LatentGraphRAG: Overcoming the Missing-Edge Problem in Knowledge Graphs with Contextual Dense Retrieval

Anonymous ACL submission

Abstract

Graph-based RAG promises structured multi-hop retrieval, but it faces a missing-edge problem: if extraction, linking, or provenance fails to create the right path, explicit traversal can miss evidence that still exists in the documents. Decomposition-free multi-hop dense retrieval offers a complementary mechanism, connecting documents through contextual embeddings computed from the question and previously retrieved evidence. We study these mechanisms beyond the clean depth-oriented Wikipedia benchmarks where graph-based RAG is usually evaluated, using MuSiQue (Trivedi et al., 2022) as the standard reference setting and extending to FanOutQA (Zhu et al., 2024) for long-tail width-oriented evidence collection and BrowseComp+ (Chen et al., 2025) for noisy web-scale retrieval. Under shared HippoRAG-style graph construction, we first ask whether the same extracted triplets are better accessed as explicit graph edges for traversal or as provenance-linked relational memories retrieved with contextual dense embeddings. Across all three regimes, relational-memory retrieval outperforms traversal over the corresponding graph, showing that extracted relations are more useful when incorporated into latent multi-hop retrieval than when used only as fixed topology. Building on this result, we introduce **LatentGraphRAG**, which combines document retrieval and relational-memory retrieval as two complementary views of the corpus. With diversified memory expansion and calibrated document-memory fusion, LatentGraphRAG achieves the best mean Recall@5 and Recall@10 among evaluated variants. These results suggest that graph-based multi-hop RAG should move from relying on brittle traversal paths alone toward using extracted graph knowledge as relational memory inside multi-hop dense retrieval. Code and data are available in an anonymized repository[1].

[1] anonymized LatentGraphRAG repository

1 Introduction

Retrieval-augmented generation (RAG) systems require reliable multi-hop retrieval to answer complex questions whose evidence is distributed across multiple documents. In such settings, the retriever must identify chains of information that connect several pieces of evidence scattered across a large corpus (Xiong et al., 2021; Zhao et al., 2021; Trivedi et al., 2022).

Two major paradigms have emerged for multi-hop retrieval. The first constructs explicit knowledge graphs from text. Systems such as HippoRAG 2 (Gutiérrez et al., 2025), GraphRAG (Edge et al., 2024), and LightRAG (Guo et al., 2025) extract relational triplets and organize them into graph structures. Retrieval then proceeds by traversing the graph, for example using Personalized PageRank (Gutiérrez et al., 2025). While these systems leverage structured knowledge, they rely on a graph constructed prior to query time. Their effectiveness therefore depends on the quality of the extracted entities, relations, provenance links, and graph connectivity.

The second paradigm performs *decomposition-free multi-hop dense retrieval*. Methods such as MDR (Xiong et al., 2021), BeamDR (Zhao et al., 2021), MoPo (Xia et al., 2025) and GritHopper (Erker et al., 2026) iteratively retrieve documents by encoding the question together with previously retrieved evidence. We refer to this as *contextual retrieval*: retrieval is conditioned not only on the query, but also on the evidence accumulated so far, inducing query- and evidence-conditioned connections through embedding similarity.

Despite addressing the same retrieval problem, graph-based retrieval and decomposition-free dense retrieval have largely been studied as separate paradigms (Luo et al., 2026; Erker et al., 2026). This leaves a central mechanism question unresolved: when relational information has already

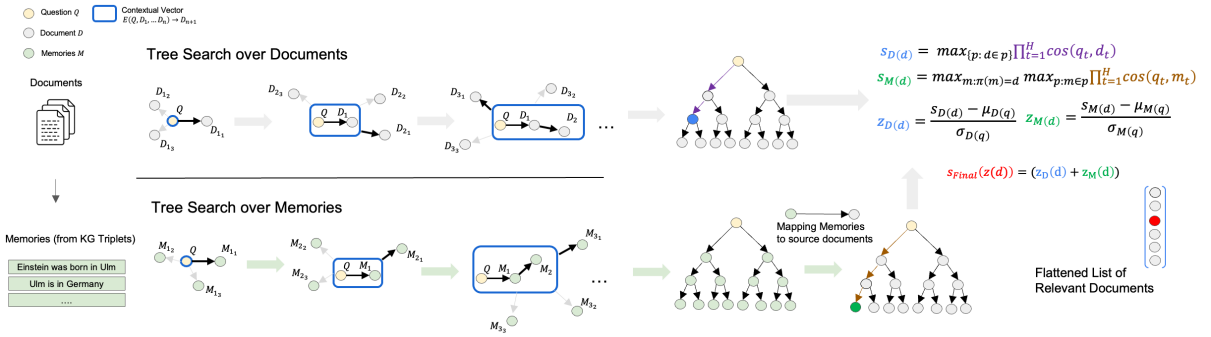

Figure 1: Overview of **LatentGraphRAG**, our framework for retrieving documents and graph-derived relational memories with the same contextual multi-hop mechanism. The shown dual-tree variant retrieves both views separately, maps memories back to source documents, and fuses the resulting document rankings through agreement.

been extracted from a corpus, is it more effective to access it through explicit graph traversal, or through contextual dense retrieval conditioned on the question and previously retrieved evidence? Existing comparisons rarely isolate this question, because they typically vary the retrieval pipeline, indexed units and graph construction at the same time. This distinction is important because graph-based systems conflate two effects: the value of extracted relational information and the value of explicit graph topology. If a graph retriever succeeds, it is unclear whether the gain comes from the extracted facts, the traversal structure, or both. If it fails, the failure may arise because the relation was not extracted, because the correct edge was not connected to the query, or because traversal did not reach the relevant provenance. A controlled comparison must therefore hold the extracted relations fixed while changing only how they are accessed.

A second uncertainty is whether graph-based retrieval conclusions transfer beyond clean Wikipedia-style depth benchmarks. Recent systems such as HippoRAG 2 (Gutiérrez et al., 2025) and G-Reasoner (Luo et al., 2026) evaluate primarily on HotpotQA, 2WikiMultiHopQA, and MuSiQue, where documents are encyclopedic, entities are relatively canonical, evidence chains are short, and provenance is easier to assign. It remains unclear how explicit graph traversal and graph-derived relational knowledge behave in breadth-oriented retrieval and noisy web-scale corpora, where extraction, linking, provenance assignment, and traversal become substantially more brittle.

Our key observation is that extracted graph edges have two separable roles: they encode relational facts, and they define the topology used for traversal. These roles need not be equally useful. A relation may contain relevant evidence even when traversal fails to reach it because of missing links, noisy entity matching, or incorrect provenance. We therefore verbalize graph-derived triplets into provenance-linked *relational memories*, allowing the same extracted information to be either traversed as graph topology or retrieved by a contextual dense retriever.

We instantiate this idea in **LatentGraphRAG**, a decomposition-free multi-hop retrieval framework that extends contextual dense retrieval from documents to graph-derived relational memories and document-memory combinations. It supports document-only, memory-only, joint-index, and dual-tree retrieval with full expansion, BeamDR-style search (Zhao et al., 2021), rank fusion, and agreement-based fusion. This enables a controlled comparison of access interfaces: by holding extracted triplets fixed, we compare explicit traversal over graph topology with contextual dense retrieval over the same relational information. At the same time, the dual-tree variants form concrete retrieval methods for combining document evidence and relational memories, with Z-Agreement calibrating their scores before fusion.

Because our memories use the same HippoRAG-style triplets as HippoRAG 2 (Gutiérrez et al., 2025), we can compare two ways of accessing the same extracted relations: explicit graph traversal and contextual dense retrieval. We evaluate three stress regimes: MuSiQue (Trivedi et al., 2022) for depth-oriented Wikipedia chains, FanOutQA (Zhu et al., 2024) for breadth-oriented evidence collection, and BrowseComp+ (Chen et al., 2025) for noisy web-scale retrieval. This leads to four research questions:

- **RQ1:** When extracted triplets are fixed, is graph-derived knowledge more effective through traversal topology or contextual retrieval?
- **RQ2:** How do documents, relational memories, and their combinations behave across depth, breadth, and noisy web-scale retrieval?
- **RQ3:** How do search strategy and score calibration affect document-memory retrieval?
- **RQ4:** How much do missing or underspecified graph edges limit relational-memory retrieval, and can document retrieval bypass them?

Our results challenge the common assumption that extracted knowledge graphs should primarily be used as traversal structures. First, when the triplets are held fixed, retrieving their verbalized forms with a contextual dense retriever consistently outperforms traversing them as graph edges. This shows that extracted relational knowledge remains useful, but that its most effective interface is often retrieval over relational memories rather than explicit topology. Second, the strongest results come from combining relational memories with full document retrieval, indicating that compact extracted facts and broader document context address different failure modes. Third, the oracle analysis shows that missing or underspecified graph edges explain a meaningful subset of memory-retrieval failures, while documents can bypass some of these failures by preserving context absent from extracted triplets.

2 Related Work

Generation-driven and retrieval-first multi-hop retrieval. Generation-driven systems such as ReAct (Yao et al., 2023), IRCoT (Trivedi et al., 2023), and Self-RAG (Asai et al., 2024) combine retrieval with query reformulation, tool use, and answer synthesis. We instead focus on retrieval-first methods that return ranked evidence documents before generation, allowing the retrieval mechanism itself to be evaluated directly with Recall@k.

Multi-hop dense retrieval. Retrieval-first dense methods such as MDR (Xiong et al., 2021), BeamDR (Zhao et al., 2021), and GritHopper (Erker et al., 2026) retrieve evidence iteratively by encoding the question together with previously retrieved documents. LatentGraphRAG extends this principle from document-only retrieval to graph-derived relational memories and studies how both views can be combined through query-conditioned latent search.

Graph-based and hybrid RAG. Graph-based RAG systems such as HippoRAG 2 (Gutiérrez et al., 2025), GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2025), KAG (Liang et al., 2025), GFM-RAG (Luo et al., 2025), and G-Reasoner (Luo et al., 2026) extract entities and relations from text and retrieve evidence through graph traversal or relevance propagation. While this provides an interpretable retrieval substrate, performance depends on graph construction quality, including entity extraction, relation extraction, linking, provenance assignment, and graph connectivity. These assumptions can become brittle on noisy web corpora.

Hybrid systems often use dense retrieval to select seed documents, entities, or triplets before continuing retrieval through explicit graph traversal. In contrast, LatentGraphRAG asks whether the same extracted triplets are more useful as dense retrievable *relational memories* than as static graph edges. This separation matters because graph-based retrieval can fail either because the required relation was not extracted, or because the relation exists but traversal does not reach it due to linking, provenance, or connectivity errors.

3 Problem Setup

Let $D = \{d_i\}_{i=1}^{|D|}$ denote a corpus of documents or document chunks. Given a multi-hop question q, the retrieval task is to return a ranked list of documents such that the evidence required to answer q appears among the top-ranked results.

Graph-based RAG first constructs a knowledge graph from D. Let

$$E = \{(h_j, r_j, t_j)\}_{j=1}^{|E|}$$

denote extracted relational edges with provenance mapping $\pi : E \to D$. Explicit graph retrieval uses E as a fixed traversal structure, making retrieval depend on extraction, linking, provenance, and connectivity.

Our key observation is that the same edge set can also be interpreted as a collection of retrievable relational memories. Each edge (h, r, t) is verbalized into a short textual statement

$$m_j = \text{verbalize}(h_j, r_j, t_j),$$

while retaining the same provenance mapping $\pi(m_j) \to D$. This lets us compare two access

interfaces for the same graph-derived information: explicit traversal over fixed graph edges versus contextual dense retrieval over verbalized edge memories.

All methods ultimately produce a ranked list of documents. We evaluate retrieval using Recall@k after mapping retrieved chunks or memories back to source documents.

4 Method

LatentGraphRAG applies decomposition-free retrieval to different corpus representations. Following MDR, BeamDR, and GritHopper, it iteratively encodes the question with previously retrieved evidence; each contextual embedding induces the next retrieval step by nearest-neighbor search. We call this query-conditioned connectivity a latent graph.

LatentGraphRAG provides a common interface for documents, relational memories, joint indices, and dual trees. Search uses full expansion or BeamDR-style pruning, and retrieved paths are evaluated separately or fused through rank-based or score-based document aggregation. This lets us isolate retrieval substrate, search strategy, and use of graph-derived information.

4.1 Contextual Dense Retrieval as Latent Graph Construction

Contextual dense retrieval expands partial evidence paths by repeatedly encoding the question and retrieved evidence. Let I be the active index, containing documents D, memories M, or $D \cup M$. At hop t, the current path contains items $x_1, \ldots, x_{t-1}$, where each x_i is a document or memory, and the retrieval context is

$$c_t = [q; x_1; \ldots; x_{t-1}],$$

with $[\cdot;\cdot]$ denoting textual concatenation. Candidates $x \in I$ are scored by cosine similarity:

$$w_t(x) = \cos(f_\theta(c_t), f_\theta(x)).$$

A retrieval path $\langle x_1, \ldots, x_t \rangle$ can be interpreted as a path in a query-conditioned latent graph: an edge from the current context to a candidate item exists when the contextual encoder assigns it high similarity. Following BeamDR-style multi-hop dense retrieval (Zhao et al., 2021), we score a path by accumulating the retrieval scores along the path:

$$S(\langle x_1, \ldots, x_t \rangle) = \prod_{j=1}^{t} w_j(x_j).$$

The product favors paths whose evidence remains consistently compatible with the evolving context. While BeamDR applies this to two-hop document retrieval, we use it for bounded multi-hop expansion over documents, memories, or their union; variants differ only in indexed units, document-level projection, and fusion.

4.2 Relational Memory Index

We convert extracted KG edges into textual relational memories. Each edge (h, r, t) is verbalized as

$$m = \text{verbalize}(h, r, t)$$

with provenance $\pi(m) \to D$. The memory index M contains the same edge-level information as the graph, but exposes each edge as a retrievable textual memory rather than as a traversal link.

This representation lets us test whether extracted KG edges are useful because they define explicit graph topology or because they provide compact relational evidence that can be retrieved under the current query-and-path context.

4.3 LatentGraphRAG Instantiations

LatentGraphRAG instantiations are orthogonal to search strategy: full expansion or BeamDR-style pruning can be applied over documents, memories, or both.

Document-only. Retrieval is performed over D, inducing a query-conditioned latent graph over documents.

Memory-only. Retrieval is performed over M, and retrieved memories are mapped to source documents through π. This directly compares dense retrieval over graph-derived information with traversal over the same information.

Joint document-memory. Documents and memories are merged into one index,

$$I_{\text{joint}} = D \cup M.$$

Retrieved memories are mapped to documents through π.

Dual latent graphs. Separate document and memory trees produce document scores $s_D(d)$ and projected memory scores

$$s_M(d) = \max_{m:\pi(m)=d} s_M(m).$$

The rankings are then combined through document-memory fusion.

4.4 Search Strategies

We use three expansion strategies, with maximum depth H, branching factor b, beam size B, and dual-tree branching factors b_d and b_t.

Full expansion. Each partial path retrieves the top-b candidates up to depth H, avoiding early pruning at higher retrieval cost.

BeamDR-style beam search. Following BeamDR (Zhao et al., 2021), each hop expands the current beam and keeps the top-B paths by cumulative score.

MMR-decoded expansion. Full expansion selects the top-b candidates by cosine similarity, which can yield redundant frontiers in clustered memory indices. We therefore adapt Maximal Marginal Relevance (Carbonell and Goldstein, 1998) to memory-tree expansion:

$$\mathrm{MMR}(x) = \lambda w_t(x) - (1-\lambda) \max_{y \in \mathcal{R}_t} \mathrm{sim}(x, y),$$

where $\mathcal{R}_t$ contains items already selected for the current expansion, sim is embedding cosine similarity, and $\lambda \in [0, 1]$ trades relevance against diversity. We apply MMR only to the memory tree, where near-duplicate clustering is strongest.

Cross-view beam search. For dual-index retrieval, cross-view beam keeps one beam of partial states instead of separate document and memory trees. Each state queries both indices and ranks candidates jointly, testing whether document-memory interaction is more useful during search or after independent view expansion.

4.5 Document-Memory Fusion

Both views are projected to document rankings before fusion. We introduce **Z-Agreement**, a parameter-free dual-tree fusion method for cross-substrate calibration: documents and verbalized memories produce different cosine-similarity distributions, so raw additive fusion assumes comparability between scores from different retrieval units.

Z-Agreement normalizes per-query, per-view scores to zero mean and unit variance before additive combination:

$$s^z_{\text{final}}(d) = \frac{s_D(d) - \mu_D(q)}{\sigma_D(q)} + \frac{s_M(d) - \mu_M(q)}{\sigma_M(q)},$$

where $\mu_D(q), \sigma_D(q)$ and $\mu_M(q), \sigma_M(q)$ are computed over each view's retrieved candidates for query q; fusion is over the union of documents, with absent view scores set to zero. Per-query normalization is established in metasearch fusion (Montague and Aslam, 2001); we apply it to cross-substrate dense retrieval. This allows agreement between document and memory views to affect the ranking without assuming that their raw cosine scores are on the same scale.

Ablations of Z-Agreement. Without normalization, the additive agreement ablation scores documents over the union of both rankings as

$$s_{\text{final}}(d) = s_D(d) + s_M(d),$$

again treating absent scores as zero. Reciprocal Rank Fusion (Cormack et al., 2009) avoids score-scale mismatch by using ranks only, but discards encoder score magnitudes. Z-Agreement preserves score information while reducing cross-substrate scale mismatch.

4.6 Comparison to Explicit Graph Traversal

Explicit graph traversal is the external comparison point for the LatentGraphRAG family. HippoRAG 2 treats extracted triplets as edges in a fixed graph and propagates relevance through this structure. LatentGraphRAG instead treats documents, memories, or both as retrievable units and applies decomposition-free retrieval over them. Comparing HippoRAG 2 with memory-only LatentGraphRAG holds the extracted triplets fixed while changing how those triplets are accessed. Comparing the LatentGraphRAG variants then shows how documents, memories, search strategies, and fusion mechanisms interact within the same framework.

5 Experimental Setup

Datasets. We evaluate on three datasets that stress distinct multi-hop retrieval regimes. **MuSiQue** (Trivedi et al., 2022) is a depth-oriented Wikipedia benchmark with compositional evidence chains. It is a standard evaluation setting for graph-based and decomposition-free multi-hop retrieval, and serves as the closest point of comparison to prior GraphRAG work. **FanOutQA** (Zhu et al., 2024) is a breadth-oriented multi-document QA benchmark whose questions require collecting evidence over lists of entities and attributes. Unlike short depth-oriented chains, FanOutQA requires longer nonlinear decompositions, with an average of seven hops and evidence spanning

seven documents. **BrowseComp+** (Chen et al., 2025) is a noisy web-scale retrieval benchmark with over 100K heterogeneous documents, making extraction, linking, and provenance assignment substantially harder than in clean Wikipedia settings.

Together, these datasets let us provide a controlled stress test of graph-based multi-hop retrieval beyond the clean depth-oriented Wikipedia settings emphasized in prior work, covering breadth-oriented evidence collection and noisy web-scale retrieval.

Graph construction. All graph-based and memory-based methods use the same HippoRAG 2 graph construction pipeline (Gutiérrez et al., 2025). The resulting triplets are either traversed as graph edges in HippoRAG 2 or verbalized as relational memories in LatentGraphRAG, controlling the extracted relational substrate while varying the access interface. We use Qwen/Qwen3-Next-80B-A3B-Instruct[2] for HippoRAG graph construction.

Baselines. We compare single-hop dense retrieval, HippoRAG 2 traversal, document-only, memory-only, joint, and dual-tree LatentGraphRAG variants. For MuSiQue, we also report prior graph-based results for context, while the controlled comparison uses the same graph construction setting. The single-hop baseline is Qwen3-Embedding-8B (Zhang et al., 2025); Appendix D also tests it as the contextual encoder.

6 Experiments

Table 1 presents the controlled comparison with shared extracted triplets. Memory-only LatentGraphRAG consistently outperforms HippoRAG 2, showing that the same relations are more effective when accessed through contextual dense retrieval than when traversed as fixed topology. MMR-Tree + Z-Agreement achieves the best mean Recall@5 and Recall@10.

6.1 RQ1: Graph-Derived Knowledge Is More Useful under Contextual Dense Retrieval

We first hold the extracted triplets fixed and compare two access mechanisms: HippoRAG 2 traverses them as graph edges, whereas memory-only LatentGraphRAG retrieves their verbalized forms with a contextual dense retriever. Memory-only retrieval outperforms HippoRAG 2 across all three regimes: 76.15 vs. 74.47 R@5 on MuSiQue, 31.45/44.14 vs. 26.07/36.86 R@5/R@10 on FanOutQA, and 9.28/13.19 vs. 8.12/12.83 R@5/R@10 on BrowseComp+. The BrowseComp+ gap is small in absolute terms, consistent with the diffuse and weakly connected entity space observed in Appendix A. Thus, the same extracted relations are more effective when accessed through query-and-path-conditioned dense retrieval than when traversed as fixed topology.

6.2 RQ2–RQ3: Robust Document-Memory Retrieval through Diversification and Calibration

Across regimes, **MMR-Tree + Z-Agreement** achieves the best mean Recall@5 (44.65) and Recall@10 (54.26), but the strongest individual variant depends on the retrieval regime. Cross-view Beam is best on MuSiQue R@5, document-only Full Tree is strongest on BrowseComp+, and memory-heavy variants perform best on FanOutQA. This shows that MMR-Tree + Z-Agreement is robust on average rather than uniformly dominant; Appendix C gives calibration diagnostics.

On MuSiQue, the setting is closest to the distribution for which decomposition-free multi-hop dense retrieval was designed as depth-oriented Wikipedia evidence chains. BeamDR-style variants are strong, and Cross-view Beam obtains the best MuSiQue R@5 of 83.21, showing that cumulative contextual path scores are informative when encoder similarities are well calibrated. This also holds across documents and memories: Full Dual-Tree + Agreement reaches 82.32 R@5 and 90.36 R@10, while MMR-Tree + Z-Agreement remains similarly strong at 82.12 R@5 and 90.02 R@10. Thus, in the clean depth-oriented regime, cumulative aggregation over document and memory paths is effective.

BrowseComp+ changes this picture. The corpus is heterogeneous, web-scale, and substantially further from the clean Wikipedia-style regime, making cumulative path scores less reliable. Document-only full expansion is strongest, while HippoRAG 2 and memory-only retrieval remain weak. This is not because extraction is small: BrowseComp+ yields 5.45M unique triples, but these relations are spread over far more canonical entities per document than MuSiQue (17.9 vs. 4.9; Appendix A), making topology-based propagation diffuse. Ap-

[2] Qwen/Qwen3-Next-80B-A3B-Instruct

Method	MuSiQue		BrowseComp+		FanOutQA		Mean	
	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
Single-Hop Embedding Baseline								
Qwen3-Embedding-8B (Zhang et al., 2025)	62.10	69.92	14.60	20.30	20.76	28.55	32.49	39.59
Explicit Graph Traversal over Extracted Triplets								
HippoRAG 2 reproduced	74.47	81.94	8.12	12.83	26.07	36.86	36.22	43.88
LatentGraphRAG Family using GritHopper (Erker et al., 2026) as Encoder								
Document-Only LatentGraphRAG								
Full Tree Expansion (b=3, H=4)	79.73	87.73	**21.18**	28.87	22.40	29.24	41.10	48.61
MMR-Tree (b=3, H=4)	69.99	85.12	20.07	**29.21**	24.12	35.65	38.06	49.99
BeamDR (B=5)	80.03	84.60	17.23	19.29	17.94	19.30	38.40	41.06
BeamDR (B=40)	78.94	87.97	11.51	19.56	15.58	22.26	35.34	43.26
Memory-Only LatentGraphRAG								
Memory-Only Full Expansion (b=3, H=4)	76.15	84.40	9.28	13.19	31.45	44.14	38.96	47.24
Joint-Index LatentGraphRAG (Documents + Memories)								
Full Tree Expansion Joint (b=5)	76.14	84.05	16.62	27.61	30.52	40.66	41.09	50.77
MMR-Tree Joint (b=5, H=4)	74.72	82.95	14.82	28.01	31.78	44.95	40.44	51.97
BeamDR-Joint (B=5)	72.18	73.57	17.05	17.05	24.54	27.22	37.92	39.28
BeamDR-Joint (B=40)	77.82	82.71	12.49	17.55	25.42	34.03	38.58	44.76
Dual-Tree LatentGraphRAG with RRF								
Dual RRF, Full Expansion (b_d=3, b_t=5)	79.64	89.49	16.31	25.65	27.85	38.82	41.27	51.32
Dual-Tree LatentGraphRAG with Cross-view Agreement								
Cross-view Beam (B=5, H=4)	83.17	85.85	17.05	18.31	21.07	22.57	40.43	42.24
Cross-view Beam (B=10, H=4)	**83.21**	88.51	15.63	18.57	22.57	26.66	40.47	44.58
MMR-Tree + Agreement (b_d=3, b_t=5)	82.16	89.73	17.95	24.18	25.37	36.82	41.83	50.24
Full Dual-Tree + Z-Agreement (b_d=3, b_t=5)	82.29	90.32	18.62	25.94	32.76	45.18	44.56	53.81
Full Dual-Tree + Agreement (b_d=3, b_t=5)	82.32	**90.36**	17.18	23.33	25.34	36.23	41.61	49.97
MMR-Tree + Z-Agreement (b_d=3, b_t=5)	82.12	90.02	18.62	25.94	**33.22**	**46.83**	**44.65**	**54.26**

Table 1: Evaluation of HippoRAG 2 graph traversal against LatentGraphRAG instantiations across retrieval regimes. All graph-based and memory-based methods use the same HippoRAG-style extracted triplets, controlling the extracted relational substrate while comparing traversal-based and contextual dense access. MuSiQue tests depth-oriented Wikipedia retrieval, BrowseComp+ tests noisy web-scale retrieval, and FanOutQA tests width-oriented retrieval. Dense retrieval over verbalized memories consistently outperforms explicit traversal over the same graph-derived information, while MMR-Tree + Z-Agreement achieves the best mean Recall@5 and Recall@10.

Method	R@5
Non-structure Methods	
BM25 (Robertson and Zaragoza, 2009)	41.2
ColBERTv2 (Santhanam et al., 2022)	49.2
Qwen3-Emb (8B) (Zhang et al., 2025)	62.1
Graph-enhanced Methods	
RAPTOR (Sarthi et al., 2024)	45.3
GraphRAG (MS) (Edge et al., 2024)	49.3
LightRAG (Guo et al., 2025)	34.7
KAG (Liang et al., 2025)	62.4
HippoRAG (Gutierrez et al., 2024)	53.2
HippoRAG 2 (Gutiérrez et al., 2025)	74.2
SubgraphRAG (Li et al., 2025)	48.1
G-retriever (He et al., 2024)	43.5
GFM-RAG (Luo et al., 2025)	57.6
G-Reasoner (Luo et al., 2026)	74.9
LatentGraphRAG	**82.32**
LatentGraphRAG Doc Only	**79.73**

Table 2: Contextual MuSiQue comparison (Recall@5); the controlled comparison is Table 1.

Method	Overall R@5	Overall R@10	Missing-Edge Full Recovery
No oracle knowledge			
Memory-only Full Exp.	76.15	84.40	0.0%
Doc-only Full Exp.	79.73	87.73	50.8%
BeamDR (B=5)	80.03	84.60	44.8%
Full Dual-Tree + Agree.	**82.32**	**90.36**	**50.3%**
Oracle setting (†): ground-truth (GT) triplets injected			
GT Triplet-only†	85.34	90.48	100.0%
GT Full Tree Exp. + Cross View	**86.07**	**93.14**	84.0%

Table 3: Oracle missing-edge analysis on MuSiQue. The oracle-sensitive subgroup contains 181 queries where standard memory-only retrieval does not recover all gold evidence documents at R@10, but memory-only retrieval with injected oracle triplets does. *Full Recovery* reports how often each method retrieves all gold evidence documents for this subgroup.

pendix Table 6 shows that this first web-scale graph stress test requires 321.7M OpenIE tokens and up to 3,252 GPU-hours of offline construction, yet document-only LatentGraphRAG performs best on BrowseComp+ without requiring graph construction.

FanOutQA shows why relational memories remain valuable. Its questions often require broad entity–attribute evidence collection based on tables rather than a short bridge chain. Here, memory-only retrieval reaches 31.45 R@5 and 44.14 R@10, outperforming document-only retrieval at 22.40 R@5 and 29.24 R@10. MMR-Tree + Z-Agreement further improves this to 33.22 R@5 and 46.83 R@10, showing that diversified memory expan-

sion preserves the strength of relational memories while still benefiting from document evidence.

Together, these results motivate the final design. Full expansion avoids premature commitment to misleading high-scoring paths in noisy or width-oriented settings. MMR reduces redundant memory expansion by penalizing already-retrieved neighbors, while Z-Agreement normalizes document and memory scores per query before fusion. MMR-Tree + Z-Agreement therefore keeps the complementary document and memory views, but corrects the redundancy and score-calibration issues exposed by the ablations.

6.3 RQ4: Diagnosing Missing Relational Edges

The results above suggest that relational memories are useful when the required relations are present, but brittle when graph construction fails to extract them. We therefore perform an oracle missing-edge analysis to estimate how much performance is lost due to absent or underspecified relational memories. This analysis is not intended as a deployable LatentGraphRAG variant. Instead, it probes the central limitation of graph-derived retrieval: if the required relation is missing from the extracted graph, neither traversal nor memory retrieval can recover it from the relational index alone.

Table 3 reports results on the 181 queries where standard memory-only retrieval fails but oracle triplets enable full recovery. By construction, standard memory-only retrieval recovers 0% of these cases, while oracle triplets raise recovery to 100%.

Document-only LatentGraphRAG recovers 50.8% of these cases without oracle knowledge, showing that contextual dense retrieval over full documents can bypass many relational coverage failures. Full Dual-Tree + Agreement recovers a comparable fraction while achieving stronger overall Recall@5 and Recall@10. The important comparison is not whether fusion recovers more of this oracle-defined subgroup than document-only retrieval: the subgroup is selected precisely by failures of the standard memory index. Rather, the result shows that document retrieval and memory retrieval fail for different reasons. Documents can sometimes bypass missing relational edges because they preserve broader context, while memories can become highly effective when the relevant relations are available.

Even in the oracle setting, combining oracle memories with document retrieval improves over oracle triplets alone, reaching 86.07 Recall@5 and 93.14 Recall@10 compared to 85.34 and 90.48 for Ground-Truth Triplet-only. This suggests that relational triplets, even when answer-aware, do not fully replace document context. Triplets can abstract away disambiguating information or overlap with distractor evidence, while document retrieval preserves broader context that helps distinguish relevant evidence from superficially matching relations.

7 Conclusion

We presented LatentGraphRAG, a framework for isolating how graph-derived knowledge should be accessed in retrieval-first multi-hop RAG. By holding HippoRAG-style triplets fixed, we separate the value of extracted relational facts from the value of explicit graph topology. Across depth, breadth, and noisy web-scale regimes, retrieving verbalized triplets with a contextual dense retriever consistently outperforms traversing the same extracted graph.

The results also show that documents and relational memories play complementary roles. Documents are most robust on noisy web-scale retrieval, while relational memories are strongest for breadth-oriented evidence collection. MMR-Tree + Z-Agreement combines both views by diversifying memory expansion and calibrating document-memory scores before fusion, achieving the best average performance.

Our oracle analysis further shows that missing or underspecified edges explain a meaningful subset of memory-retrieval failures, but that oracle triplets still do not fully replace document context. Overall, our findings suggest a shift in graph-based RAG: extracted graph knowledge should not only define brittle traversal paths, but should also serve as compact relational memory accessed through contextual dense retrieval and fused with document evidence.

Limitations

This work studies retrieval-first multi-hop RAG systems: methods that return a ranked list of evidence documents before any answer generation step. We therefore evaluate retrieval quality directly with document-level Recall@k, following the setting used by multi-hop dense retrieval and graph-based retrieval work. This scope intention-

ally differs from agentic or generation-interleaved systems, where a language model repeatedly reasons, reformulates queries, calls tools, and decides when to stop. Such systems are important, but they conflate retrieval quality with prompting, reasoning, tool-use policies, and answer generation. Our goal is instead to isolate the retrieval mechanism itself.

Our controlled graph comparison uses HippoRAG 2 as a recent state-of-the-art traversal baseline. Using a shared graph construction pipeline is necessary for the main ablation: we compare whether the same extracted relational information is more effective as fixed topology or as retrievable memories. Our conclusions therefore apply most directly to this HippoRAG-style construction setting. Although the triplets are fixed, traversal and contextual dense retrieval still differ in scoring and inductive bias.

LatentGraphRAG also inherits limitations from the extracted relational memories. If relevant relations are not extracted, are underspecified, are linked to the wrong provenance document, or are fragmented across noisy entities, memory-only retrieval cannot recover them from the relational index alone. We view this not only as a limitation but also as the core phenomenon studied in this paper: explicit graph and memory-based retrieval are both bounded by the completeness and quality of the extracted relational substrate. Our oracle analysis shows that missing or incomplete relational edges account for a meaningful subset of failures, while document retrieval can bypass some of these failures by preserving broader textual context.

Finally, our experiments cover three complementary retrieval regimes: MuSiQue for depth-oriented Wikipedia retrieval, FanOutQA for width-oriented evidence collection, and BrowseComp+ for noisy web-scale retrieval. These settings allow us to test whether conclusions from clean multi-hop benchmarks transfer to broader and noisier corpora, but they do not cover all domains, languages, or deployment settings. We also do not evaluate downstream generation quality, since the retrieved document list is always intended to be passed to a downstream model. Future work can study how the retrieval improvements measured here translate into answer accuracy, attribution quality, and interactive agentic pipelines.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. 2025. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Team Cohere, :, Aakanksha, Arash Ahmadian, Marwan Ahmed, Jay Alammar, Milad Alizadeh, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, Zahara Aviv, Sammie Bae, Saurabh Baji, Alexandre Barbet, Max Bartolo, Björn Bebensee, Neeral Beladia, Walter Beller-Morales, Alexandre Bérard, Andrew Berneshawi, Anna Bialas, Phil Blunsom, Matt Bobkin, Adi Bongale, Sam Braun, Maxime Brunet, Samuel Cahyawijaya, David Cairuz, Jon Ander Campos, Cassie Cao, Kris Cao, Roman Castagné, Julián Cendrero, Leila Chan Currie, Yash Chandak, Diane Chang, Giannis Chatziveroglou, Hongyu Chen, Claire Cheng, Alexis Chevalier, Justin T. Chiu, Eugene Cho, Eugene Choi, Eujeong Choi, Tim Chung, Volkan Cirik, Ana Cismaru, Pierre Clavier, Henry Conklin, Lucas Crawhall-Stein, Devon Crouse, Andres Felipe Cruz-Salinas, Ben Cyrus, Daniel D'souza, Hugo Dalla-Torre, John Dang, William Darling, Omar Darwiche Domingues, Saurabh Dash, Antoine Debugne, Théo Dehaze, Shaan Desai, Joan Devassy, Rishit Dholakia, Kyle Duffy, Ali Edalati, Ace Eldeib, Abdullah Elkady, Sarah Elsharkawy, Irem Ergün, Beyza Ermis, Marzieh Fadaee, Boyu Fan, Lucas Fayoux, Yannis Flet-Berliac, Nick Frosst, Matthias Gallé, Wojciech Galuba, Utsav Garg, Matthieu Geist, Mohammad Gheshlaghi Azar, Ellen Gilsenan-McMahon, Seraphina Goldfarb-Tarrant, Tomas Goldsack, Aidan Gomez, Victor Machado Gonzaga, Nithya Govindarajan, Manoj Govindassamy, Nathan Grinsztajn, Nikolas Gritsch, Patrick Gu, Shangmin Guo, Kilian Haefeli, Rod Hajjar, Tim Hawes, Jingyi He, Sebastian Hofstätter, Sungjin Hong, Sara Hooker, Tom Hosking, Stephanie Howe, Eric Hu, Renjie Huang, Hemant Jain, Ritika Jain, Nick Jakobi, Madeline Jenkins, JJ Jordan, Dhruti Joshi, Jason Jung, Trushant Kalyanpur, Siddhartha Rao Kamalakara, Julia Kedrzycki, Gokce Keskin, Edward Kim, Joon

Kim, Wei-Yin Ko, Tom Kocmi, Michael Kozakov, Wojciech Kryściński, Arnav Kumar Jain, Komal Kumar Teru, Sander Land, Michael Lasby, Olivia Lasche, Justin Lee, Patrick Lewis, Jeffrey Li, Jonathan Li, Hangyu Lin, Acyr Locatelli, Kevin Luong, Raymond Ma, Lukáš Mach, Marina Machado, Joanne Magbitang, Brenda Malacara Lopez, Aryan Mann, Kelly Marchisio, Olivia Markham, Alexandre Matton, Alex McKinney, Dominic McLoughlin, Jozef Mokry, Adrien Morisot, Autumn Moulder, Harry Moynehan, Maximilian Mozes, Vivek Muppalla, Lidiya Murakhovska, Hemangani Nagarajan, Alekhya Nandula, Hisham Nasir, Shauna Nehra, Josh Netto-Rosen, Daniel Ohashi, James Owers-Bardsley, Jason Ozuzu, Dennis Padilla, Gloria Park, Sam Passaglia, Jeremy Pekmez, Laura Penstone, Aleksandra Piktus, Case Ploeg, Andrew Poulton, Youran Qi, Shubha Raghvendra, Miguel Ramos, Ekagra Ranjan, Pierre Richemond, Cécile Robert-Michon, Aurélien Rodriguez, Sudip Roy, Sebastian Ruder, Laura Ruis, Louise Rust, Anubhav Sachan, Alejandro Salamanca, Kailash Karthik Saravanakumar, Isha Satyakam, Alice Schoenauer Sebag, Priyanka Sen, Sholeh Sepehri, Preethi Seshadri, Ye Shen, Tom Sherborne, Sylvie Shang Shi, Sanal Shivaprasad, Vladyslav Shmyhlo, Anirudh Shrinivason, Inna Shteinbuk, Amir Shukayev, Mathieu Simard, Ella Snyder, Ava Spataru, Victoria Spooner, Trisha Starostina, Florian Strub, Yixuan Su, Jimin Sun, Dwarak Talupuru, Eugene Tarassov, Elena Tommasone, Jennifer Tracey, Billy Trend, Evren Tumer, Ahmet Üstün, Bharat Venkitesh, David Venuto, Pat Verga, Maxime Voisin, Alex Wang, Donglu Wang, Shijian Wang, Edmond Wen, Naomi White, Jesse Willman, Marysia Winkels, Chen Xia, Jessica Xie, Minjie Xu, Bowen Yang, Tan Yi-Chern, Ivan Zhang, Zhenyu Zhao, and Zhoujie Zhao. 2025. Command a: An enterprise-ready large language model. Preprint arXiv:2504.00698.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

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

Justus-Jonas Erker, Nils Reimers, and Iryna Gurevych. 2026. GRITHopper: Decomposition-free multi-hop dense retrieval. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–94, Rabat, Morocco. Association for Computational Linguistics.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, Suzhou, China. Association for Computational Linguistics.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, Wen Zhang, and Huajun Chen. 2025. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 334–343, New York, NY, USA. Association for Computing Machinery.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2025. GFM-RAG: Graph foundation model for retrieval augmented generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Linhao Luo, Zicheng Zhao, Junnan Liu, Zhangchi Qiu, Junnan Dong, Serge Panev, Chen Gong, Thuy-Trang Vu, Gholamreza Haffari, Dinh Phung, Alan Wee-Chung Liew, and Shirui Pan. 2026. G-reasoner: Foundation models for unified reasoning over graph-structured knowledge. In *The Fourteenth International Conference on Learning Representations*.

Mark H. Montague and Javed A. Aslam. 2001. Relevance score normalization for metasearch. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*, pages 427–433. ACM.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. ReasonIR: Training retrievers for reasoning tasks. In *Second Conference on Language Modeling*.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. 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)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Zehua Xia, Yuyang Wu, Yiyun Xia, and Cam Tu Nguyen. 2025. Momentum posterior regularization for multi-hop dense retrieval. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8255–8271, Abu Dhabi, UAE. Association for Computational Linguistics.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. Preprint, arXiv:2506.05176.

Retriever	Recall@5	nDCG@10
Evidence Document Retrieval		
BM25 (Robertson and Zaragoza, 2009)	1.2	1.6
Qwen3-Embed-0.6B (Zhang et al., 2025)	6.2	8.0
Qwen3-Embed-4B (Zhang et al., 2025)	9.8	14.0
Qwen3-Embed-8B (Zhang et al., 2025)	14.5	20.3
ReasonIR-8B (Shao et al., 2025)	12.2	16.8
LatentGraphRAG on Docs	**21.18**	**26.98**
Gold Document Retrieval		
BM25 (Robertson and Zaragoza, 2009)	1.4	1.7
Qwen3-Embed-0.6B (Zhang et al., 2025)	8.5	7.4
Qwen3-Embed-4B (Zhang et al., 2025)	13.0	13.6
Qwen3-Embed-8B (Zhang et al., 2025)	18.5	19.5
ReasonIR-8B (Shao et al., 2025)	15.3	15.5
LatentGraphRAG on Docs	**27.28**	**25.51**

Table 4: Retrieval results on BrowseComp+.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. 2021. Multi-step reasoning over unstructured text with beam dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4635–4641, Online. Association for Computational Linguistics.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37, Bangkok, Thailand. Association for Computational Linguistics.

A Knowledge Graph Scale and Construction Cost

Table 5 highlights that BrowseComp+ differs from MuSiQue not only in corpus scale but also in graph geometry. Although BrowseComp+ contains many more relational memories overall, those memories are distributed across a much larger entity space: 17.86 canonical entities per document compared with 4.94 on MuSiQue. The extracted topology is therefore more diffuse — graph propagation must rank evidence through a much larger set of weakly connected entity neighborhoods, so additional triples do not translate directly into stronger graph-based retrieval.

B Oracle Triplet Experiment

To measure the impact of missing graph edges, we construct an oracle knowledge setting in which high-quality relational triplets are available for all gold evidence documents.

Triplet generation. For each query in the MuSiQue development set, we provide a language

Dataset	Docs	Unique triples	Unique ent.	Ent/Doc	triples / 1K tok.
MuSiQue	11,656	128,159	57,536	4.94	132.92
FanOutQA	40,209	1,203,683	374,128	9.30	26.83
BrowseComp+	107,045	5,450,875	1,911,428	**17.86**	74.68

Table 5: Relational-memory index statistics. *Docs* is the retrieval corpus size; *Unique ent.* counts canonicalized entities after the HippoRAG NER step. The structural difference between BrowseComp+ and MuSiQue is not absolute index size but the *entities-per-document ratio*: ∼17.9 on BrowseComp+ versus 4.9 on MuSiQue — a 3.6× larger fan-out. Each canonical entity is therefore anchored to far fewer documents, so the entity graph is much more fragmented and per-entity neighborhoods provide less structure for PPR-style propagation or entity-anchored memory retrieval.

Stage	Input tokens	Output tokens	Total tokens
NER extraction	89,761,155	22,128,638	111,889,793
Triple extraction	143,349,786	66,485,708	209,835,494
Total OpenIE	233,110,941	88,614,346	321,725,287

Runtime measure	Hours
Total log wall-clock span	438.10
OpenIE wall-clock span	406.49
OpenIE worker batch time	268.97

Table 6: BrowseComp+ OpenIE token usage and runtime. Graph construction processes 321.7M total OpenIE tokens. The OpenIE jobs ran for 406.49 wall-clock hours with two parallel vLLM servers on eight GPUs, corresponding to up to approximately 3,252 GPU-hours of allocated graph-construction compute.

model with the multi-hop question, all gold evidence documents, and the correct answer. The model, Command A (Cohere et al., 2025), is prompted to generate relational triplets that capture the specific factual connections required to answer the question. These oracle triplets are more precise and complete than the triplets extracted by the standard HippoRAG 2 pipeline, which processes documents independently without knowledge of query relevance.

Injection into the retrieval index. The oracle triplets are verbalized and added to the standard relational memory index, supplementing (not replacing) the existing triplets extracted from the full corpus. Retrieval is then performed with the augmented index using the same retrieval architecture and hyperparameters.

Prompt template. Figure 2 shows the full prompt used for oracle triplet generation.

Subgroup definition. The *missing-edge subgroup* is defined as the set of queries for which standard memory-only retrieval fails to recover all gold evidence documents at Recall@10, while memory-only retrieval with injected oracle triplets recovers all gold evidence documents at Recall@10. This identifies queries where the standard relational-memory index lacks the relations needed for complete evidence recovery, but where those failures can be removed by adding answer-aware oracle triplets.

Statistic	Value
σ_{doc}, mean ± std	0.180 ± 0.027
σ_{mem}, mean ± std	0.165 ± 0.026
$\sigma_{\text{doc}}/\sigma_{\text{mem}}$, median / p10 / p90 / max	$1.09/0.88/1.36/2.01$
Queries with $\sigma_{\text{doc}} > \sigma_{\text{mem}}$	69.0%
Queries with $\sigma_{\text{doc}} \geq 2\sigma_{\text{mem}}$	0.1%
Overlap Pearson, median	0.57
Additive vs. Z-Agreement Spearman, median	0.82

Table 7: MuSiQue score diagnostics for document-memory fusion. Z-Agreement corrects query-dependent score-scale differences while preserving a moderate cross-view agreement signal.

Limitations. This experiment measures an upper bound on performance under complete relational knowledge and is used to showcase the missing edge problem. The oracle triplets are generated with access to gold documents and the answer, making this setting not deployable at test time.

C Z-Agreement Calibration Diagnostics

We further inspect whether Z-Agreement only rescales one dominant view or captures a useful cross-view signal. Table 7 reports per-query diagnostics on 1,000 MuSiQue queries. The score-scale mismatch is modest on average but query-dependent: the median $\sigma_{\text{doc}}/\sigma_{\text{mem}}$ ratio is 1.09, while the range reaches 2.01 and document scores have larger variance in 69.0% of queries. At the same time, documents retrieved by both views have moderate score agreement (median Pearson 0.57), and Z-Agreement changes the additive-fusion ranking non-trivially (median Spearman 0.82).

Oracle Triplet Generation Prompt

```
You are an expert knowledge graph annotator specializing in
decomposition-free multi-hop reasoning chains.

### Task
Given a multi-hop question, supporting evidence documents, and the final
answer, create a chain of verbalized knowledge graph triplets that show
the complete reasoning path from question to answer.

### Important Guidelines
1. Verbalized Format: Write triplets as natural sentences like
"Paris is the capital of France", NOT as tuples like
"(Paris, capital_of, France)".
2. Verbalized Format Examples:
- ✓ "Einstein developed the theory of relativity"
- ✓ "The Eiffel Tower is located in Paris"
- × "(Einstein, developed, theory of relativity)" — NO tuples!
3. Hop Numbering: Mark each triplet with explicit tags:
<triplet_1>...</triplet_1>, <triplet_2>...</triplet_2>, etc.

### Requirements
1. Minimum Triplets: Generate at least {num_documents} triplets
(one per evidence document minimum)
2. Coherence: Ensure NO reasoning gaps between triplets
3. NO SHORTCUTS: Show the complete reasoning chain
4. Answer Target: The final triplet(s) must logically lead to: "{answer}"

### Scratchpad Workflow
Part 1: Plan the reasoning chain; iterate until gap-free
Part 2: Generate positive chain with numbered triplets

### Output Format
[SCRATCHPAD] ... [/SCRATCHPAD]
[POSITIVE_CHAIN]
<triplet_1>{first verbalized triplet}</triplet_1>
<triplet_N>{final verbalized triplet}</triplet_N>
[/POSITIVE_CHAIN]


### Input
[QUESTION] {question} [/QUESTION]
[EVIDENCE_DOCUMENTS] {evidence_docs} [/EVIDENCE_DOCUMENTS]
[ANSWER] {answer} [/ANSWER]
[NUM_DOCUMENTS] {num_documents} [/NUM_DOCUMENTS]
```

Figure 2: Full prompt template used to generate oracle triplets for the missing-edge analysis. Placeholders in curly braces ({question}, {evidence_docs}, {answer}, {num_documents}) are filled per query.

Encoder	Method	R@5	R@10
Qwen3-Embedding-8B	Memory-only (b=5, H=4)	69.79	80.49
Qwen3-Embedding-8B	MMR-Tree + Z-Agreement	73.99	83.87
GritHopper	Memory-only Full Expansion	76.15	84.40
GritHopper	MMR-Tree + Z-Agreement	82.12	90.02

Table 8: MuSiQue encoder substitution. Replacing GritHopper with Qwen3-Embedding-8B preserves the relative benefit of dual-tree Z-Agreement over memory-only retrieval, but the contextual GritHopper encoder remains stronger.

D Encoder Substitution on MuSiQue

To separate the retrieval formulation from the contextual encoder, we also substitute Qwen3-Embedding-8B for GritHopper in the MuSiQue multi-hop retriever. Table 8 shows that the same memory-only and dual-tree retrieval formulations remain effective with Qwen3, but underperform the corresponding GritHopper variants from Table 1. This indicates that the gains come from both the relational-memory retrieval formulation and the contextual multi-hop encoder.

E Agreement Weight Ablation

We ablate the agreement weight γ in the cross-view scoring formula:

$$s_{\text{final}}(d) = (s_D(d) + s_M(d)) \cdot (1 + \gamma) \quad (1)$$

for documents d found by both views. Negative γ penalizes agreement (reducing the combined score), while positive γ boosts it. Table 9 shows results across a range of values.

Key findings: (1) Performance is stable for $\gamma \in [-0.3, 0.9]$, with R@5 varying by only 0.08% on MuSiQue. (2) Slight penalties ($\gamma = -0.3$) marginally improve R@5 and R@10. (3) Extreme penalties ($\gamma < -0.5$) severely degrade performance, confirming that the cross-view agreement signal is valuable. These results justify using $\gamma = 0$ (pure additive combination) as the default, as it achieves near-optimal performance without hyper-parameter tuning.

F Graph Construction and Inference Cost

F.1 Inference-Time Latency: Head-to-Head

We compare end-to-end retrieval latency on FanOutQA dev (310 queries) for the strongest LatentGraphRAG configuration (MMR-Tree + Z-Agreement) against a fully reproduction of HippoRAG 2. Both runs re-encode every query and reranker prompt at inference time. Document and triplet embeddings, the knowledge graph, and the model weights are treated as offline artifacts and stay loaded between queries in both pipelines.

Hardware footprint. The two methods have different GPU requirements at inference time. LatentGraphRAG runs on three A100 80GB: one for the GritHopper-7B encoder, one for the document fp16 index, and one for the memory fp16 index, with the two index searches overlapping on parallel CUDA streams. HippoRAG 2 requires four A100 80GB to host Qwen3-Next-80B-A3B-Instruct via vLLM with tensor parallelism TP $= 4$, plus the Qwen3-Embedding-8B query encoder co-located with the vLLM workers. HippoRAG 2 makes between five and ten LLM calls per query (entity recognition, DSPy rerank, query rewriting); our method makes zero.

Discussion. Table 10 shows that the two methods land within 17% of one another on per-query wall-clock at dataset throughput, despite substantially different inference paths. HippoRAG 2 amortizes well across queries because the underlying MoE model (Qwen3-Next-80B-A3B activates only ∼3B parameters per token) generates reranker outputs in a few seconds per call. LatentGraphRAG has no inference-time LLM at all but spends `(encoder calls/query)` × `(per-call latency)` = 196 × 50 ms on iterative GritHopper-7B forward passes (Table 11), which dominates 96% of its runtime. Because the encoder is the wall and the GPU is saturated at our chunk size, batching across queries (full-dataset mode) gives no measurable per-query speed-up over the single-sample regime: the single-sample p50 of 10.2 s essentially equals the dataset-amortized 10.5 s.

The Pareto curve for LatentGraphRAG follows directly from the encoder-call budget of each configuration. The headline MMR-Tree + Z-Agreement configuration prioritizes recall and uses 196 encoder calls per query, whereas cheaper variants such as document-only BeamDR (B=5) require only 16 calls. At the measured ∼50 ms per encoder call, this corresponds to roughly 0.8 s per query, about 11× faster than the headline configuration and HippoRAG 2, while also avoiding knowledge-graph construction. Method choice therefore becomes an explicit recall-latency trade-off rather than an artifact of implementation.

	MuSiQue				
γ	**R@1**	**R@5**	**R@10**	**R@50**	**R@100**
-0.9	0.53	0.82	1.27	2.32	6.22
-0.7	3.63	16.82	28.65	69.34	85.26
-0.5	35.09	77.91	85.83	95.86	97.79
-0.3	**35.99**	**82.21**	**90.25**	97.66	98.95
-0.1	35.99	82.13	90.07	97.52	98.95
0.0	35.99	82.13	90.02	97.37	98.92
0.1	35.99	82.13	90.02	97.14	98.90
0.3	35.99	82.13	90.02	97.02	98.87
0.5	35.99	82.13	90.02	97.02	98.83
0.7	35.99	82.13	90.02	96.91	98.78
0.9	35.99	82.13	90.02	96.91	98.78

Table 9: Agreement weight (γ) ablation on MuSiQue. The final score for agreement documents is $(s_D + s_M) \cdot (1 + \gamma)$. Negative values penalize agreement; positive values boost it. Results show that γ has minimal effect on R@5 and R@10 for $\gamma \geq -0.3$, confirming that the additive score combination already captures the agreement signal. Extreme penalties ($\gamma < -0.5$) severely degrade performance, demonstrating that the cross-modal agreement signal is valuable.

Method	**GPUs**	**Inference Engine**	**Total wall (310 queries)**	**ms / query amortized**
HippoRAG 2 (cache-cold)	4 × A100 80GB	vLLM (TP=4) + Qwen3-Emb-8B	2784.1 s	8 981
LatentGraphRAG MMR-Tree + Z-Agreement	3 × A100 80GB	GritHopper-7B + dual fp16 index	3262.4 s	10 524

Table 10: End-to-end retrieval wall-clock on FanOutQA dev (310 queries), A100 80GB, bf16/fp16 throughout. Both runs are cache-cold: HippoRAG 2's SQLite LLM cache is deleted before QA, vLLM is launched with `--no-enable-prefix-caching`, and DSPy is pointed at a fresh empty cache directory. LatentGraphRAG has no inference-time LLM cache to clear. LatentGraphRAG is within 17% of HippoRAG 2 in per-query latency while using one fewer GPU and no inference-time LLM calls.

G Use of AI Assistance

The initial manuscript was written by the authors. We used Grammarly and ChatGPT for language editing, including grammar, phrasing, shortening and clarity improvements. During implementation, we used GitHub Copilot and Claude Code for code suggestions, refactoring, and documentation support. After author-led prototyping and experimentation, Claude Code helped us organize the experimental setup into a production-ready open-source package with documentation and a project page illustrating the method. The research ideas, experimental design, results, analyses, and final claims were developed, checked, and validated by the authors.

Component	ms / query	% of total
HippoRAG 2 (cache-cold, 8 981 ms/query)		
Recognition memory (LLM NER + entity matching)	6 820	76.0
Personalized PageRank traversal	1 553	17.3
Miscellaneous (DSPy rerank, fact filtering, I/O)	608	6.7
LatentGraphRAG MMR-Tree + Z-Agreement (10 197 ms/query, single-sample)		
GritHopper-7B encoder forward passes (196 calls)	9 770	96.0
Per-query Python selection (mask, MMR, edge construction)	358	3.5
Dual fp16 search (cuda:1 ‖ cuda:2 streams)	68	0.7
Z-Agreement fusion (per-query z-norm + sort)	2	0.02

Table 11: Per-component latency breakdown. HippoRAG 2 spends most of its time in LLM-driven entity matching; our method spends almost all of its time in GritHopper-7B forward passes. Both are essentially bound by a single learned component, but in different parts of the pipeline.