Single-hop Question Composition, 2022. Transactions of the Association for Computational Linguistics, 10: 539–554.

T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G Bouchard. Complex Embeddings for Simple Link Prediction, 2016. In Proceedings of ICML 2016, 2071–2080.

Y. Wang, Z. Zhang, M. Chi, K. Yu, Y. Li, M. Peng, B. Tong, C. Zhang, Y. Zhou, and J Li. EvoMemBench: Benchmarking Agent Memory from a Self-Evolving Perspective, 2026. arXiv:2605.18421.

T. Wei, N. Sachdeva, B. Coleman, Z. He, Y. Bei, X. Ning, M. Ai, Y. Li, J. He, E. H. Chi, C. Wang, S. Chen, F. Pereira, W.-C. Kang, and D. Z Cheng. Evo-Memory: Benchmarking LLM Agent Test-time Learning with Self-Evolving Memory, 2025. arXiv:2511.20857, revised 2026.

D. Wu, H. Wang, W. Yu, Y. Zhang, K.-W. Chang, and D Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory, 2025. In International Conference on Learning Representations.

W. Xu, Z. Liang, K. Mei, H. Gao, J. Tan, and Y Zhang. A-MEM: Agentic Memory for LLM Agents, 2025. In Advances in Neural Information Processing Systems 38.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering, 2018. In Proceedings of EMNLP 2018, 2369–2380.

Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M Sun. DocRED: A Large-Scale Document-Level Relation Extraction Dataset, 2019. In Proceedings of ACL 2019, 764–777.

Y. Yao, J. Du, Y. Lin, P. Li, Z. Liu, J. Zhou, and M Sun. CodRED: A Cross-Document Relation Extraction Dataset for Acquiring Knowledge in the Wild, 2021. In Proceedings of EMNLP 2021, 4452–4472.

J. Yue, C. Hu, J. Sheng, Z. Zhou, W. Zhang, T. Liu, L. Guo, and Y Deng. HyperMem: Hypergraph Memory for Long-Term Conversations, 2026. In Proceedings of ACL 2026.

A. Zhao, D. Huang, Q. Xu, M. Lin, Y.-J. Liu, and G Huang. ExpeL: LLM Agents Are Experiential Learners, 2024. Proceedings of the AAAI Conference on Artificial Intelligence, 38(17): 19632–19642.

X. Zhu, Y. Xie, Y. Liu, Y. Li, and W Hu. Knowledge Graph-Guided Retrieval Augmented Generation, 2025. In Proceedings of NAACL 2025, 8912–8924.