HYPERRECO: RETRIEVING AND CONNECTING EVIDENCE WITH HYPERGRAPH NEURAL NETWORKS FOR LLM MULTI-HOP REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown strong capabilities, with retrieval-augmented generation (RAG) supporting complex multi-hop reasoning by retrieving evidence distributed across documents. Graph-based approaches exploit connections among evidence, and hypergraph-based retrieval further preserves higher-order entity associations within documents and connects documents through shared entities. However, existing hypergraph retrievers often rely on predefined structural expansion or diffusion, which may miss query-dependent interactions needed to identify relevant evidence. They also leave connections among retrieved evidence implicit, requiring LLMs to reconstruct these connections before reasoning. Therefore, we propose `HyperReCo`, a framework for retrieving and connecting evidence with a hypergraph neural network (HyperGNN). We represent each document as a hyperedge over its extracted entities, with shared entities connecting the hyperedges. Through hypergraph message passing with joint supervision over documents and entities, the HyperGNN learns query-dependent interactions to retrieve complementary evidence. We further introduce Gradient-Guided Hyper-Path Decoding (GGHD), which uses gradient attribution to interpret the learned interactions and translate them into explicit hyper-paths that help LLMs combine complementary facts for multi-hop reasoning. Experiments on six benchmarks show that `HyperReCo` achieves the best retrieval performance among the compared methods on all three multi-hop QA datasets, together with strong downstream QA performance. Case studies and further analyses demonstrate the utility of decoded hyper-paths for connecting retrieved evidence.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities (Achiam et al., 2023). However, solving complex tasks often requires multi-hop reasoning over evidence from multiple sources (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022). This motivates the need for methods that can effectively *retrieve* and *connect* evidence across different documents to support LLM reasoning (Trivedi et al., 2023). Retrieval-augmented generation (RAG) has been proposed to address this challenge by retrieving relevant documents that provide evidence for LLM reasoning (Lewis et al., 2020). Recently, GraphRAG (Edge et al., 2024) has been introduced to extend RAG by organizing knowledge into graph structures and capturing the evidence connections via graph search, relevance propagation, or learned message passing for improved reasoning (Jimenez Gutierrez et al., 2024; Luo et al., 2025b; Yan et al., 2026).

Despite the advances, GraphRAG methods are often criticized for the cost overhead of extracting relational triples to construct knowledge graph (KG) indexes (Edge et al., 2024). More importantly, KG triples only encode binary relations, making them insufficient for modeling the n-ary relations among multiple entities that are widespread in real-world domain knowledge (Wen et al., 2016; Huang et al., 2025; 2026). Recently, hypergraph-based retrieval methods have been proposed to address this limitation by representing evidence as hyperedges over sets of entities (Luo et al., 2025a; Wang et al., 2026; Feng et al., 2026). Hypergraphs can capture higher-order connections across evidence through sharing entities without explicit relation extraction, which makes them a promising alternative for efficient evidence retrieval.

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.

However, existing hypergraph retrievers often rely on predefined structural expansion or graph diffusion mechanisms (Song et al., 2026; Feng et al., 2026). Although hyperedges can encode complex n-ary relations among multiple entities, retrieving useful evidence requires identifying how these entities jointly relate to the query. Structural expansion or graph diffusion alone may not adequately capture such query-dependent interactions, making it difficult to distinguish relevant evidence from connected but irrelevant content (Luo et al., 2025a; Wang et al., 2026). Consequently, these retrievers may miss complementary evidence needed for multi-hop reasoning.

Moreover, existing hypergraph retrievers often leave the connection among retrieved evidence implicit (Wang et al., 2026; Feng et al., 2026). Exploiting hypergraph connectivity during retrieval does not by itself provide LLMs with an explicit representation of which documents contain complementary facts and how they are linked. Therefore, LLMs must reconstruct these connections before reasoning. For example, Figure 1 shows the question: "In what year did the Space Shuttle on which Bill Nelson flew as a payload specialist first launch?" Answering this question requires linking the document about *Bill Nelson*'s shuttle mission to the document about *Space Shuttle Columbia*'s first launch through the shared entity *Columbia*. Explicitly representing this connection can help LLMs distinguish the shuttle's first launch from Nelson's later mission and facilitate reasoning. Addressing these challenges requires *capturing query-dependent interactions to retrieve relevant evidence* and *explicitly representing connections among the retrieved evidence* to guide LLM reasoning.

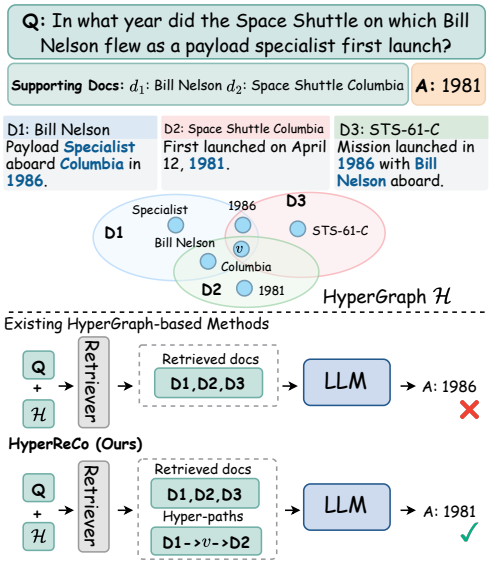

Figure 1: Illustration of the evidence routing gap. HyperReCo augments the retrieved evidence with a decoded hyper-path linking retrieved documents.

To address these challenges, we propose HyperReCo, a framework for retrieving and connecting evidence with a hypergraph neural network (HyperGNN) to enhance LLM multi-hop reasoning. We first extract entities from the documents and represent each document as a hyperedge connecting its entity nodes. These hyperedges are linked through shared entities to form a hypergraph. On this hypergraph, we propose a **HyperGNN** to capture higher-order interactions through hypergraph message passing, which aggregates information within hyperedges and propagates it across hyperedges through shared entities. Through joint supervision of document and entity relevance, the model learns to capture query-dependent interactions that guide the retrieval of complementary evidence. We then introduce **Gradient-Guided Hyper-Path Decoding (GGHD)** to interpret the query-dependent interactions learned by the HyperGNN and translate them into explicit hyper-paths connecting the retrieved evidence. Building on gradient-based graph interpretation (Zhu et al., 2021; Luo et al., 2025b), GGHD performs gradient attribution on the trained HyperGNN to identify how messages passed through shared entities influence its document relevance predictions. It uses these attributions to derive connection costs and combines them with document-dependent connectivity penalties in a query-conditioned prize-collecting Steiner forest (Q-PCSF) objective (Ahmadi et al., 2025). The resulting forest is serialized into grouped hyper-paths that make the connections through shared entities explicit, helping the LLM connect complementary facts across documents for multi-hop reasoning.

In experiments, HyperReCo achieves state-of-the-art performance across three widely-used QA datasets, which demonstrates the effectiveness and efficiency in multi-hop reasoning. Meanwhile, HyperReCo also generalizes well to domain-specific QA datasets spanning medical, novel, and computer science domains without domain-specific training. Ablation studies assess the contributions of the core components, while quantitative and qualitative analysis show that the decoded hyper-paths provide explicit evidence connections that improve downstream question answering.

2 Related Work

Graph Retrieval-Augmented Generation. GraphRAG augments semantic retrieval with structured knowledge to support evidence retrieval and reasoning. GraphRAG (Edge et al., 2024) combines entity graphs with community summaries, LightRAG (Guo et al., 2025) integrates graph indexing with dual-level retrieval, and HippoRAG (Jimenez Gutierrez et al., 2024) propagates query relevance over extracted knowledge graphs using personalized PageRank. Learning-based approaches further model query-dependent structural relevance. GFM-RAG (Luo et al., 2025b) learns transferable message passing over knowledge graphs, and Quest-GNN (Yan et al., 2026) performs question-guided propagation over multi-level graphs. `HyperReCo` follows this learning-based direction but operates over evidence-bearing hyperedges, avoiding explicit relation extraction while learning higher-order structural relevance.

Hypergraph Reasoning. Hypergraphs naturally represent higher-order associations among multiple entities and have recently been incorporated into retrieval-augmented generation. HyperGraphRAG (Luo et al., 2025a) represents n-ary facts as hyperedges, while Hyper-RAG (Feng et al., 2026) combines semantic retrieval with expansion over higher-order correlations. HGRAG (Wang et al., 2026) grounds hyperedges in passages and performs hypergraph diffusion, and EHRAG (Song et al., 2026) further introduces semantic hyperedges constructed from entity clusters. Hyper-RAG (Lien et al., 2026) instead learns a plausibility scorer for query-conditioned traversal. These methods demonstrate the value of higher-order structure, but differ in how query relevance is propagated or learned. In contrast, `HyperReCo` trains a query-conditioned HyperGNN with joint document and entity supervision, enabling structural relevance to be learned directly over overlapping evidence hyperedges.

Path Interpretability. Paths provide an explicit view of how structural information contributes to graph predictions. NBFNet (Zhu et al., 2021) uses gradient-based attribution to identify influential reasoning paths, and GFM-RAG (Luo et al., 2025b) similarly analyzes learned graph predictions through gradients. Other approaches directly use paths as retrieval or generation context: PathRAG (Chen et al., 2026) retrieves relational paths for LLM prompting, while G-retriever (He et al., 2024) formulates subgraph selection as prize-collecting Steiner tree optimization. Our GGHD connects these two directions: it uses message sensitivities from the trained HyperGNN to identify query-relevant entity-mediated connections and organizes them with a query-conditioned forest objective. The resulting grouped hyper-paths translate learned structural relevance into explicit evidence connections for downstream LLM reasoning.

3 Preliminary

Multi-hop Reasoning. Multi-hop reasoning over documents aims to answer a question by combining complementary facts distributed across multiple documents and using the connections among them. Formally, given a question q and a corpus $\mathcal{D} = \{d_1, \ldots, d_N\}$, we seek to design a retriever f_{ret} with parameters θ that selects relevant documents $\mathcal{D}_q = f_{\text{ret}}(q, \mathcal{D}, \theta)$ and captures their connections $d_1 \xrightarrow{r_1} d_2 \xrightarrow{r_2} \cdots \xrightarrow{r_{h-1}} d_h$ that reveal how their facts jointly support an answer a.

Hypergraph. We represent the corpus as a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, with entities as nodes and documents as hyperedges. Following prior work (Wang et al., 2026), we use LLM-based named entity recognition (NER) to extract entity mentions, and let $\mathcal{V}$ denote the resulting set of canonical entities. Each document $d \in \mathcal{D}$ induces a hyperedge as

$$e_d = \{v \in \mathcal{V} \mid v \text{ has a mention in } d\}. \tag{1}$$

The hyperedge family $\mathcal{E} = (e_d)_{d \in \mathcal{D}}$ is indexed by documents, retaining a separate hyperedge for each document even when their entity sets coincide. An entity mentioned in multiple documents belongs to each corresponding hyperedge and connects them. This representation preserves within-document entity associations and cross-document connections without extracting relational triples.

Hyper-Path. For distinct documents $d_0, \ldots, d_h$ with $h \geq 1$, we define a *hyper-path* as a sequence of their hyperedges connected through nonempty shared entity sets:

$$\mathcal{R} = e_{d_0} \xrightarrow{\widehat{\mathcal{B}}_{0,1}} e_{d_1} \xrightarrow{\widehat{\mathcal{B}}_{1,2}} \cdots \xrightarrow{\widehat{\mathcal{B}}_{h-1,h}} e_{d_h}, \tag{2}$$

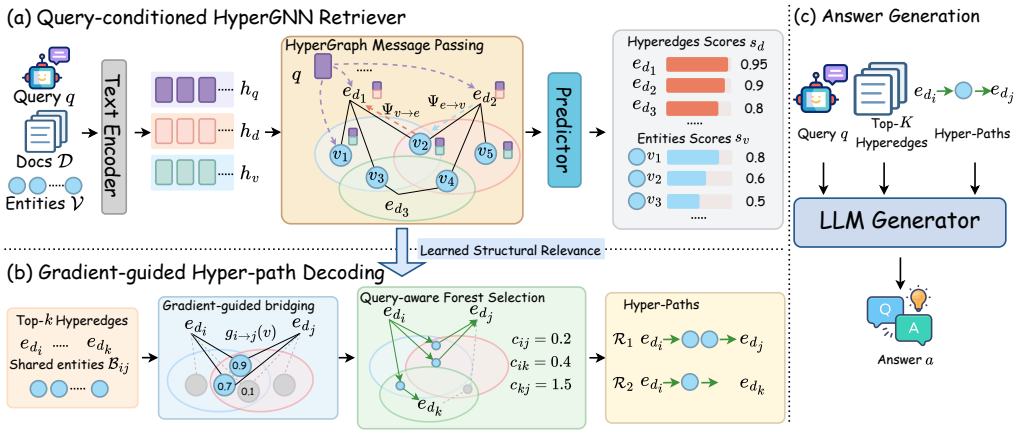

Figure 2: Overview of HyperReCo. (a) The query-conditioned HyperGNN learns structural relevance over hypergraph for evidence retrieval. (b) Gradient-Guided Hyper-Path Decoding (GGHD) uses gradient attribution and Q-PCSF to translate the learned structural relevance into explicit grouped hyper-paths. (c) The retrieved documents and decoded hyper-paths are provided to the LLM for answer generation.

where $\varnothing \neq \widehat{\mathcal{B}}_{t-1,t} \subseteq \mathcal{B}_{t-1,t} = e_{d_{t-1}} \cap e_{d_t}$ for $t = 1, \ldots, h$. Here, $\mathcal{B}_{t-1,t}$ contains all entities shared by the two documents, while $\widehat{\mathcal{B}}_{t-1,t}$ contains those used to connect them along the path. The path spans h document-to-document transitions and makes their entity-mediated connections explicit, providing structural guidance for reasoning over the source documents.

4 Approach

Building on the hypergraph constructed in Section 3, HyperReCo combines two complementary modules to *retrieve relevant evidence* and *make its connections explicit* for LLM reasoning. For evidence retrieval, we propose a **HyperGNN** retriever (Figure 2(a)) to capture query-dependent interactions and retrieve complementary evidence. To connect the retrieved evidence, we introduce **Gradient-Guided Hyper-Path Decoding (GGHD)** (Figure 2(b)), which uses gradient attribution to identify the query-dependent relevance captured by the HyperGNN and translate it into explicit hyper-paths connecting the retrieved documents. These paths guide the LLM in combining complementary facts across documents.

4.1 HyperGraph Neural Network Retriever

Although hypergraphs model complex higher-order connections, their overlapping entity associations introduce enormous candidate evidence links, making it difficult to identify query-relevant interactions from structure alone (Song et al., 2026; Wang et al., 2026). Graph neural networks can learn expressive structural representations on hypergraphs (Huang et al., 2025; 2026) and have also been applied to evidence retrieval (Luo et al., 2025b). We therefore propose a HyperGNN retriever that learns query-dependent interactions through alternating hypergraph message passing to retrieve complementary evidence.

Query-dependent Representation Initialization. We first adopt a frozen text encoder to encode the query, documents, and extracted entities into embeddings x_q, x_d, and x_v, respectively. We then project x_q to obtain the query representation h_q, and inject this query signal into the document and entity representations to initialize the corresponding hyperedge and entity states, denoted by $h_{e_d}^{(0)}$ and $h_v^{(0)}$. Detailed initialization is provided in Appendix B.1.1.

Hypergraph Message Passing. At each layer ℓ, the HyperGNN alternates between hyperedge-to-entity and entity-to-hyperedge propagation. We first aggregate information from the hyperedges containing entity v to update its representation as:

$$h_v^{(\ell+1)} = \Phi_V^{(\ell)}\left(h_v^{(\ell)}, \mathrm{AGG}_{e_d:\, v\in e_d}\, \Psi_{E\to V}^{(\ell)}\left(h_{e_d}^{(\ell)}, h_v^{(\ell)}, h_q\right)\right), \tag{3}$$

where $\Psi^{(\ell)}_{E\to V}$ computes query-conditioned messages from hyperedges to entities, AGG performs attention-based aggregation, and $\Phi^{(\ell)}_{V}$ updates the entity state. The updated entity representations are then propagated back to their associated hyperedges:

$$h^{(\ell+1)}_{e_d} = \Phi^{(\ell)}_{E}\left(h^{(\ell)}_{e_d}, \mathrm{AGG}_{v\in e_d}\, \Psi^{(\ell)}_{V\to E}\big(h^{(\ell+1)}_{v}, h^{(\ell)}_{e_d}, h_q\big)\right), \tag{4}$$

where $\Psi^{(\ell)}_{V\to E}$ and $\Phi^{(\ell)}_{E}$ denote the corresponding message and update functions. By repeatedly exchanging information between entities and hyperedges, the HyperGNN learns how shared entities contribute differently across evidence units under the current query, thereby capturing query-conditioned structural relevance. Detailed message functions and attention mechanisms are provided in Appendix B.1.2.

Relevance Scoring. After L propagation layers, we predict relevance for both hyperedges and entities:

$$s_d = f_{\mathrm{doc}}\big([h^{(L)}_{e_d}; h_q]\big), s_v = f_{\mathrm{ent}}\big([h^{(L)}_{v}; h_q]\big), \tag{5}$$

where s_d denotes the relevance logit associated with hyperedge e_d, and s_v denotes the relevance logit of entity v. Joint supervision at the hyperedge and entity levels encourages the HyperGNN to learn structural relevance from both evidence and its constituent entities. We rank the hyperedges by s_d and return the source documents corresponding to the top-K hyperedges as $\mathcal{D}_q$.

4.2 Gradient-Guided Hyper-Path Decoding

The predicted relevance scores identify important evidence units, but do not explicitly reveal how the retrieved evidence should be connected. We therefore introduce **Gradient-Guided Hyper-Path Decoding (GGHD)** to expose the structural relevance learned by the HyperGNN as explicit evidence connections. Inspired by gradient-based path interpretation (Zhu et al., 2021; Luo et al., 2025b), GGHD attributes document predictions to entity-mediated messages and organizes the resulting connections with a query-conditioned prize-collecting Steiner forest (Q-PCSF).

Gradient-guided connections. Let $\mathcal{D}_q$ denote the set of documents retrieved for query q, and let e_d be the document-induced hyperedge associated with document d. For two distinct retrieved documents $d_i, d_j \in \mathcal{D}_q$, their shared-entity set is $\mathcal{B}_{ij} = e_{d_i} \cap e_{d_j}$. To measure how a shared entity $v \in \mathcal{B}_{ij}$ contributes to the relevance prediction of d_j, we place independent scalar gates $\gamma^{(\ell)}_{e_{d_i}\to v}$ and $\gamma^{(\ell)}_{v\to e_{d_j}}$ on the attention-normalized hyperedge-to-entity and entity-to-hyperedge messages at propagation layer ℓ, respectively. The model parameters are frozen during attribution, and all gates are evaluated at one so that the original forward prediction is unchanged. We define the directional bridge attribution as

$$g^{(\ell)}_{i\to j}(v) = \frac{1}{2}\left[\frac{\partial s_{d_j}}{\partial \gamma^{(\ell)}_{e_{d_i}\to v}} + \frac{\partial s_{d_j}}{\partial \gamma^{(\ell)}_{v\to e_{d_j}}}\right]_{\boldsymbol{\gamma}=1}, \tag{6}$$

where s_{d_j} is the relevance logit predicted by the HyperGNN for target document d_j. Therefore, $g^{(\ell)}_{i\to j}(v)$ measures the local first-order sensitivity of the target prediction to the two messages associated with the entity-mediated connection $e_{d_i} \to v \to e_{d_j}$. Since two documents may share many incidental entities, for each direction and propagation layer we retain only the Top-M shared entities with positive attribution, denoted by $\widehat{\mathcal{B}}^{(\ell)}_{i\to j} \subseteq \mathcal{B}_{ij}$.

Gradient-derived connection costs. We next convert the directional bridge attributions into an undirected connection score between each pair of retrieved documents. Let

$$w_j = \frac{\exp(s_{d_j})}{\sum_{d\in\mathcal{D}_q}\exp(s_d)}, \qquad b^{(\ell)}_{i\to j} = w_j \max_{v\in\widehat{\mathcal{B}}^{(\ell)}_{i\to j}} g^{(\ell)}_{i\to j}(v), \tag{7}$$

where w_j is the normalized relevance of target document d_j within the retrieved set and $b^{(\ell)}_{i\to j}$ measures the relevance-weighted strength of the corresponding directional connection at layer ℓ. Because the same document pair may interact in either direction and at multiple propagation layers, we aggregate them by

$$b_{ij} = \max_{\ell}\left\{b^{(\ell)}_{i\to j}, b^{(\ell)}_{j\to i}\right\}, \qquad c_{ij} = \log\left(1 + \frac{\mu_q}{b_{ij}}\right), \tag{8}$$

where b_{ij} is the final connection strength and μ_q is the median positive connection strength among candidate document pairs for query q. The resulting c_{ij} serves as the connection cost: document pairs whose relevance predictions are more strongly supported by shared-entity interactions receive lower costs. We retain only pairs with $b_{ij} > 0$, yielding the candidate connection set E_q over the retrieved documents.

Query-conditioned forest decoding. Gradient-derived costs characterize individual connections, whereas evidence routing requires selecting connections jointly to organize the retrieved evidence. Inspired by prize-collecting graph retrieval (He et al., 2024), we use a query-conditioned prize-collecting Steiner forest (Q-PCSF) objective to balance connection costs against relevance-dependent connectivity demands. Let $G_q = (\mathcal{D}_q, E_q)$ denote the candidate graph, and select the highest-scoring retrieved document $d_a \in \arg\max_{d_i \in \mathcal{D}_q} s_{d_i}$ as the anchor. For every other retrieved document d_j, we introduce an anchor-to-document connectivity demand with penalty as

$$\pi_{aj} = \lambda \sqrt{p_{d_a} p_{d_j}}, \qquad p_d = \sigma(s_d), \tag{9}$$

where p_d is the predicted relevance probability and $\lambda > 0$ controls the trade-off between selecting additional connections and leaving relevant documents disconnected. Higher-relevance documents therefore incur larger penalties when they are not connected to the anchor.

We select the forest edge set by minimizing

$$F_q^\star \in \arg\min_{F \in \mathfrak{F}_q} \left\{ \sum_{\{d_i, d_j\} \in F} c_{ij} + \sum_{d_j \in \mathcal{D}_q \setminus \{d_a\}} \pi_{aj} 1[d_a \not\leftrightarrow_F d_j] \right\}, \tag{10}$$

where $\mathfrak{F}_q$ is the family of edge sets $F \subseteq E_q$ for which $(\mathcal{D}_q, F)$ forms a forest. The indicator $1[d_a \not\leftrightarrow_F d_j]$ equals one when the anchor and document d_j belong to different connected components. The first term favors strongly attributed connections through their lower costs, while the second penalizes unsatisfied connectivity demands. A demand can be satisfied through intermediate retrieved documents, allowing multiple connections to jointly support an evidence route. The objective also permits demands to remain unsatisfied when the additional connection costs outweigh the penalties avoided. The detailed optimization procedure is provided in Appendix B.2.

4.3 Optimization and Answer Generation

Given the supporting-document set $\mathcal{S}_q$, we supervise hyperedge relevance using document labels and derive weak entity supervision from entities appearing in the supporting documents (Luo et al., 2025b). Together with a KL-based semantic distillation objective, the retriever is optimized as

$$\theta^\star = \arg\min_\theta \mathbb{E}_{q \sim \mathcal{Q}_{\text{train}}} \left[\mathcal{L}_{\text{ret}}(\theta; q) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(\theta; q) \right], \tag{11}$$

where λ_{KL} balances supervised relevance learning and semantic distillation. Detailed definitions of the supervision signals and loss terms are provided in Appendix B.3.

After training, GGHD decodes the learned structural relevance without updating $\theta^\star$. The resulting forest $F_q^\star$ is serialized into hyper-paths $\mathcal{R}_q$ and provided together with the retrieved documents $\mathcal{D}_q$ for answer generation, which can be formulated as

$$a = f_{\text{gen}}(q, \mathcal{D}_q, \mathcal{R}_q). \tag{12}$$

The hyper-paths provide explicit guidance for composing evidence across documents, while factual claims remain grounded in the retrieved content. Serialization and prompt details are provided in Appendix B.5.

5 Experiment

In this section, we attempt to answer the following research questions: **RQ1:** How does `HyperReCo` perform in multi-hop evidence retrieval and question answering compared with existing methods? **RQ2:** Can `HyperReCo` effectively generalize across different domains? **RQ3:** How do the key components of `HyperReCo` contribute to the overall performance? **RQ4:** How do the hyper-paths derived from `HyperReCo` contribute to the downstream task? **RQ5:** How efficient is `HyperReCo` in terms of index construction and inference?

Table 1: QA reasoning performance comparison. GPT-4o-mini is used as the LLM for reasoning. The best result is in bold.

Method	HotpotQA		MuSiQue		2Wiki		G-bench (Novel)	G-bench (Medical)	G-bench (CS)
	EM	F1	EM	F1	EM	F1	ACC	ACC	ACC
Non-structure Methods									
None (GPT-4o-mini) (OpenAI, 2024)	28.6	41.0	11.2	36.3	30.2	36.3	51.4	67.1	70.7
BM25 (Robertson & Walker, 1994)	52.0	63.4	20.3	28.8	47.9	51.2	56.5	68.7	71.7
ColBERTv2 (Santhanam et al., 2022)	43.4	57.7	15.5	26.4	33.4	43.3	56.2	71.8	71.9
Qwen3-Emb (8B) (Zhang et al., 2025)	53.4	67.6	31.9	44.1	57.2	63.2	56.2	70.4	73.5
NV-Embed-v2 (7B) (Lee et al., 2025)	56.9	70.9	31.8	45.5	57.6	64.3	**65.3**	77.7	74.2
Graph-enhanced Methods									
RAPTOR (Sarthi et al., 2024)	50.6	64.7	27.7	39.2	39.7	48.4	43.2	57.1	73.6
GraphRAG (MS) (Edge et al., 2024)	51.4	67.6	27.0	42.0	34.7	61.0	50.9	45.2	72.5
LightRAG (Guo et al., 2025)	9.9	20.2	2.0	9.3	2.5	12.1	45.1	63.9	71.2
HippoRAG (Jimenez Gutierrez et al., 2024)	46.3	60.0	24.0	35.9	59.4	67.3	44.8	59.1	72.6
HippoRAG 2 (Gutiérrez et al., 2025)	56.3	71.1	35.0	49.3	60.5	69.7	56.5	64.9	-
GFM-RAG (Luo et al., 2025b)	56.2	69.5	30.2	49.2	69.8	77.7	58.6	72.2	72.1
G-retriever (He et al., 2024)	41.4	53.4	23.6	34.3	33.5	39.6	-	-	69.8
Quest-GNN (Yan et al., 2026)	54.9	68.3	28.5	40.1	51.5	57.7	-	-	-
EHRAG (Song et al., 2026)	54.5	68.9	34.8	48.5	65.3	73.1	64.8	75.9	73.9
HGRAG (Wang et al., 2026)	55.9	70.5	36.2	48.6	68.2	76.3	64.9	76.1	**74.5**
`HyperReCo`	**60.7**	**74.0**	**39.9**	**51.5**	**72.2**	**78.7**	65.2	**78.4**	73.9

5.1 Experimental Setup

Datasets. We first evaluate the effectiveness of `HyperReCo` on three widely-used multi-hop QA datasets, consisting of HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), following the settings used in (Luo et al., 2025b; Wang et al., 2026) for a fair comparison. To assess the generalization of `HyperReCo` across different domains, we further evaluate `HyperReCo` on three GraphRAG benchmark datasets: G-bench (Novel) (Xiang et al., 2026), G-bench (Medical) (Xiang et al., 2026), and G-bench (CS) (Xiao et al., 2025). Detailed statistics of both the training and evaluation datasets are provided in Appendix A.1.

Baseline methods. We compare `HyperReCo` with two groups of baseline methods: (1) *Non-structure Methods*: BM25 (Robertson & Walker, 1994), ColBERTv2 (Santhanam et al., 2022), Qwen3-Embedding-8B (Qwen3-Emb) (Zhang et al., 2025), and NV-Embed-v2 (7B) (Lee et al., 2025); (2) *Graph-enhanced methods*: RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2025), HippoRAG (Jimenez Gutierrez et al., 2024), HippoRAG 2 (Gutiérrez et al., 2025), GFM-RAG (Luo et al., 2025b), G-retriever (He et al., 2024), Quest-GNN (Yan et al., 2026), EHRAG (Song et al., 2026), and HGRAG (Wang et al., 2026). For a fair comparison, we re-run EHRAG and HGRAG using the same LLM for graph construction and retrain Quest-GNN on the same training datasets used in our experiments. Detailed descriptions of these baseline methods are provided in Appendix B.6.

Metrics. We evaluate `HyperReCo` from two perspectives: downstream QA performance and retrieval performance. For multi-hop QA, we report Exact Match (EM) and F1 following prior work (Jimenez Gutierrez et al., 2024; Luo et al., 2025b), while for the G-bench datasets, we report Accuracy (ACC) following their original evaluation protocols (Xiang et al., 2026; Xiao et al., 2025). For retrieval performance, we use document recall@2 (R@2) and recall@5 (R@5) on multi-hop QA datasets, and evidence recall (Recall) on the G-bench datasets (Xiang et al., 2026).

Implementation Details. We train `HyperReCo` using large-scale datasets curated from GFM-RAG (Luo et al., 2025b), which consists of 6,000 query samples and 53,573 documents. GPT-4o-mini is consistently used for QA inference across all methods. More training and implementation details are provided in Appendix B.4.

5.2 Main Results (RQ1 & RQ2)

Multi-hop QA and Retrieval. As shown in Tables 1 and 2, `HyperReCo` achieves the strongest performance on all three multi-hop QA benchmarks in both downstream QA and evidence retrieval. In particular, the consistent gains in R@2 and R@5 indicate that the query-conditioned HyperGNN better prioritizes complementary evidence beyond semantic similarity alone. These retrieval improvements are accompanied by higher EM and F1 scores, suggesting that combining learned structural

Table 2: Retrieval performance comparison. Recall@k (R@k) is used for multi-hop QA datasets, and evidence recall (Recall) is used for G-bench (Xiang et al., 2026). The best result is in bold.

Method	HotpotQA		MuSiQue		2Wiki		G-bench (Novel)	G-bench (Medical)
	R@2	R@5	R@2	R@5	R@2	R@5	Recall	Recall
Non-structure Methods								
BM25 (Robertson & Walker, 1994)	55.4	72.2	32.3	41.2	51.8	61.9	82.1	87.9
ColBERTv2 (Santhanam et al., 2022)	64.7	79.3	37.9	49.2	59.2	68.2	82.4	89.5
Qwen3-Emb (8B) (Zhang et al., 2025)	74.1	88.8	46.8	62.1	66.2	74.1	82.6	92.7
NV-Embed-v2 (7B) (Lee et al., 2025)	84.1	94.4	52.7	69.5	69.1	76.5	**92.5**	**96.5**
Graph-enhanced Methods								
RAPTOR (Sarthi et al., 2024)	58.1	71.2	35.7	45.3	46.3	53.8	66.1	84.2
GraphRAG (MS) (Edge et al., 2024)	58.3	76.6	35.4	49.3	61.6	77.3	67.4	56.4
LightRAG (Guo et al., 2025)	38.8	54.7	24.8	34.7	45.1	59.1	79.6	82.6
HippoRAG (Jimenez Gutierrez et al., 2024)	60.1	78.5	41.2	53.2	68.4	87.0	81.2	84.0
HippoRAG 2 (Gutiérrez et al., 2025)	80.5	95.7	53.5	74.2	75.2	90.5	66.2	73.6
G-retriever (He et al., 2024)	51.8	63.6	35.6	43.5	60.9	66.5	-	-
GFM-RAG (Luo et al., 2025b)	75.6	89.6	43.5	57.6	79.1	92.4	75.9	82.2
Quest-GNN (Yan et al., 2026)	76.7	89.1	38.7	54.0	61.1	71.2	-	-
EHRAG (Song et al., 2026)	76.6	92.7	53.0	70.7	69.4	85.6	92.3	93.0
HGRAG (Wang et al., 2026)	77.8	94.0	53.7	73.1	74.9	91.4	92.2	93.8
`HyperReCo`	**87.1**	**96.8**	**57.2**	**74.5**	**81.2**	**97.8**	91.9	95.9

Table 3: Qualitative example of hyper-path-guided reasoning

Question	When did Lord George Scott's father die?	
Gold Answer	5 November 1914	
Supporting Docs.	["Lord George Scott", "William Montagu Douglas Scott, 6th Duke of Buccleuch"]	
Retrieved Docs.	[D1: **"Lord George Scott"**, D2: "George Lyon, 5th Lord Glamis", D3: **"William Montagu Douglas Scott, 6th Duke of Buccleuch"**, D4: "Charlotte Montagu Douglas Scott, Duchess of Buccleuch", D5: "Walter Montagu Douglas Scott, 8th Duke of Buccleuch"]	
Evidence Input	D1, D2, D3, D4, D5	**w/o Hyper-paths**
Answer	The passage does not provide the date of Lord George Scott's father's death.	✗ EM = 0, F1 = 0
Evidence Input	D1, D2, D3, D4, D5	**`HyperReCo`**
Decoded Hyper-paths	Path 1: **D1** → **[William Montagu Douglas Scott; 6th Duke of Buccleuch]** → **D3**;	
Answer	**5 November 1914**	✓ EM = 1, F1 = 1

relevance with explicit hyper-path guidance benefits downstream multi-hop reasoning. Compared with retrievers based on predefined search or propagation, `HyperReCo` learns query-dependent interactions directly over hypergraphs and further exposes the resulting evidence connections to the LLM through GGHD.

Cross-domain Generalization. We further evaluate `HyperReCo` on G-bench datasets covering novel, medical, and computer science domains to examine its generalization ability. As shown in Tables 1 and 2, `HyperReCo` remains competitive with strong semantic and graph-enhanced baseline methods without domain-specific training, and achieves the best QA performance on the medical benchmark. Notably, its QA performance can remain strong even when evidence recall is comparable to or slightly below the strongest retrievers, indicating that downstream reasoning benefits not only from relevant evidence, but also from explicit evidence connections.

5.3 Ablation Study (RQ3)

We conduct ablation studies to examine the contribution of query-conditioned structural relevance learning in the HyperGNN retriever. Specifically, we compare the full model with: (1) *w/o HyperGNN*, which directly uses NV-Embed-v2 for retrieval; (2) *w/o MP* (w/o message passing), which removes structural propagation between hyperedges and entities; and (3) *w/o Q-dep* (w/o query-dependent modeling), which removes query conditioning from the HyperGNN. As shown in Figure 3, removing these components consistently degrades R@5 across the three multi-hop QA datasets, demonstrating that the retrieval gains arise from learning query-conditioned structural interactions rather than semantic relevance alone.

5.4 Analysis of Hyper-Path Effectiveness (RQ4)

Decoding Strategies. To examine whether the benefit of GGHD arises from identifying query-relevant evidence connections rather than merely providing structural paths, we compare the full

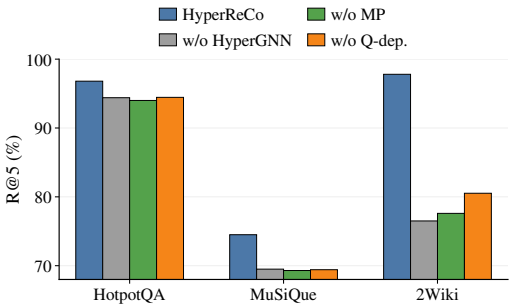

Figure 3: Ablation of structural relevance learning.

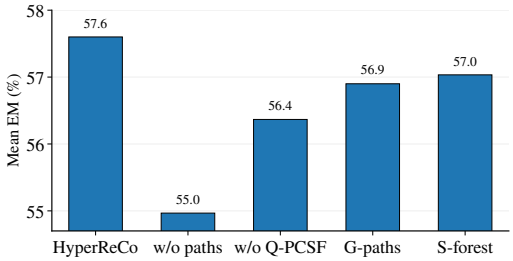

Figure 4: Analysis of hyper-path decoding strategies.

GGHD with four variants: *w/o paths*, which provides only the retrieved documents; *w/o Q-PCSF*, which removes global forest optimization; *G-paths*, a gradient-based path construction strategy; and *S-forest*, a semantic-similarity-based forest construction strategy. As shown in Figure 4, GGHD achieves the highest mean EM, demonstrating the benefit of combining gradient-guided connection scoring with global forest optimization. Detailed definitions of these variants are provided in Appendix C.3 and we further provide an LLM-as-a-judge analysis in Appendix C.4 to assess whether GGHD produces more compact and useful hyper-paths.

Case Study. As shown in Table 3, we further analyze a case where both settings retrieve the same top-5 documents with perfect supporting-document recall, but only `HyperReCo` produces the correct answer. Without hyper-path guidance, the LLM fails to associate the document identifying Lord George Scott's father with the document containing his death date. In contrast, GGHD explicitly exposes the key connection D1 $\rightarrow$ [William Montagu Douglas Scott, 6th Duke of Buccleuch] $\rightarrow$ D3, allowing the LLM to compose the two pieces of evidence and recover the correct answer. This example illustrates the central role of hyper-paths: beyond retrieving relevant documents, they explicitly identify how evidence distributed across documents should be connected for downstream reasoning. Full case details are provided in Appendix C.5.

5.5 Efficiency Analysis (RQ5)

We evaluate the efficiency of `HyperReCo` in terms of offline index construction and online inference. As shown in Table 4, `HyperReCo` requires only 3.3M LLM tokens per 10K documents for index construction, substantially fewer than existing graph-based methods, as it avoids explicit relation extraction and summarization. For online inference, we report the macro-average end-to-end latency over HotpotQA, MuSiQue, and 2Wiki. `HyperReCo` takes 3.55 seconds per query, remaining comparable to GFM-RAG and HGRAG while achieving stronger QA performance. These results demonstrate that `HyperReCo` introduces limited inference overhead while enabling substantially lighter index construction. Detailed per-dataset latency and QA results are provided in Appendix C.2.

Table 4: Efficiency comparison. Token cost is measured per 10K documents. Time and EM are macro-averaged over HotpotQA, MuSiQue, and 2Wiki.

Method	Tokens / 10K ↓	Time (s) ↓	EM ↑
GraphRAG	76M	5.02	37.7
LightRAG	55M	8.39	4.8
GFM-RAG	48M	3.36	52.1
HGRAG	**3.3M**	**3.08**	53.4
`HyperReCo`	**3.3M**	3.55	**57.6**

6 Conclusion

We propose `HyperReCo`, a hypergraph-based framework for retrieving and connecting evidence for LLM multi-hop reasoning. A query-conditioned HyperGNN learns structural relevance over higher-order evidence associations to retrieve complementary evidence, while GGHD translates the learned interactions into explicit hyper-paths that guide cross-document evidence composition. Experiments on six benchmarks demonstrate strong retrieval and QA performance, with ablations and qualitative analyses confirming the effectiveness of both structural relevance learning and hyper-path guidance. Overall, our results show that successful multi-hop reasoning requires not only retrieving relevant evidence, but also exposing how that evidence is connected.

DECLARATION OF LARGE LANGUAGE MODEL USAGE

Generative AI tools were used to assist with proofreading and language polishing, as well as code implementation and debugging. LLMs were also used as experimental components for entity extraction, answer generation, and hyper-path evaluation are described in the corresponding method and experimental sections. All AI-assisted text was carefully reviewed and revised by the authors, and all AI-assisted code was inspected, tested, and verified for correctness before use. The authors take full responsibility for the accuracy, integrity, and final content of this work.

ETHICS STATEMENT

Our research addresses scientific questions only and does not involve human subjects, animals, or environmentally sensitive materials. Therefore, we do not anticipate any significant ethical risks or conflicts of interest. We are committed to maintaining high standards of scientific integrity and research ethics to ensure the validity, reliability, and transparency of our findings.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method and its training and inference procedures in Section 4. Experimental settings, implementation details, hyperparameters, and data processing procedures are provided in Section 5.1 and Appendix B. Experimental settings and baselines have been rigorously verified to ensure fair comparison. We also specify the datasets, backbone models, retrieval settings, and language models used throughout the evaluation. Code and configurations for reproducing the main experiments will be made available.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ali Ahmadi, Iman Gholami, MohammadTaghi Hajiaghayi, Peyman Jabbarzade, and Mohammad Mahdavi. 2-approximation for prize-collecting steiner forest. *Journal of the ACM*, 72(2):1–27, 2025.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30183–30191, 2026.

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

Yifan Feng, Hao Hu, Shihui Ying, Xingliang Hou, Shiquan Liu, Mingyuan Yang, Junchang Li, Shaoyi Du, Nanning Zheng, Han Hu, et al. Hyper-rag: Combating llm hallucinations using hypergraph-driven retrieval-augmented generation. *Nature Communications*, 2026.

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 21497–21515, 2025.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

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

Xingyue Huang, Miguel Romero Orth, Pablo Barceló, Michael M. Bronstein, and İsmail İlkan Ceylan. Link prediction with relational hypergraphs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=S6fe4aH6YA.

Xingyue Huang, Mikhail Galkin, Michael M. Bronstein, and İsmail İlkan Ceylan. HYPER: A foundation model for inductive link prediction with knowledge hypergraphs. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 126914–126947, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/cdf80d651420b47da80f789f2631b1b5-Paper-Conference.pdf.

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

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. In *International Conference on Learning Representations*, volume 2025, pp. 79310–79333, 2025.

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

Wen-Sheng Lien, Yu-Kai Chan, Hao-Lung Hsiao, Bo-Kai Ruan, Meng-Fen Chiang, Chien-An Chen, Yi-Ren Yeh, and Hong-Han Shuai. Hyperrag: Reasoning n-ary facts over hypergraphs for retrieval augmented generation. In *Proceedings of the ACM Web Conference 2026*, pp. 2465–2476, 2026.

Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, et al. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. *Advances in Neural Information Processing Systems*, 38:152206–152234, 2025a.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: graph foundation model for retrieval augmented generation. *Advances in Neural Information Processing Systems*, 38:36371–36405, 2025b.

OpenAI. Hello gpt-4o, 2024. URL https://openai.com/index/hello-gpt-4o/.

Stephen E Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 232–241. Springer, 1994.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 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*, pp. 3715–3734, 2022.

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

Yifan Song, Xingjian Tao, Zhicheng Yang, Yihong Luo, and Jing Tang. Ehrag: Bridging semantic gaps in lightweight graphrag via hybrid hypergraph construction and retrieval. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 24638–24652, 2026.

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

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

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

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 1300–1307. AAAI Press, 2016. ISBN 9781577357704.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 66145–66178, 2026.

Yilin Xiao, Junnan Dong, Chuang Zhou, Su Dong, Qian-wen Zhang, Di Yin, Xing Sun, and Xiao Huang. Graphrag-bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation. *arXiv preprint arXiv:2506.02404*, 2025.

Yuchen Yan, Peiyan Zhang, Zhihua Liu, Hao Wang, Yatao Bian, Weiming Li, and Xiaoshuai Hao. Question-adaptive graph learning for multi-hop retrieval augmented generation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2150–2161, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021.

A DATASETS

A.1 DATASET STATISTICS

Training Datasets. We train `HyperReCo` on 6,000 training questions collected from HotpotQA, MuSiQue, and 2Wiki, following the training corpus used in GFM-RAG (Luo et al., 2025b). The resulting training corpus contains 53,573 documents and 326,286 extracted entities. Detailed statistics are shown in Table 5.

Table 5: Statistics of the training datasets.

Dataset	#Questions	#Documents	#Entities
HotpotQA	2,000	19,527	130,484
MuSiQue	2,000	22,417	126,603
2Wiki	2,000	11,629	69,199
Total	**6,000**	**53,573**	**326,286**

Evaluation Datasets. We evaluate `HyperReCo` on three multi-hop QA benchmarks, including HotpotQA, MuSiQue, and 2Wiki, and three domain-specific GraphRAG benchmarks, including G-bench (Novel), G-bench (Medical), and G-bench (CS). The detailed statistics are summarized in Table 6.

Table 6: Statistics of the evaluation datasets.

Dataset	# Query	# Document
HotpotQA (Yang et al., 2018)	1,000	9,221
MuSiQue (Trivedi et al., 2022)	1,000	11,656
2Wiki (Ho et al., 2020)	1,000	6,119
G-bench (Novel) (Xiang et al., 2026)	2,010	461
G-bench (Medical) (Xiang et al., 2026)	2,062	2,406
G-bench (CS) (Xiao et al., 2025)	1,018	24,534

B METHODS

B.1 HYPERGRAPH NEURAL NETWORK RETRIEVER

B.1.1 QUERY-DEPENDENT REPRESENTATION INITIALIZATION

A frozen text encoder maps the query q, source document d, and entity v into embeddings $x_q, x_d, x_v \in \mathbb{R}^D$, respectively. We project the query embedding into the HyperGNN hidden space as

$$h_q = W_q x_q, \tag{13}$$

where $W_q \in \mathbb{R}^{d_h \times D}$ is a learnable projection matrix.

For each document-induced hyperedge e_d, we first compute the query-document similarity $c_{qd} = \cos(x_q, x_d)$ and select a dense seed set $\mathcal{A}_q$ containing the top-K documents according to c_{qd}. We define the normalized seed relevance as

$$g_d = \begin{cases} \dfrac{\max(c_{qd}, 0)}{\max_{d' \in \mathcal{A}_q} \max(c_{qd'}, 0) + \epsilon}, & d \in \mathcal{A}_q, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where ϵ is a small constant for numerical stability. The initial hyperedge state is then

$$h_{e_d}^{(0)} = W_E x_d + \alpha_{\text{init}} g_d h_q, \tag{15}$$

where W_E projects the source-document embedding and α_{init} is a learnable scalar controlling the strength of query injection. Therefore, semantically relevant seed hyperedges receive stronger query conditioning before structural propagation.

For entities, we additionally incorporate query-linking and corpus-level structural features. Let $f_{v,q} \in \mathbb{R}^F$ denote the feature vector associated with entity v under query q, including query-linking and entity-frequency information. We initialize the entity state as

$$h_v^{(0)} = W_f f_{v,q} + \sigma(c_{qv}) W_V x_v, \tag{16}$$

where $c_{qv} = \cos(x_q, x_v)$ and $\sigma(\cdot)$ denotes the sigmoid function. The query therefore conditions entity initialization through both $f_{v,q}$ and the query-entity similarity gate.

B.1.2 Query-Conditioned Hypergraph Message Passing

We provide the detailed message, aggregation, and update operators used by the HyperGNN. Let q denote the input query, d a source document, e_d the hyperedge induced by d, and v an entity contained in e_d. At propagation layer ℓ, $h_{e_d}^{(\ell)} \in \mathbb{R}^{d_h}$ and $h_v^{(\ell)} \in \mathbb{R}^{d_h}$ denote the hidden states of hyperedge e_d and entity v, respectively.

Query-Conditioned Roles. For each entity-hyperedge membership $v \in e_d$, we construct a query-conditioned role representation that characterizes how entity v participates in hyperedge e_d under query q. Let $x_q, x_d, x_v \in \mathbb{R}^D$ denote the frozen text-encoder representations of the query, source document, and entity name, respectively. We define the query-entity and query-document similarities as

$$c_{qv} = \cos(x_q, x_v), \qquad c_{qd} = \cos(x_q, x_d), \tag{17}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity.

We concatenate the semantic representations, membership features, and query similarities to form

$$u_{v,e_d|q} = [x_v; x_d; x_q; \xi_{v,e_d}; c_{qv}; c_{qd}], \tag{18}$$

where $[\cdot;\cdot]$ denotes vector concatenation and $\xi_{v,e_d} \in \mathbb{R}^6$ contains the auxiliary attributes associated with the membership $v \in e_d$, including the query-link indicator used in our implementation. Thus, $u_{v,e_d|q} \in \mathbb{R}^{3D+8}$ combines entity semantics, document semantics, query semantics, and membership-level information.

The role representation is then computed by

$$z_{v,e_d|q} = \mathrm{LN}\left(W_{r,2}\,\mathrm{Dropout}\left(\mathrm{GELU}(W_{r,1}u_{v,e_d|q} + b_{r,1})\right) + b_{r,2}\right), \tag{19}$$

where $W_{r,1} \in \mathbb{R}^{d_h \times (3D+8)}$ and $b_{r,1} \in \mathbb{R}^{d_h}$ project the concatenated input into the HyperGNN hidden space, while $W_{r,2} \in \mathbb{R}^{d_z \times d_h}$ and $b_{r,2} \in \mathbb{R}^{d_z}$ produce the final role representation $z_{v,e_d|q} \in \mathbb{R}^{d_z}$. $\mathrm{GELU}(\cdot)$ denotes the GELU activation, and $\mathrm{LN}(\cdot)$ denotes layer normalization. All projection matrices and bias terms are learned jointly with the HyperGNN.

Message Passing. At each layer ℓ, the HyperGNN performs hyperedge-to-entity propagation followed by entity-to-hyperedge propagation. For each membership $v \in e_d$, we first construct direction-specific interaction representations:

$$u_{e_d \to v}^{(\ell)} = [h_{e_d}^{(\ell)}; h_v^{(\ell)}; z_{v,e_d|q}], \qquad u_{v \to e_d}^{(\ell)} = [h_v^{(\ell+1)}; h_{e_d}^{(\ell)}; z_{v,e_d|q}]. \tag{20}$$

Here, $u_{e_d \to v}^{(\ell)}$ represents the interaction used to propagate information from hyperedge e_d to entity v, while $u_{v \to e_d}^{(\ell)}$ uses the updated entity state for the reverse propagation. Both representations concatenate the sender state, receiver state, and query-conditioned entity–hyperedge role $z_{v,e_d|q}$.

For each direction $r \in \{EV, VE\}$, where EV and VE denote hyperedge-to-entity and entity-to-hyperedge propagation, respectively, a two-layer MLP produces the message:

$$\Psi_r^{(\ell)}(u) = W_{r,2}^{(\ell)}\,\mathrm{GELU}\left(W_{r,1}^{(\ell)}u + b_{r,1}^{(\ell)}\right) + b_{r,2}^{(\ell)}. \tag{21}$$

A separate linear scoring function

$$\psi_r^{(\ell)}(u) = (a_r^{(\ell)})^\top u + b_r^{(\ell)} \tag{22}$$

produces a scalar attention logit for each directed message. Message and attention parameters are learned independently across propagation directions and layers.

For hyperedge-to-entity propagation, attention is normalized over all hyperedges containing the receiving entity v:

$$\beta^{(\ell)}_{e_d \to v} = \frac{\exp\left(\psi^{(\ell)}_{EV}(u^{(\ell)}_{e_d \to v})\right)}{\sum_{d' :\, v \in e_{d'}} \exp\left(\psi^{(\ell)}_{EV}(u^{(\ell)}_{e_{d'} \to v})\right)}. \tag{23}$$

The resulting entity message is the attention-weighted aggregation

$$m^{(\ell)}_v = \sum_{d :\, v \in e_d} \beta^{(\ell)}_{e_d \to v} \Psi^{(\ell)}_{EV}(u^{(\ell)}_{e_d \to v}). \tag{24}$$

Therefore, an entity selectively aggregates information from the source hyperedges in which it occurs.

After updating the entity states, information is propagated back to the hyperedges. For entity-to-hyperedge propagation, attention is normalized over all entities contained in the receiving hyperedge e_d:

$$\alpha^{(\ell)}_{v \to e_d} = \frac{\exp\left(\psi^{(\ell)}_{VE}(u^{(\ell)}_{v \to e_d})\right)}{\sum_{v' \in e_d} \exp\left(\psi^{(\ell)}_{VE}(u^{(\ell)}_{v' \to e_d})\right)}. \tag{25}$$

The aggregated hyperedge message is

$$m^{(\ell)}_{e_d} = \sum_{v \in e_d} \alpha^{(\ell)}_{v \to e_d} \Psi^{(\ell)}_{VE}(u^{(\ell)}_{v \to e_d}). \tag{26}$$

This second propagation step allows each hyperedge to integrate query-conditioned information from its constituent entities, including information transferred through entities shared with other hyperedges.

State Updates. For both propagation directions, we update the receiving state using a GRU followed by residual addition, dropout, and layer normalization:

$$\Phi^{(\ell)}_T(h, m) = \mathrm{LN}^{(\ell)}_T\left(h + \mathrm{Dropout}\left(\mathrm{GRU}^{(\ell)}_T(m, h)\right)\right), \qquad T \in \{V, E\}. \tag{27}$$

Here, $T = V$ and $T = E$ denote entity and hyperedge updates, respectively, m is the aggregated incoming message and h is the previous receiver state. Within each layer, we first compute $h^{(\ell+1)}_v = \Phi^{(\ell)}_V(h^{(\ell)}_v, m^{(\ell)}_v)$ for all entities, and then use these updated entity states to compute $h^{(\ell+1)}_{e_d} = \Phi^{(\ell)}_E(h^{(\ell)}_{e_d}, m^{(\ell)}_{e_d})$ for all hyperedges.

B.2 Exact Q-PCSF Solver

We provide the exact optimization procedure used to solve the Q-PCSF objective in Equation (10). Decoding is performed over the $K = 5$ retrieved documents. For each candidate subset $F \subseteq E_q$, we first determine whether $(\mathcal{D}_q, F)$ is acyclic using a disjoint-set union structure. Subsets containing cycles are discarded. For each valid forest, we evaluate

$$\mathcal{J}(F) = \sum_{\{d_i, d_j\} \in F} c_{ij} + \sum_{d_j \in \mathcal{D}_q \setminus \{d_a\}} \pi_{aj} 1[d_a \not\leftrightarrow_F d_j], \tag{28}$$

The Equation (28) is identical to the objective in Equation (10). Connectivity is evaluated over the entire selected forest, so a demand is considered satisfied whenever d_a and d_j are connected either directly or through intermediate retrieved documents. The empty forest is also included as a valid candidate, with objective

$$\mathcal{J}(\varnothing) = \sum_{d_j \in \mathcal{D}_q \setminus \{d_a\}} \pi_{aj}. \tag{29}$$

The exact solver proceeds as follows:

1. Initialize the best solution with the empty forest.

2. Enumerate all nonempty subsets $F \subseteq E_q$ and discard those containing cycles.
3. For each valid forest, compute $\mathcal{J}(F)$ using the selected-edge costs and the penalties of unsatisfied connectivity demands.
4. Return the forest with the minimum objective value. Exact ties are resolved by preferring fewer selected edges and then the lexicographically smaller sequence of document-index pairs.

B.3 Training Objectives

Given a query q, let $\mathcal{S}_q \subseteq \mathcal{D}$ denote its supporting documents. We assign each document-induced hyperedge e_d a binary relevance label $y_d = 1[d \in \mathcal{S}_q]$. Following GFM-RAG (Luo et al., 2025b), we further derive weak entity supervision from the supporting documents:

$$\mathcal{V}_q^+ = \bigcup_{d \in \mathcal{S}_q} e_d, \qquad y_v = 1[v \in \mathcal{V}_q^+]. \tag{30}$$

Let $p_d = \sigma(s_d)$ and $p_v = \sigma(s_v)$ denote the predicted relevance scores of documents and entities, respectively.

Document Relevance. We supervise document relevance using binary cross-entropy:

$$\mathcal{L}_{\text{ret}}^d = -\frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \left[y_d \log p_d + (1 - y_d) \log(1 - p_d) \right]. \tag{31}$$

Following GFM-RAG (Luo et al., 2025b), we additionally employ a ranking loss $\mathcal{L}_{\text{rank}}^d$ to encourage supporting documents to receive higher relevance scores than negative documents.

Entity Relevance. For entity supervision, we combine pointwise and listwise relevance learning:

$$\mathcal{L}_{\text{ret}}^e = \alpha \mathcal{L}_{\text{BCE}}^e + (1 - \alpha) \mathcal{L}_{\text{list}}^e, \tag{32}$$

where $\mathcal{L}_{\text{BCE}}^e$ supervises binary entity relevance and $\mathcal{L}_{\text{list}}^e$ encourages positive entities to receive higher relevance mass over the complete entity set. Following the GFM-RAG training setup (Luo et al., 2025b), we set $\alpha = 0.3$. The complete retrieval objective is

$$\mathcal{L}_{\text{ret}} = \mathcal{L}_{\text{ret}}^d + \lambda_{\text{rank}} \mathcal{L}_{\text{rank}}^d + \lambda_{\text{ent}} \mathcal{L}_{\text{ret}}^e. \tag{33}$$

Semantic Distillation. To prevent structural learning from drifting away from the semantic relevance captured by the frozen text encoder, we further introduce a semantic distillation objective. We use the frozen text encoder as a semantic teacher to regularize document relevance prediction. Specifically, the encoder similarity and HyperGNN logit are converted into Bernoulli relevance scores,

$$p_{\text{enc}}(d \mid q) = \sigma(c_{qd}), \qquad p_{\text{hyp}}(d \mid q) = \sigma(s_d), \tag{34}$$

and we minimize their KL divergence:

$$\mathcal{L}_{\text{KL}} = \sum_{d \in \mathcal{D}} D_{\text{KL}} \left(\text{Bern}(p_{\text{enc}}(d \mid q)) \parallel \text{Bern}(p_{\text{hyp}}(d \mid q)) \right). \tag{35}$$

The final optimization objective is

$$\theta^\star = \arg\min_\theta \mathbb{E}_{q \sim \mathcal{Q}_{\text{train}}} \left[\mathcal{L}_{\text{ret}}(\theta; q) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(\theta; q) \right]. \tag{36}$$

B.4 Implementation Details

Hypergraph Construction. Following HGRAG (Wang et al., 2026), we use the same LLM-based entity extraction prompt to identify entities from each document. The extracted mentions are normalized into canonical entities and used to construct the hypergraph as described in Preliminary. Since we directly follow the entity extraction protocol of HGRAG, we do not introduce additional prompting strategies for hypergraph construction.

Model Settings. We use frozen NV-Embed-v2 representations with an embedding dimension of 4096 for questions, documents, and entities. The HyperGNN uses a hidden dimension of 384, an entity-hyperedge role dimension of 192, and two message-passing layers. We use five dense seed documents and apply dropout with rate 0.1. Both the document and entity scoring heads use two-layer MLPs with hidden dimension 384 and GELU activation.

Training Settings. We train on 6,000 questions from HotpotQA, MuSiQue, and 2Wiki. The model is optimized with AdamW using a learning rate of 3×10^{-4}, weight decay of 10^{-2}, and $(\beta_1, \beta_2) = (0.9, 0.999)$. Training is performed on two NVIDIA RTX 4090 GPUs with an effective batch size of 2. We train for at most 30 epochs with early-stopping patience 5 and random seed 42. The ranking margin is set to 1.0 with at most 128 hard negatives. Following the official training configuration of GFM-RAG (Luo et al., 2025b), we set the weights of the entity BCE and listwise objectives to 0.3 and 0.7, respectively, with an adversarial temperature of 0.2 for negative reweighting. Detailed training objectives are provided in Appendix B.3.

B.5 Detailed Prompts for Inference

For QA inference, we follow the prompting protocols adopted by prior GraphRAG systems such as GFM-RAG (Luo et al., 2025b) and HGRAG (Wang et al., 2026) for the baseline methods. For `HyperReCo`, we additionally provide the decoded hyper-paths alongside the retrieved documents as explicit cross-document navigation cues, while the document contents remain the factual evidence for answer generation. The complete inference prompt used by `HyperReCo` is shown in Figure 5.

B.6 Detailed Descriptions of Baseline Methods

We compare `HyperReCo` with two groups of representative retrieval methods: *non-structure methods* and *graph-enhanced methods*. The detailed descriptions of the baselines are as follows.

Non-structure Methods. These methods retrieve documents mainly based on lexical or semantic relevance without explicitly modeling cross-document structures.

- **BM25** (Robertson & Walker, 1994) is a classical sparse retrieval method that ranks documents according to query-term statistics based on the probabilistic retrieval model.
- **ColBERTv2** (Santhanam et al., 2022) is a dense retriever based on token-level late interaction, together with residual compression and denoised supervision for effective and efficient retrieval.
- **Qwen3-Embedding-8B** (Zhang et al., 2025) is an LLM-based embedding model built on Qwen3 and trained with a multi-stage pipeline for general-purpose semantic retrieval.
- **NV-Embed-v2** (Lee et al., 2025) is a general-purpose LLM-based embedding model that employs latent-attention pooling and contrastive instruction tuning to produce strong representations for dense retrieval.

Graph-enhanced Methods. These methods introduce graph structures to capture dependencies beyond independent document similarity.

- **RAPTOR** (Sarthi et al., 2024) recursively embeds, clusters, and summarizes text chunks to construct a hierarchical tree, enabling retrieval at different levels of abstraction.
- **GraphRAG** (Edge et al., 2024) constructs an entity-based knowledge graph and applies hierarchical community detection to generate community summaries for retrieval and LLM generation.
- **LightRAG** (Guo et al., 2025) incorporates graph structures into text indexing and employs dual-level retrieval to access both low-level entities and high-level semantic information.
- **HippoRAG** (Jimenez Gutierrez et al., 2024) constructs a knowledge graph from the corpus and applies Personalized PageRank to propagate query relevance for associative multi-hop retrieval.
- **HippoRAG 2** (Gutiérrez et al., 2025) extends HippoRAG by more deeply integrating passage nodes and semantic passage signals into graph-based retrieval.

Hyper-path-guided LLM Reasoning Prompt

```
As an advanced multi-hop reading comprehension assistant, analyze
the supplied documents and question step by step.

Suggested Hyper-paths connect evidence documents. Each entity list
contains shared bridge entities for the adjacent document pair. Use
 a Hyper-path only as optional navigation, and verify every fact in
 the evidence documents. A path or entity name is not factual
evidence. If no useful Hyper-path is supplied, reason from the
documents normally.

Identify the intermediate facts required by the question and
combine them. For comparison or multi-branch questions, resolve
each branch separately before comparing or combining the results.

Return a JSON object with exactly two fields. Put the multi-hop
derivation in "reasoning". Put only the concise requested value in
"answer", with no explanation, preamble, or repeated question.

### Question:
<Question>

### Suggested Hyper-paths:
Path 1: D1 --> [entity1; entity2] --> D2 --> [entity3] --> D3
Path 2: D1 --> [entity4] --> D4
...

### Evidence Documents:

D1 <Document 1>
D2 <Document 2>
...
Dn <Document n>

...

Use useful Hyper-paths to navigate between documents, verify the
intermediate facts in document contents, and keep the answer field
to the requested value only.

Return only the requested JSON object.
```

Figure 5: The prompt template used in `HyperReCo` for hyper-path-guided LLM reasoning.

- **GFM-RAG** (Luo et al., 2025b) trains a query-dependent GNN through graph completion pre-training and supervised retrieval fine-tuning to capture query–knowledge relationships over knowledge graphs.
- **G-retriever** (He et al., 2024) formulates graph retrieval as a prize-collecting Steiner tree optimization problem to select query-relevant subgraphs for LLM-based question answering.
- **Quest-GNN** (Yan et al., 2026) constructs a multi-information-level knowledge graph and performs question-guided intra- and inter-level message passing for multi-hop retrieval.
- **EHRAG** (Song et al., 2026) constructs a hybrid hypergraph with structural and semantic hyperedges and retrieves evidence using structure–semantic hypergraph diffusion with relevance refinement.

Table 7: Retrieval performance with different embedding models. We report document Recall@2 (R@2) and Recall@5 (R@5). "Embedding" denotes direct dense retrieval using the corresponding embedding model.

Embedding Model	Method	HotpotQA		MuSiQue		2Wiki	
		R@2	R@5	R@2	R@5	R@2	R@5
NV-Embed-v2 (7B) (Lee et al., 2025)	Embedding	84.1	94.4	52.7	69.5	69.1	76.5
	HGRAG	77.8	94.0	53.7	73.1	74.9	91.4
	HyperReCo	**87.1**	**96.8**	**57.2**	**74.5**	**81.2**	**97.8**
Qwen3-Emb (8B) (Zhang et al., 2025)	Embedding	74.1	88.8	46.8	62.1	66.2	74.1
	HGRAG	72.2	88.0	47.2	61.1	66.3	74.1
	HyperReCo	**82.3**	**95.0**	**52.6**	**69.3**	**80.7**	**97.6**
Qwen3-Emb (4B) (Zhang et al., 2025)	Embedding	71.6	86.3	44.5	59.0	66.1	72.7
	HGRAG	70.7	86.4	44.4	58.8	65.6	72.5
	HyperReCo	**81.5**	**94.3**	**51.2**	**68.9**	**80.8**	**97.4**
Qwen3-Emb (0.6B) (Zhang et al., 2025)	Embedding	66.5	79.2	41.7	54.3	64.6	70.3
	HGRAG	65.7	79.3	41.3	54.0	64.0	69.8
	HyperReCo	**78.5**	**91.9**	**50.3**	**66.5**	**79.6**	**97.0**

Table 8: Detailed inference efficiency across three multi-hop QA datasets. Avg. denotes the macro-average across HotpotQA, MuSiQue, and 2Wiki.

Method	HotpotQA		MuSiQue		2Wiki		Avg.	
	Time (s) ↓	EM ↑	Time (s) ↓	EM ↑	Time (s) ↓	EM ↑	Time (s) ↓	EM ↑
GraphRAG	4.10	51.4	6.18	27.0	4.79	34.7	5.02	37.7
LightRAG	8.71	9.9	8.56	2.0	7.89	2.5	8.39	4.8
GFM-RAG	3.83	56.2	**3.27**	30.2	**2.99**	69.8	3.36	52.1
HGRAG	**2.92**	55.9	3.33	36.2	2.99	68.2	**3.08**	53.4
HyperReCo	3.34	**60.7**	4.01	**39.9**	3.30	**72.2**	3.55	**57.6**

- **HGRAG** (Wang et al., 2026) represents entities as nodes and source passages as hyperedges, integrating entity- and passage-level semantic relevance through hypergraph diffusion for multi-hop retrieval.

C Additional Experiments and Analyses

C.1 Robustness to Different Embedding Models

As shown in Table 7, we further examine whether the performance of HyperReCo depends on a particular embedding model. We replace the default NV-Embed-v2 encoder with Qwen3-Embedding models of different scales (8B, 4B, and 0.6B), while keeping all other components and experimental settings unchanged. HyperReCo maintains strong retrieval performance across all embedding backbone models, even when the encoder size is substantially reduced. For example, on 2Wiki, R@5 remains at or above 97% across all four embedding models, while on HotpotQA it remains above 91% even with the 0.6B encoder. Moreover, HyperReCo consistently outperforms HGRAG under every embedding setting. These results demonstrate that HyperReCo is largely agnostic to the choice of query encoder as it remains effective across embedding models of different scales.

C.2 Detailed Inference Efficiency

To provide a more detailed view of online efficiency, we report the end-to-end inference latency and corresponding QA performance on each multi-hop QA dataset in Table 8. The end-to-end time includes both retrieval and downstream QA inference.

Table 9: Detailed EM (%) results for different hyper-path decoding strategies. Mean denotes the macro average over HotpotQA, MuSiQue, and 2Wiki.

Method	HotpotQA	MuSiQue	2Wiki	Mean
HyperReCo	**60.7**	**39.9**	**72.2**	**57.6**
w/o paths	57.7	35.6	71.6	55.0
w/o Q-PCSF	59.8	38.5	70.8	56.4
G-paths	59.9	38.8	72.0	56.9
S-forest	60.2	39.4	71.5	57.0

As shown in Table 8, `HyperReCo` maintains consistent end-to-end efficiency across all three datasets, with an average latency of 3.55 seconds per query. Its latency is close to HGRAG (3.08 s) and GFM-RAG (3.36 s), while being substantially lower than GraphRAG (5.02 s) and LightRAG (8.39 s). More importantly, `HyperReCo` achieves the highest EM on all three datasets, improving the macro-average EM by 4.2 points over HGRAG and 5.5 points over GFM-RAG. These results indicate that the additional structural reasoning and hyper-path decoding introduce only moderate computational overhead while providing consistent gains in downstream QA performance.

C.3 Additional Analysis of Hyper-Path Decoding

We provide additional details of the hyper-path decoding variants evaluated in Figure 4. For all variants, we keep the retrieved document set, downstream reader, and answer-generation procedure unchanged, and modify only how the connections among retrieved documents are constructed and presented to the reader. This isolates the contribution of gradient-guided connection scoring and Q-PCSF-based global structure optimization.

HyperReCo (GGHD). Our full decoder derives query-dependent connection scores from retriever gradients, converts them into costs, and applies Q-PCSF to select a compact forest over the retrieved documents. The forest is then expanded into hyper-paths and provided to the reader.

***w/o paths*.** This variant removes hyper-path decoding and provides only the retrieved documents to the reader, measuring the benefit of explicitly exposing evidence connections.

***w/o Q-PCSF*.** This variant keeps the gradient-based connection scores but removes global Q-PCSF optimization, selecting connections only from local scores.

***G-paths*.** G-paths directly constructs evidence paths from gradient-based importance scores, following the path extraction principle of NBFNet-style reasoning, without global forest optimization.

***S-forest*.** S-forest keeps the forest-based decoding procedure but replaces gradient-guided connection costs with semantic-similarity-based costs.

As shown in Table 9, removing explicit hyper-paths leads to the largest degradation, decreasing the mean EM from 57.6 to 55.0, which confirms that making evidence connections explicit is beneficial to the reader. Restoring explicit evidence connections improves performance, while both G-paths and S-forest remain below the full GGHD. These results suggest that the gain is not explained simply by adding arbitrary structural paths: both query-dependent connection scoring and global forest optimization contribute to the final performance.

C.4 LLM-as-a-Judge Evaluation of Hyper-Paths

Evaluation protocol. We use GPT-4o as an LLM judge to assess whether decoded hyper-paths contribute text-grounded evidence toward answering a question. We randomly sample 100 questions from each of HotpotQA, MuSiQue, and 2Wiki, resulting in 300 questions in total for evaluation. For each question, the judge receives the question, reference answer, and the paths produced by a decoding strategy, including their document order, bridge entities, and associated document text. We

Hyper-path Helpfulness Judge Prompt

```
You are judging whether each supplied hyper-path helps answer a
question.
Use only the question, reference answer, path order, bridge
entities, and supplied document text. Do not use outside knowledge.
 The reference answer is the target claim to verify, not evidence.
A shared entity, graph connection, or topical similarity alone does
 not make a path helpful.

Assess the paths as a group first so you can recognize
complementary paths. Then label each path exactly once:
- helpful=1 only when its document text provides evidence for a
claim in the reference answer, or a necessary intermediate fact/
relation in a coherent route toward that answer. A path may be
helpful even if it supplies only one necessary step and the full
set still has a gap.
- helpful=0 when it is off-topic, connected only by a superficial
shared entity, or does not contribute evidence to answering the
question. Redundancy alone is not a reason for 0 if the path
independently supports or corroborates a relevant claim.
```

Figure 6: The prompt template used for LLM-as-a-judge evaluation of hyper-path helpfulness.

Table 10: LLM-judged hyper-path utility and compactness. Helpful-path ratios are averaged over questions. GGHD achieves the highest helpful-path ratio with the fewest paths per question.

Method	Helpful-path ratio (%) ↑	Mean paths per question ↓
GGHD	**59.2**	**1.68**
w/o Q-PCSF	28.4	5.39
G-paths	43.1	2.97
S-forest	50.6	2.11

apply the same evaluation prompt to GGHD, *w/o Q-PCSF*, *G-paths*, and *S-forest*, with all variants described in detail in Section C.3. The detailed judging prompt is provided in Figure 6.

Helpfulness criterion. The judge first considers the paths jointly to identify complementary evidence, then assigns each path a binary helpfulness label. A hyper-path is considered helpful if it establishes an evidence-supported connection between retrieved documents that facilitates multi-hop reasoning toward the reference answer.

Metrics. We report the *helpful-path ratio* and the *mean number of paths per question*, measuring judged utility and output compactness, respectively. Let K_q and U_q denote the numbers of supplied and helpful paths for question q. Over the evaluated question set $\mathcal{Q}$, the two metrics are

$$\text{HelpfulRatio} = \frac{100}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{U_q}{\max(K_q, 1)}, \qquad \text{MeanPaths} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} K_q. \tag{37}$$

Results and analysis. As shown in Table 10, GGHD achieves the highest helpful-path ratio (59.2%) with the fewest paths per question (1.68). Removing Q-PCSF substantially increases the number of paths while reducing their helpfulness, suggesting that global forest optimization improves path selectivity. GGHD also outperforms G-paths and S-forest while generating fewer paths, indicating that it produces more compact and answer-relevant evidence connections.

C.5 Qualitative Analysis of Decoded Hyper-Paths

Table 11 provides the complete decoded hyper-paths for the qualitative example discussed in Section 5.4. The question asks for the death date of Lord George Scott's father. The required reasoning spans two supporting documents: D1 identifies *William Montagu Douglas Scott, 6th Duke of Buccleuch* as Lord George Scott's father, while D3 provides his death date as *5 November 1914*. Without hyper-path guidance, these two facts remain implicitly distributed across the retrieved documents, and the LLM fails to associate the entity identified in D1 with the information provided in D3.

Among the decoded routes, Path 1,

$$\text{D1} \rightarrow [\text{William Montagu Douglas Scott, 6th Duke of Buccleuch}] \rightarrow \text{D3},$$

makes the required cross-document dependency explicit. Importantly, the hyper-path does not introduce any additional factual information, and it only indicates that D1 and D3 are connected through a shared entity. This structural cue provides the LLM with a direct navigation route from identifying the father in D1 to locating the corresponding death date in D3, leading to the correct answer. The result supports our motivation that successful multi-hop reasoning depends not only on retrieving the necessary evidence, but also on making the connections required for evidence composition explicit.

Table 11: Qualitative case study of cross-document evidence composition with and without hyper-path guidance. Both settings retrieve all supporting documents (Recall@5 = 1.0), yet only the hyper-path-guided setting produces the correct answer.

Question	When did Lord George Scott's father die?
Gold Answer	5 November 1914
Supporting Docs.	["Lord George Scott", "William Montagu Douglas Scott, 6th Duke of Buccleuch"]
Retrieved Docs.	[D1: **"Lord George Scott"**, D2: "George Lyon, 5th Lord Glamis", D3: **"William Montagu Douglas Scott, 6th Duke of Buccleuch"**, D4: "Charlotte Montagu Douglas Scott, Duchess of Buccleuch", D5: "Walter Montagu Douglas Scott, 8th Duke of Buccleuch"] **(Recall@5 = 1.0)**
w/o Hyper-paths	
LLM Input	**Question:** When did Lord George Scott's father die? **Retrieved Docs:** D1, D2, D3, D4, D5
Reasoning	The passage provides information about Lord George Scott and his family, specifically mentioning that he was the son of William Montagu Douglas Scott, 6th Duke of Buccleuch. However, it does not provide a specific date of death for Lord George Scott's father. The only date mentioned in the passage is the death date of Lord George Scott himself, which is 23 February 1947. Since the passage does not include any information about the death of William Montagu Douglas Scott, 6th Duke of Buccleuch, I cannot determine when he died based on the provided text.
Answer	The passage does not provide the date of Lord George Scott's father's death. **✗ EM = 0.0, F1 = 0.0**
HyperReCo	
LLM Input	**Question:** When did Lord George Scott's father die? **Suggested Hyper-paths:** Path 1: **D1** → **[William Montagu Douglas Scott; 6th Duke of Buccleuch]** → **D3**; Path 2: D1 → [John Montagu Douglas Scott; 7th Duke of Buccleuch] → D5; **Evidence Docs:** D1, D2, D3, D4, D5
Reasoning	To find the death date of Lord George Scott's father, we identify him as William Montagu Douglas Scott, 6th Duke of Buccleuch from D1. D3 provides the death date of William Montagu Douglas Scott as 5 November 1914. Therefore, the answer is 5 November 1914.
Answer	5 November 1914 **✓ EM = 1.0, F1 = 1.0**