# Cross-Granularity Hypergraph Retrieval-Augmented Generation for Multi-hop Question Answering

**Changjian Wang**[1,2], **Weihong Deng**[1], **Weili Guan**[2], **Quan Lu**[1], **Ning Jiang**[1]
[1]Mashang Consumer Finance Co., Ltd.
[2]Harbin Institute of Technology, Shenzhen
{changjian.wang, weihong.deng, quan.lu, ning.jiang}@msxf.com, guanweili@hit.edu.cn

## Abstract

Multi-hop question answering (MHQA) requires integrating knowledge scattered across multiple passages to derive the correct answer. Traditional retrieval-augmented generation (RAG) methods primarily focus on coarse-grained textual semantic similarity and ignore structural associations among dispersed knowledge, which limits their effectiveness in MHQA tasks. GraphRAG methods address this by leveraging knowledge graphs (KGs) to capture structural associations, but they tend to overly rely on structural information and fine-grained word- or phrase-level retrieval, resulting in an underutilization of textual semantics. In this paper, we propose a novel RAG approach called HGRAG for MHQA that achieves cross-granularity integration of structural and semantic information via hypergraphs. Structurally, we construct an entity hypergraph where fine-grained entities serve as nodes and coarse-grained passages as hyperedges, and establish knowledge association through shared entities. Semantically, we design a hypergraph retrieval method that integrates fine-grained entity similarity and coarse-grained passage similarity via hypergraph diffusion. Finally, we employ a retrieval enhancement module, which further refines the retrieved results both semantically and structurally, to obtain the most relevant passages as context for answer generation with the LLM. Experimental results on benchmark datasets demonstrate that our approach outperforms state-of-the-art methods in QA performance, and achieves a 6× speedup in retrieval efficiency.

## Introduction

Retrieval-augmented generation (RAG) (Lewis et al. 2020; Gao et al. 2023) has emerged as a promising approach to enhance the capabilities of large language models (LLMs). By integrating external knowledge sources, RAG enables LLMs to access up-to-date and domain-specific information not contained within their static parameters. This augmentation improves the accuracy and reliability of the generated responses, addressing the limitations of hallucination (Huang et al. 2025) and outdated knowledge in standalone LLMs.

Despite the achievements of RAG methods, their vector similarity retrieval manner still struggles with knowledge-intensive tasks, such as multi-hop question answering (MHQA), which require knowledge integration across passages. To address this, some iterative retrieval methods have been proposed, such as IRCoT (Trivedi et al. 2023) which interleaves retrieval with chain-of-thought (CoT) (Wei et al. 2022) reasoning to improve the performance on the MHQA task. However, the low efficiency and high response latency of iterative retrieval seriously undermine the user experience. Graph retrieval-augmented generation (GraphRAG) methods (Edge et al. 2024; Guo et al. 2024; Jimenez Gutierrez et al. 2024), which establish the association of scattered knowledge through the knowledge graphs (KGs), have received widespread attention in recent years. Instead of directly retrieving text chunks, GraphRAGs retrieve nodes, edges, or subgraph communities, and expand information through graph structure, thereby enhancing the performance of retrieval and generation.

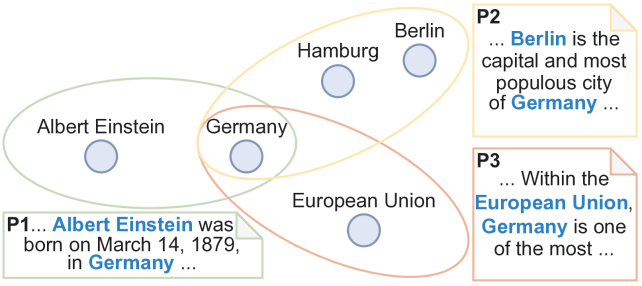


Figure 1: An example of passage association via entities. P1, P2, and P3 are passages about Albert Einstein, Germany, and the European Union, respectively.

From the perspective of retrieval granularity, traditional RAGs retrieve coarse-grained text chunks, which focus on semantic similarity but ignore the structural association of text; GraphRAGs retrieve fine-grained words or phrases, which rely heavily on the graph structure, leading to insufficient utilization of textual semantics, and the deficiencies are further amplified when the graph structure contains omissions or errors. The fragmented utilization of granularity information in existing methods overlooks the complementarity between different levels of granularity, making it difficult to effectively combine structural and semantic information. However, integrating cross-granularity structural and semantic information can effectively support the MHQA

task. As illustrated in Figure 1, we establish direct structural connections between coarse-grained passages via shared fine-grained entities as intermediaries. For a given multi-hop question such as "What is the capital of the country where Albert Einstein was born?", leveraging fine-grained entity semantic similarity, we can match the entity "Albert Einstein" and quickly locate passage P1. Then we use the shared entity "Germany" to find passages P2 and P3 through structural association. By further integrating fine-grained entity similarity and coarse-grained passage similarity, we can determine P1 and P2 (P2 is more similar to the question than P3) as the most relevant passages. We note that a hypergraph, a generalization of graphs whose edges can connect any number of nodes, can naturally model high-order structures for the MHQA task. If we treat entities as nodes and passages as hyperedges, a hypergraph can effectively model structural associations across different granularities through its inherent structure, while also facilitating semantic information propagation via hypergraph diffusion.

In this paper, we propose HGRAG, a hypergraph-based RAG method for MHQA, which enables cross-granularity integration from both structural and semantic perspectives. Structurally, we construct an entity hypergraph by first extracting entities from each passage and then treating entities as nodes and passages containing multiple entities as hyperedges. Leveraging the hypergraph, scattered passages can be connected via shared entities, thereby establishing structural associations between fine-grained entities and coarse-grained passages. Semantically, we propose a hypergraph retrieval method based on hypergraph diffusion. For each query, we construct an entity similarity vector (between query entities and corpus entities) and a passage similarity vector (between query text and passage texts). The passage similarity vector is used to form a passage-weighted hypergraph Laplacian which serves as the diffusion operator applied to the entity similarity vector. Through this process, fine-grained entity-level and coarse-grained passage-level semantic similarities are naturally integrated. Furthermore, we introduce a retrieval enhancement module to refine the hypergraph retrieval results both semantically and structurally, aiming to obtain higher-quality related passages as context for LLM answer generation. We evaluate our proposed method for MHQA on three benchmark datasets, and the experimental results show that HGRAG achieves superior performance in both QA and retrieval efficiency.

In summary, our contributions are as follows:

- We propose HGRAG, a novel hypergraph-based RAG method for MHQA. HGRAG achieves cross-granularity integration of structure and semantics through entity hypergraph construction, hypergraph retrieval, and retrieval enhancement modules.
- We evaluate our method on three benchmark datasets: HotpotQA, 2WikiMultiHopQA, and MuSiQue. Experimental results demonstrate that HGRAG significantly and consistently outperforms state-of-the-art methods.
- We compare the retrieval efficiency of HGRAG with the state-of-the-art HippoRAG 2. The results show that our approach not only significantly reduces redundant nodes (by 40%) and edges, but also achieves a 6× speedup.

## Related Work

RAG enhances LLMs by integrating retrieved knowledge and has been widely adopted in various NLP tasks. Traditional RAG methods employ retrievers to obtain texts most relevant to a given query, and use them as context for answer generation by LLMs. Early RAG commonly relied on lightweight dense retrievers (Izacard et al. 2022; Santhanam et al. 2022), which embed both queries and passages into a vector space and retrieve based on semantic vector similarity. These dense retrievers have shown better performance than sparse retrievers like BM25 (Robertson and Walker 1994). Recent LLM-based retrievers (Li et al. 2023; Muennighoff et al. 2025; Lee et al. 2025) further improve retrieval quality by leveraging richer semantic and contextual representations. However, these methods primarily focus on textual semantics while neglecting structural association, which is crucial for knowledge-intensive tasks like MHQA.

To enhance RAG's ability to associate disparate knowledge, several structure-augmented methods leveraging tree (Sarthi et al. 2024) or graph (Peng et al. 2024) have been proposed. Among them, GraphRAGs based on KGs are the most widely adopted. Graph RAG (Edge et al. 2024) leverages LLMs to construct KGs and generate community summaries over detected multi-layer communities to answer global queries. LightRAG (Guo et al. 2024) simplifies Graph RAG with dual-level retrieval strategies on KGs to construct a fast and scalable RAG system. Inspired by human memory, HippoRAG (Jimenez Gutierrez et al. 2024) employs the Personalized PageRank (PPR) (Haveliwala 2002) algorithm on KGs to identify passages associated with important entities. Building upon HippoRAG, HippoRAG 2 (Gutiérrez et al. 2025) introduces passage nodes and improves the linking strategy to propose a non-parametric continual learning framework. However, these methods heavily rely on graph structures constructed from KGs, while underutilizing coarse-grained textual semantics. Moreover, their KGs consist of triples obtained from open information extraction, which is prone to inaccuracies and omissions. The resulting noisy or incomplete graph structures can further degrade the effectiveness of GraphRAG methods.

Several iterative retrieval methods leverage subquestion decomposition (Press et al. 2023), CoT-based intermediate query generation (Trivedi et al. 2023), or KGs (Liang et al. 2024) for MHQA, requiring multiple rounds of LLM inference and retrieval. In addition, some non-RAG methods (Panda et al. 2024; Li and Du 2023) also leverage KGs or triples to introduce structure information for MHQA. Orthogonal to these approaches, we follow a non-iterative retrieval RAG paradigm, which achieves competitive performance while significantly reducing latency (Jimenez Gutierrez et al. 2024). We note that some prior studies (Luo et al. 2025; Feng et al. 2025) introduce hypergraphs into the RAG framework to model n-ary relations for domain-specific tasks. In contrast, our method leverages hypergraphs to model cross-granularity interactions between entities and passages for MHQA. Furthermore, unlike their separate retrieval strategies, our hypergraph diffusion retrieval method integrates cross-granularity semantics in a unified manner.

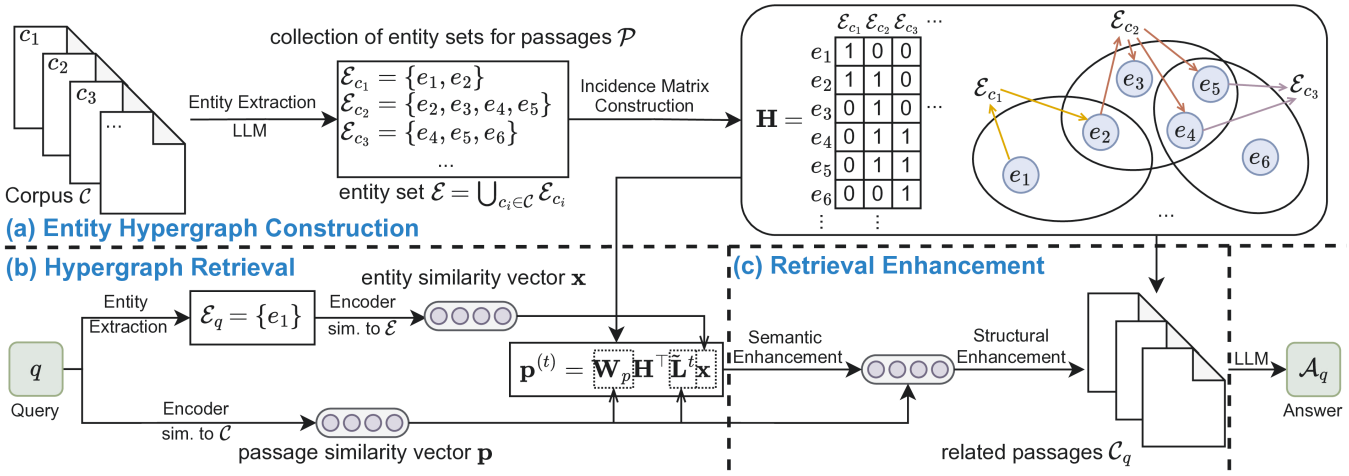


Figure 2: The overview of HGRAG. For a corpus $\mathcal{C}$ composed of multiple passages, in module (a), we extract entities for each passage using an LLM , and construct an entity hypergraph, which views entities as nodes and passages containing multiple entities as hyperedges. Given a query $q$, in module (b), we construct two semantic similarity vectors: $\mathbf{x}$ (entities in $q$ to entities in $\mathcal{E}$) and $\mathbf{p}$ ($q$'s text to passages in $\mathcal{C}$). $\mathbf{x}$ is used as the initial vector for hypergraph diffusion, while $\mathbf{p}$ is used to construct the hypergraph weight matrix and the passage-weighted hypergraph Laplacian. After $t$ steps of diffusion according to Eq. 5, we obtain a new passage relevance vector $\mathbf{p}^{(t)}$, which integrates both entity-level and passage-level semantic similarities. As shown in the illustrated hypergraph, the diffusion process follows an entity–passage–entity pattern, starting from entity $e_1$ that is most similar to the entities in $q$ (arrows in different colors indicating different diffusion steps). In module (c), $\mathbf{p}^{(t)}$ is combined with $\mathbf{p}$ for semantic enhancement. Then, guided by hyperedge associations, structural enhancement is applied to identify the most relevant passages $\mathcal{C}_q$. Finally, $\mathcal{C}_q$ are used as context for the LLM to generate the answer.

## Problem Definition

The multi-hop question answering task is to derive the correct answer to a given question by reasoning over multiple pieces of information from a provided corpus. Formally, given a natural language query $q$ and a corpus $\mathcal{C} = \{c_i \mid i = 1, 2, \ldots, n\}$ which is a set consisting of multiple passages, the task aims to obtain the answer $\mathcal{A}_q$ of $q$ according to $\mathcal{C}$, i.e., $\mathcal{A}_q = f(q, \mathcal{C})$, where $f$ is typically implemented by an LLM in state-of-the-art approaches. In real-world scenarios, the corpus $\mathcal{C}$ is usually large, making it impractical to feed the entire corpus into an LLM due to the input token limits, high computational cost, and the potential introduction of irrelevant information (noise). Therefore, MHQA tasks typically employ a retrieval step to obtain a subset $\mathcal{C}_q = \text{Retrieve}(q, \mathcal{C}) \subseteq \mathcal{C}$ consisting of passages highly relevant to the query $q$. $\mathcal{C}_q$ is used to construct a context for the LLM to generate the final answer:

$$\mathcal{A}_q = \text{LLM}\left(\text{prompt}(q, \mathcal{C}_q)\right), \tag{1}$$

where $\text{prompt}(\cdot)$ is a function that constructs a prompt for the LLM, which typically includes the query, the retrieved passages, and task instructions.

## Methodology

In this section, we will introduce the three components of our method in detail, including the entity hypergraph construction module, the hypergraph retrieval module, and the retrieval enhancement module. Figure 2 shows the overview of our method. The entity hypergraph construction module extracts entities from passage texts and constructs a hypergraph with entities as nodes and passages as hyperedges. For a given query, the hypergraph retrieval module employs hypergraph diffusion to integrate entity-level and passage-level similarities, producing refined passage relevance scores. Furthermore, the retrieval enhancement module further improves the retrieval results from the semantic and structural perspective to obtain the most relevant passages for answer generation.

### Entity Hypergraph Construction

To associate cross-granularity information from a structural perspective, we construct an entity-centric hypergraph for the corpus, where fine-grained entities are treated as nodes and coarse-grained passages containing multiple entities serve as hyperedges, and passages can be connected via shared entities. Specifically, we employ an LLM to extract entities from each passage, and then build an entity-passage incidence matrix to represent the hypergraph structure.

**Entity Extraction** As fine-grained information, entities can establish structural associations across different passages. We extract entities from a corpus using an instruction-tuned LLM. Specifically, for a given corpus, i.e., a set of passages $\mathcal{C} = \{c_i \mid i = 1, 2, \ldots, n\}$, we extract entities for each passage in the corpus via an LLM with one-shot prompting, and construct a collection of entity sets for all passages $\mathcal{P} = \{\mathcal{E}_{c_i} \mid c_i \in \mathcal{C}\}$, where $\mathcal{E}_{c_i}$ denotes the set of entities contained in passage $c_i$. By taking the union of all entities across the passages, we obtain the global entity set $\mathcal{E} = \bigcup_{c_i \in \mathcal{C}} \mathcal{E}_{c_i}$.

**Incidence Matrix Construction** In a hypergraph, a hyperedge is defined as a non-empty subset of the node set. We treat each entity as a node and each passage containing multiple entities as a hyperedge. A hypergraph can be represented by an incidence matrix. Specifically, we construct an entity-passage incidence matrix $\mathbf{H} \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{P}|}$ with entries defined as:

$$H_{ij} = \mathbb{I}[e_i \in \mathcal{E}_{c_j}] = \begin{cases} 1, & \text{if } e_i \in \mathcal{E}_{c_j} \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function. This matrix builds the relationship between entities and passages. If an entity appears in a passage, the corresponding entry in the matrix is set to 1, otherwise, it is set to 0.

## Hypergraph Retrieval

To fuse cross-granularity information on semantics, we propose a hypergraph retrieval method. This method integrates fine-grained entity-level similarity (between query entities and corpus entities) and coarse-grained passage-level similarity (between query text and passage texts). A passage-weighted hypergraph Laplacian is constructed using the passage semantic similarity, which is used to perform hypergraph diffusion starting from an entity similarity vector. After multiple steps of diffusion, we obtain a passage relevance vector that reflects the relevance between the query and each passage, which incorporates semantic information at both the entity and passage levels.

**Semantic Similarity Vector Construction** We first construct an entity similarity vector and a passage similarity vector for subsequent hypergraph diffusion.

The entity similarity vector reflects the semantic similarity between entities in the query and entities in the corpus. We first extract the entity set $\mathcal{E}_q$ from the query $q$ using an instruction-tuned LLM. For each query entity $e_q \in \mathcal{E}_q$ and corpus entity $e_i \in \mathcal{E}$, we use a dense encoder $E$ fine-tuned for retrieval to obtain their embeddings $E(e_q)$ and $E(e_i)$. The similarity between entities is then computed using the cosine similarity function $\mathrm{Sim}(\cdot)$. Let $v_i = \max_{e_q \in \mathcal{E}_q} \mathrm{Sim}\left(E(e_q), E(e_i)\right)$ and given a threshold $\eta$, we define the entity similarity vector $\mathbf{x} \in \mathbb{R}^{|\mathcal{E}|}$ with elements $x_i = v_i \cdot \mathbb{I}[v_i > \eta]$.

The passage similarity vector reflects the semantic similarity between the query and each passage in the corpus. Following the traditional RAG framework, we encode both the query text and passage texts using the encoder $E$, and compute their cosine similarities. Each element of the passage similarity vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{P}|}$ is defined as $p_i = \mathrm{Sim}(E(q), E(c_i))$, where $c_i \in \mathcal{C}$ is the $i$-th passage in the corpus.

**Passage-weighted Hypergraph Laplacian** The hypergraph Laplacian encodes the structure of a hypergraph and serves as a fundamental operator for diffusion processes. In this section, we construct a passage-weighted Laplacian matrix that incorporates passage similarity for the hypergraph diffusion described in the next section. Specifically, we first construct a diagonal hypergraph weight matrix $\mathbf{W}_p \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{P}|}$ from the passage similarity vector $\mathbf{p}$, where each diagonal entry is defined as $\mathbf{W}_p(i,i) = \mathbf{p}(i)$. Then we combine $\mathbf{W}_p$ with the hypergraph structure to construct the following symmetric normalized hypergraph Laplacian (Zhou, Huang, and Schölkopf 2006):

$$\mathbf{L} = \mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}_p\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-\frac{1}{2}}, \tag{3}$$

where $\mathbf{I}$ is the identity matrix. $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is a diagonal matrix of node degrees and it is defined as $\mathbf{D}_v(i,i) = \sum_j \mathbf{H}(i,j)$, which means the number of passages that an entity appears in. $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{P}|}$ is a diagonal matrix of hyperedge degrees, with $\mathbf{D}_e(j,j) = \sum_i \mathbf{H}(i,j)$ representing the number of entities contained in a passage.

**Hypergraph Diffusion** Hypergraph diffusion refers to the process of propagating information through a hypergraph, allowing the model to capture complex and high-order interactions by leveraging hyperedges that simultaneously connect multiple nodes. In our scenario, we use hypergraph diffusion to realize the interaction of cross-granularity semantic information between fine-grained entities and coarse-grained passages. This process starts from the initial entity similarity vector and iteratively propagates information on the hypergraph based on the passage-weighted hypergraph Laplacian.

According to the continuous-time heat kernel graph diffusion formulation $\mathbf{x}^{(t)} = e^{-t\mathbf{L}}\mathbf{x}^{(0)}$, we adopt a discrete-time first-order approximation (Chung 1997) $\mathbf{x}^{(t+1)} = (\mathbf{I} - \alpha\mathbf{L})\mathbf{x}^{(t)}$, where $t$ denotes the iteration step and $\alpha$ controls the extent of signal propagation in each iteration. For simplicity, we fix $\alpha = 1$. Let $\tilde{\mathbf{L}} = (\mathbf{I} - \mathbf{L}) = \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}_p\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-\frac{1}{2}}$ and the entity similarity vector $\mathbf{x}$ as $\mathbf{x}^{(0)}$, the result after $t$ steps of diffusion is:

$$\mathbf{x}^{(t)} = (\mathbf{I} - \mathbf{L})\mathbf{x}^{(t-1)} = \tilde{\mathbf{L}}^t\mathbf{x}. \tag{4}$$

After $t$ steps of hypergraph diffusion, we perform another entity-to-passage diffusion to obtain a new passage relevance vector $\mathbf{p}^{(t)} \in \mathbb{R}^{|\mathcal{P}|}$ which reflects the relevance between the query and passages after cross-granularity information fusion:

$$\mathbf{p}^{(t)} = \mathbf{W}_p\mathbf{H}^\top\mathbf{x}^{(t)} = \mathbf{W}_p\mathbf{H}^\top\tilde{\mathbf{L}}^t\mathbf{x}. \tag{5}$$

Reviewing the above iterative form diffusion process, semantic similarity is propagated in a random walk-like manner. The propagation path follows an entity–passage–entity pattern: in each diffusion step, information flows from entities (nodes) to passages (hyperedges) that contain them, and then to entities (nodes) in these passages. From a numerical perspective, after multiple multiplications between Laplacian matrix (containing passage similarity) and entity similarity vector, the resulting passage relevance vector assigns higher values to passages that have higher entity-level and passage-level semantic similarity to the query.

## Retrieval Enhancement

In this section, we further refine the hypergraph retrieval results with respect to semantics and structure to obtain high-quality related passages for a given query. These passages

along with the query and a task instruction are used to construct a prompt for the LLM, and the final answer is obtained from the LLM's response.

**Semantic Enhancement** The passage similarity vector $\mathbf{p}$ captures the semantic relevance from dense retrieval. We retain these original semantic retrieval results and combine them with our hypergraph retrieval results to form the final passage relevance vector:

$$\tilde{\mathbf{p}} = (1 - \beta) \cdot \mathbf{p}^{(t)} + \beta \cdot \mathbf{p}, \tag{6}$$

where the hyperparameter $\beta$ is a balancing parameter that controls the contribution of the original semantic retrieval result. This semantic enhancement strategy resembles a residual connection, which improves robustness and mitigates potential semantic degradation caused by incomplete or deficient graph structures.

**Structural Enhancement** Our retrieval goal is to obtain a set of high-quality related passages $\mathcal{C}_q$ for a query $q$. The common approach is to select the top-$k$ passages based on similarity scores. However, this approach yields a fixed-size $\mathcal{C}_q$ for each query, which lacks flexibility. The performance of question answering is sensitive to the choice of $k$: a small $k$ may fail to cover all relevant passages, while a large $k$ may introduce noise and increase the number of input tokens.

Instead of a fixed-size $\mathcal{C}_q$, we propose a dynamic-size $\mathcal{C}_q$ selection mechanism utilizing the hypergraph structure. Specifically, we define a range $[k_1, k_2]$ and obtain two top-$k$ passage sets: $\mathcal{C}_q^{k_1} = \text{Top}(\tilde{\mathbf{p}}, k_1)$ and $\mathcal{C}_q^{k_2} = \text{Top}(\tilde{\mathbf{p}}, k_2)$. Then we filter $\mathcal{C}_q^{k_2}$ by retaining only $\mathcal{C}_q^{k_1}$ and $\mathcal{C}_q^{k_1}$'s first-order hyperedge neighbors (i.e., passages that share entities with $\mathcal{C}_q^{k_1}$ in $\mathcal{C}_q^{k_2}$). More formally, for a query $q$, let $\mathbf{s} = \mathbf{H}^\top \mathbf{H} h(\mathcal{C}_q^{k_1})$ where $\left(h(\mathcal{C}_q^{k_1})\right)_i = \mathbb{I}[c_i \in \mathcal{C}_q^{k_1}]$ is a function that converts $\mathcal{C}_q^{k_1}$ into a multi-hot vector and $s_i$ reflects the number of entities shared between $c_i$ and $\mathcal{C}_q^{k_1}$. The related passage set of $q$ is defined as:

$$\mathcal{C}_q = \left\{ c_i \mid c_i \in \mathcal{C}_q^{k_2} \wedge s_i > 0 \right\}. \tag{7}$$

# Experiments

## Experimental Settings

**Datasets** To evaluate the performance of our proposed method, we perform experiments on three widely-used MHQA datasets: HotpotQA (Yang et al. 2018), 2WikiMultiHopQA (Ho et al. 2020), and MuSiQue (Trivedi et al. 2022). For a fair comparison, we use the subsets of the original datasets following HippoRAG 2 (Gutiérrez et al. 2025), which randomly extracts 1,000 questions and collect all candidate passages (including supporting and distractor passages) forming a corpus for each dataset.

**Baselines** To provide a comprehensive evaluation, we select three types of baseline methods following (Gutiérrez et al. 2025):

- **Classic Retrievers.** These include traditional dense and sparse retrieval methods that retrieve relevant passages based solely on query-passage similarity. Representative approaches include BM25 (Robertson and Walker 1994), Contriever (Izacard et al. 2022), and GTR (Ni et al. 2022).
- **Large Embedding Models.** These methods leverage LLMs for embedding generation, enabling semantically richer and more context-aware retrieval. We choose three high-performing models: Alibaba-NLP/GTE-Qwen2-7B-Instruct (Li et al. 2023), GritLM/GritLM-7B (Muennighoff et al. 2025), and NVIDIA/NV-Embedv2 (Lee et al. 2025).
- **Structure-Augmented RAG Methods.** These methods enhance traditional RAGs by integrating structural information. We include four state-of-the-art approaches: RAPTOR, Graph RAG (Edge et al. 2024), HippoRAG (Jimenez Gutierrez et al. 2024), and HippoRAG 2 (Gutiérrez et al. 2025).

**Metrics** The performance evaluation of MHQA methods is typically conducted on two subtasks: retrieval and QA. We use Recall@5 to evaluate the retrieval task, which calculates the hit rates based on the top-5 retrieval passages and assesses whether any of the top-5 retrieved passages contain the gold evidence. For the QA task, we adopt two metrics: Exact Match (EM) and F1 score. EM reflects strict correctness by requiring an exact string match with the ground-truth answers, while F1 measures the overlap between predicted and ground-truth answers at the token level.

**Implementation Details** We use NVEmbed-v2 (Lee et al. 2025) as the dense encoder $E$, and Llama-3.3-70B-Instruct (AI@Meta 2024) with temperature of 0 as our LLM for entity extraction and answer generation. We fix the $k$ range in structural enhancement with $k_1 = 5$ and $k_2 = 10$. The three hyperparameters $\eta$, $\beta$, and $t$ are chosen using 100 examples from training sets of the respective datasets. More details can be found at https://github.com/MF-AIR/HGRAG.

## Results

In this section, we present the performance of our method on both the retrieval and QA subtasks. We report the baseline results from (Gutiérrez et al. 2025) and use the same encoder, LLM, and prompt (instruction and demonstrations) for fair comparison.

| | Method | MuSiQue | 2Wiki | HotpotQA | Avg |
|---|---|---|---|---|---|
| *Classic* | BM25 | 43.5 | 65.3 | 74.8 | 61.2 |
| | Contriever | 46.6 | 57.5 | 75.3 | 59.8 |
| | GTR (T5-base) | 49.1 | 67.9 | 73.9 | 63.6 |
| *Large* | GTE-Qwen2-7B-Instruct | 63.6 | 74.8 | 89.1 | 75.8 |
| | GritLM-7B | 65.9 | 76.0 | 92.4 | 78.1 |
| | NV-Embed-v2 (7B) | 69.7 | 76.5 | 94.5 | 80.2 |
| *Structure* | RAPTOR | 57.8 | 66.2 | 86.9 | 70.3 |
| | HippoRAG | 53.2 | <u>90.4</u> | 77.3 | 73.6 |
| | HippoRAG 2 | **74.7** | <u>90.4</u> | **96.3** | <u>87.1</u> |
| | HGRAG | <u>74.1</u> | **93.0** | <u>95.5</u> | **87.5** |

Table 1: Retrieval performance on MHQA benchmarks.

| Category | Method | MuSiQue | | 2Wiki | | HotpotQA | | Avg | |
|---|---|---|---|---|---|---|---|---|---|
| | | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| *Classic* | None | 17.6 | 26.1 | 36.5 | 42.8 | 37.0 | 47.3 | 30.4 | 38.7 |
| | BM25 (Robertson and Walker 1994) | 20.3 | 28.8 | 47.9 | 51.2 | 52.0 | 63.4 | 40.1 | 47.8 |
| | Contriever (Izacard et al. 2022) | 24.0 | 31.3 | 38.1 | 41.9 | 51.3 | 62.3 | 37.8 | 45.2 |
| | GTR (Ni et al. 2022) | 25.8 | 34.6 | 49.2 | 52.8 | 50.6 | 62.8 | 41.9 | 50.1 |
| *Large* | GTE-Qwen2-7B-Instruct (Li et al. 2023) | 30.6 | 40.9 | 55.1 | 60.0 | 58.6 | 71.0 | 48.1 | 57.3 |
| | GritLM-7B (Muennighoff et al. 2025) | 33.6 | 44.8 | 55.8 | 60.6 | 60.7 | 73.3 | 50.0 | 59.6 |
| | NV-Embed-v2 (7B) (Lee et al. 2025) | 34.7 | 45.7 | 57.5 | 61.5 | 62.8 | 75.3 | 51.7 | 60.8 |
| *Structure* | RAPTOR (Sarthi et al. 2024) | 20.7 | 28.9 | 47.3 | 52.1 | 56.8 | 69.5 | 41.6 | 50.2 |
| | Graph RAG (Edge et al. 2024) | 27.3 | 38.5 | 51.4 | 58.6 | 55.2 | 68.6 | 44.6 | 55.2 |
| | HippoRAG (Jimenez Gutierrez et al. 2024) | 26.2 | 35.1 | 65.0 | 71.8 | 52.6 | 63.5 | 47.9 | 56.8 |
| | HippoRAG 2 (Gutiérrez et al. 2025) | 37.2 | 48.6 | 65.0 | 71.0 | 62.7 | 75.5 | 55.0 | 65.0 |
| *Ours* | HGRAG (top-5) | 39.4 | 50.7 | 67.7 | 74.9 | 62.8 | 75.5 | 56.6 | 67.0 |
| | HGRAG | **42.2** | **53.8** | **70.3** | **78.3** | **63.9** | **76.8** | **58.8** | **69.6** |

Table 2: QA performance on MHQA benchmarks. None denotes the performance of the LLM without any retrieved passages.

**Retrieval Results** The retrieval results on benchmark datasets are presented in Table 1. It can be observed that large embedding models outperform classic retrievers on Recall@5. This performance gain can be attributed to their foundation on LLMs, which enable them to capture richer semantic representations than small models. The structure-augmented RAG methods outperform classic retrievers, highlighting the importance of incorporating structural information into the retrieval process. However, compared to large embedding models, tree-based (e.g., RAPTOR) and graph-based (e.g., HippoRAG) approaches do not exhibit a clear advantage, which is primarily due to their heavy reliance on structural information while underutilizing semantic information. Especially on more challenging datasets such as MuSiQue requiring longer reasoning hops, structural modeling is more prone to incompleteness and errors, hindering the retrieval performance of structure-augmented RAG methods. HippoRAG 2 enhances semantic utilization by introducing passage nodes and improves the overall retrieval performance. However, due to the lack of explicit cross-granularity modeling, HippoRAG 2 introduces a large amount of redundant entities and relations, leading to lower retrieval efficiency. Our cross-granularity modeling method HGRAG obtains the best Recall@5 scores on 2Wiki, surpassing HippoRAG 2 by a margin of 2.6%. On the MuSiQue and HotpotQA datasets, HGRAG performs comparably to HippoRAG 2, and our method achieves higher retrieval efficiency (see Analysis section for details) and yields higher-quality retrieved passages which is reflected in the following QA task performance.

**QA Results** Table 2 shows the QA performance of various methods on benchmark datasets, using Llama-3.3-70B-Instruct as the QA reader. The QA performance of different methods follows a similar trend as in the retrieval task: both large embedding models and structure-augmented RAG methods generally outperform classic retrievers. However, their relative performance varies across different datasets and evaluation metrics, as each method primarily focuses on either semantic or structural information. Our proposed HGRAG consistently outperforms all baseline methods, achieving up to a 10.7% relative improvement in F1 score on the MuSiQue dataset. The experimental results demonstrate the effectiveness of our cross-granularity modeling in both structural and semantic aspects. For a fair comparison, we also include a variant HGRAG (top-5) without structural enhancement, which uses only the top-5 retrieved passages as context. HGRAG (top-5) still consistently outperforms baseline methods. A detailed ablation study is presented in the following sections. Notably, although strong retrieval performance often contributes to improved QA performance, the correlation between the two is not absolute. For example, RAPTOR achieves a higher Recall@5 than HippoRAG on MuSiQue, but its EM and F1 score is lower. This may be because some methods retrieve fewer relevant passages, but these passages are of higher quality and provide more informative context for answering the question. Compared to HippoRAG 2, although our method obtains slightly lower Recall@5 scores on the MuSiQue and HotpotQA datasets, it still outperforms HippoRAG 2 on QA performance. This suggests that the passages retrieved by our method are of higher quality.

## Analysis

**Ablation Study** To further investigate the effectiveness of our method in integrating semantic and structural information, we conduct an ablation study on the HotpotQA dataset.

For the semantic ablation, we apply two modifications: (1) replacing the hypergraph weight matrix $\mathbf{W}_p$ with an identity matrix (denoted as HGRAG w/o $\mathbf{W}_p$), and (2) removing the semantic enhancement module (denoted as HGRAG w/o SE). The retrieval results are reported in Table 3. As shown in Table 3, the semantic-related hypergraph weight matrix $\mathbf{W}_p$ has a significant impact on our method's performance, with its removal leading to a 26% drop on Recall@5. The

| Method | Recall@5 | EM | F1 |
|---|---|---|---|
| HGRAG w/o $\mathbf{W}_p$ | 69.5 | 56.9 | 68.3 |
| HGRAG w/o SE | 91.0 | 60.9 | 73.1 |
| HGRAG | **95.5** | **63.9** | **76.8** |

Table 3: Semantic ablation results on HotpotQA.

| Method | Recall@5 | Recall@10 | F1 |
|---|---|---|---|
| NV-Embed-v2 | 94.5 | 97.4 | 75.3 |
| HGRAG (top-5) | 95.5 | - | 75.5 |
| HGRAG (top-10) | 95.5 | 98.7 | 76.0 |
| HGRAG (avg-8.7) | **95.5** | **98.7** | **76.8** |

Table 4: Structural ablation results on HotpotQA.

semantic enhancement module also plays an important role, with its removal resulting in up to a 4.5% decrease on Recall@5. These results highlight the importance of semantic information in our method.

For the structural ablation, we use NV-Embed-v2, a structure-free semantic retrieval method, as a baseline. Additionally, we investigate the impact of removing the structural enhancement module from our method by using the top-5 and top-10 retrieved passages as context for answer generation, denoted as HGRAG (top-5) and HGRAG (top-10), respectively. The retrieval and QA results are shown in Table 4. HGRAG, HGRAG (top-5), and HGRAG (top-10) all outperform NV-Embed-v2 on Recall@5, Recall@10, and F1 score, with up to a 2% relative improvement in F1. Furthermore, HGRAG outperforms HGRAG (top-5) and HGRAG (top-10), achieving a maximum F1 relative improvement of 1.7%. Compared to HGRAG (top-5), HGRAG incorporates more structure-related passages, thereby enhancing QA performance. Compared to HGRAG (top-10), HGRAG achieves better QA performance with fewer passages on an average of 8.7 per query, which demonstrates that HGRAG can reduce redundant passages and improve retrieval quality with lower input token costs. The experimental results validate the effectiveness of integrating structural information within our method.

**Efficiency Analysis** To demonstrate the retrieval efficiency of our method, we conduct a comparative analysis with the state-of-the-art method HippoRAG 2 on the MuSiQue dataset. The analysis is conducted from two perspectives: graph scale and retrieval time.

The comparison of graph scale is shown in Table 5. Since two methods adopt different graph structures, it is difficult to directly compare their complexities. However, considering only the nodes that are common to both structures, the number of nodes required by HGRAG is merely 59.5% of that in HippoRAG 2. Moreover, unlike HippoRAG 2 which relies on a large number of edges (1,399,367), our method requires only a limited number of hyperedges (11,656) to establish connections among nodes.

For a fair comparison of retrieval time, we only consider

| Method | # of nodes | # of edges | # of hyperedges |
|---|---|---|---|
| HippoRAG 2 | 96,944 | 1,399,367 | - |
| HGRAG | 57,684 | - | 11,656 |

Table 5: Graph scale comparison on MuSiQue.

| | HippoRAG 2 | HGRAG | HGRAG (GPU) |
|---|---|---|---|
| Time(s) | 86.3 | 13.7 | 2.0 |

Table 6: Retrieval time comparison on MuSiQue.

the execution time of the core retrieval components of both methods, i.e., the PPR module in HippoRAG 2 and the hypergraph diffusion module in HGRAG. Both are executed on the same Intel Xeon Platinum 8558 CPU. Table 6 reports the comparison of retrieval time for 1,000 queries. Our method is approximately 6.3× faster than HippoRAG 2. Since our method is implemented using matrix and vector operations, it is well-suited for parallel acceleration on GPUs. In Table 6, we also report the retrieval time of our method on an NVIDIA H200 GPU, which is 43.2× faster than the CPU-based implementation of HippoRAG 2. Considering the additional LLM inference latency introduced by triple filtering in HippoRAG 2, HGRAG is also more efficient, since its prompt for query entity extraction is much shorter.

We further analyze the reasons behind the retrieval efficiency of our method. One reason is that the graph scale of HGRAG is smaller. As mentioned earlier, HGRAG avoids a large number of redundant nodes and edges. Another reason is that HGRAG requires fewer iterations than HippoRAG 2. On the MuSiQue dataset, HippoRAG 2 typically performs around 15 iterations per query on the entire graph, whereas HGRAG requires only 4 fixed iterations on the local hypergraph (this number can be further reduced through techniques such as fast matrix exponentiation and caching). Overall, HGRAG achieves higher retrieval efficiency than HippoRAG 2 with lower structural cost and faster retrieval time.

## Conclusion

In this paper, we propose a novel hypergraph-based RAG method called HGRAG for MHQA, which enables cross-granularity integration of structural and semantic information. HGRAG consists of three key modules: (1) the entity hypergraph construction module, which builds a hypergraph over the corpus to establish structural associations between fine-grained entities and coarse-grained passages; (2) the hypergraph retrieval module, which performs hypergraph diffusion to integrate semantic similarity at both the entity and passage levels; and (3) the retrieval enhancement module, which further refines the retrieval results both semantically and structurally to obtain the most relevant passages for answer generation with the LLM. Experimental results on benchmark datasets demonstrate that our approach outperforms state-of-the-art methods in both QA performance and retrieval efficiency.

## References


AI@Meta. 2024. Llama 3 Model Card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Chung, F. R. 1997. *Spectral graph theory*, volume 92. American Mathematical Soc.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Feng, Y.; Hu, H.; Hou, X.; Liu, S.; Ying, S.; Du, S.; Hu, H.; and Gao, Y. 2025. Hyper-RAG: Combating LLM Hallucinations using Hypergraph-Driven Retrieval-Augmented Generation. *arXiv preprint arXiv:2504.08758*.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; and Wang, H. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv e-prints*, arXiv–2312.

Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Gutiérrez, B. J.; Shu, Y.; Qi, W.; Zhou, S.; and Su, Y. 2025. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *Forty-second International Conference on Machine Learning*.

Haveliwala, T. H. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, 517–526.

Ho, X.; Nguyen, A.-K. D.; Sugawara, S.; and Aizawa, A. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6609–6625.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55.

Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; and Grave, E. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*.

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

Lee, C.; Roy, R.; Xu, M.; Raiman, J.; Shoeybi, M.; Catanzaro, B.; and Ping, W. 2025. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. In *The Thirteenth International Conference on Learning Representations*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474.

Li, R.; and Du, X. 2023. Leveraging Structured Information for Explainable Multi-hop Question Answering and Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 6779–6789.

Li, Z.; Zhang, X.; Zhang, Y.; Long, D.; Xie, P.; and Zhang, M. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Liang, L.; Sun, M.; Gui, Z.; Zhu, Z.; Jiang, Z.; Zhong, L.; Qu, Y.; Zhao, P.; Bo, Z.; Yang, J.; et al. 2024. KAG: Boosting LLMs in Professional Domains via Knowledge Augmented Generation. *arXiv preprint arXiv:2409.13731*.

Luo, H.; Chen, G.; Zheng, Y.; Wu, X.; Guo, Y.; Lin, Q.; Feng, Y.; Kuang, Z.; Song, M.; Zhu, Y.; et al. 2025. HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation. *arXiv preprint arXiv:2503.21322*.

Muennighoff, N.; SU, H.; Wang, L.; Yang, N.; Wei, F.; Yu, T.; Singh, A.; and Kiela, D. 2025. Generative Representational Instruction Tuning. In *The Thirteenth International Conference on Learning Representations*.

Ni, J.; Qu, C.; Lu, J.; Dai, Z.; Abrego, G. H.; Ma, J.; Zhao, V.; Luan, Y.; Hall, K.; Chang, M.-W.; et al. 2022. Large Dual Encoders Are Generalizable Retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 9844–9855.

Panda, P.; Agarwal, A.; Devaguptapu, C.; Kaul, M.; and Ap, P. 2024. HOLMES: Hyper-Relational Knowledge Graphs for Multi-hop Question Answering using LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13263–13282.

Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; and Tang, S. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N. A.; and Lewis, M. 2023. Measuring and Narrowing the Compositionality Gap in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 5687–5711.

Robertson, S. E.; and Walker, S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, 232–241. Springer.

Santhanam, K.; Khattab, O.; Saad-Falcon, J.; Potts, C.; and Zaharia, M. 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*, 3715–3734.

Sarthi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; and Manning, C. D. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop

Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 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)*, 10014–10037.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2369–2380.

Zhou, D.; Huang, J.; and Schölkopf, B. 2006. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19.