EHRAG: Bridging Semantic Gaps in Lightweight GraphRAG via Hybrid Hypergraph Construction and Retrieval

Yifan Song[1], Xingjian Tao[1], Zhicheng Yang[1], Yihong Luo[2], and Jing Tang[1,2]*
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
ysong853@connect.hkust-gz.edu.cn, jingtang@hkust-gz.edu.cn

Abstract

Graph-based Retrieval-Augmented Generation (GraphRAG) enhances LLMs by structuring corpus into graphs to facilitate multi-hop reasoning. While recent lightweight approaches reduce indexing costs by leveraging Named Entity Recognition (NER), they rely strictly on structural co-occurrence, failing to capture latent semantic connections between disjoint entities. To address this, we propose **EHRAG**, a lightweight RAG framework that constructs a hypergraph capturing both structure and semantic level relationships, employing a hybrid structural-semantic retrieval mechanism. Specifically, EHRAG constructs structural hyperedges based on sentence-level co-occurrence with lightweight entity extraction and semantic hyperedges by clustering entity text embeddings, ensuring the hypergraph encompasses both structural and semantic information. For retrieval, EHRAG performs a structure-semantic hybrid diffusion with topic-aware scoring and personalized pagerank (PPR) refinement to identify the top-k relevant documents. Experiments on four datasets show that EHRAG outperforms state-of-the-art baselines while maintaining linear indexing complexity and zero token consumption for construction. Code is available at https://github.com/yfsong00/EHRAG.

1 Introduction

Retrieval Augmented Generation (RAG) emerges as a promising paradigm for minimizing hallucinations in Large Language Models (LLMs) by based on external knowledge (Lewis et al., 2020; Shuster et al., 2021; Ji et al., 2023; Gao et al., 2023; Asai et al., 2023; Qiu and Tang, 2025; Tao et al., 2024; Linghu et al., 2025). While standard RAG systems excel at retrieving explicit information, they often struggle with complex queries that require multi-hop reasoning across disparate documents. To address this, Graph-based RAG (GraphRAG) approaches (Edge et al., 2024) have been introduced, which organize the corpus as a structured graph to facilitate multi-step information propagation.

Despite their effectiveness, existing GraphRAG methods face a significant efficiency bottleneck (Han et al., 2025; Peng et al., 2024). Traditional approaches typically rely on LLMs to extract entity-relation triplets and then construct the knowledge graphs. This process incurs prohibitive computational costs, which makes them impractical for large-scale corpus. To mitigate this, recent lightweight frameworks such as LinearRAG (Zhuang et al., 2025) have proposed replacing expensive LLM-based relation extraction with efficient Named Entity Recognition (NER) and modeling document structures directly.

However, we argue that these lightweight approaches introduce a critical limitation: the **Semantic Gap**. By relying solely on explicit structural co-occurrence (i.e., entities appearing in the same sentence or document), these methods fail to capture latent connections between semantically related but structurally disjoint entities. Consider the example illustrated in Figure 1. To answer the question *'Who is the spouse of the current monarch of the UK?'*, a system must connect 'monarch' in document B to 'Queen' in document A. Structure-only graph would treat these as distinct nodes because they are not same phrase and never appear in the same context window. Consequently, the reasoning chain is broken, which leads to retrieval failure.

To bridge this gap without sacrificing efficiency, we propose **EHRAG** (Efficient Hypergraph-based RAG), a novel framework that unifies structural and semantic indexing via hypergraphs. Unlike simple graphs, hypergraphs use hyperedges to connect arbitrary sets of nodes, making them naturally suitable for modeling both explicit document inclusion (structural hyperedges) and implicit semantic clusters (semantic hyperedges). Specifically, we

* Corresponding Author: Jing Tang.

Findings of the Association for Computational Linguistics: ACL 2026, pages 24638–24652

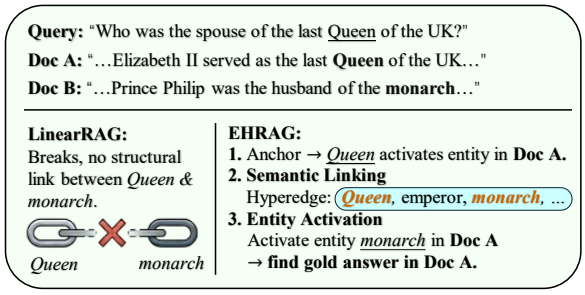

Figure 1: An example of the semantic gap. While structural graph fails to link disjoint entities (monarch and Queen), EHRAG bridges them via a semantic hyperedge, enabling the retrieval of the multi-hop reasoning.

employ lightweight NER for node extraction and construct structural hyperedges based on sentence-level co-occurrence. Simultaneously, we cluster entity embeddings to form semantic hyperedges, which connect semantically similar entities across different documents. During retrieval, we introduce a hybrid retrieval mechanism that utilizes diffusion-based entity activation to propagate query influence through explicit and latent pathways, followed by topic-aware passage scoring and PPR refinement to accurately identify the top-k relevant documents based on both content and graph structure.

To validate the effectiveness of EHRAG, we conduct experiments on four multi-hop QA benchmarks. Experimental results demonstrate that our method significantly outperforms state-of-the-art baselines including HippoRAG2 and LinearRAG, improving accuracy by up to 6.9% on 2WikiMultiHop while maintaining zero token consumption for indexing.

In summary, our contributions are as follows:

- We identify the *semantic gap* in existing lightweight GraphRAG methods, where the lack of semantic connectivity compromises multi-hop reasoning.
- We propose EHRAG, a hybrid hypergraph-based RAG framework that efficiently integrates structural co-occurrence and latent semantic correlations via entity activation by hypergraph diffusion and well designed passage scoring strategy.
- Extensive experiments across four benchmarks demonstrate that EHRAG achieves state-of-the-art performance with linear indexing complexity, offering a scalable solution for knowledge-intensive tasks.

2 Preliminaries

2.1 Backgrounds

Given a corpus $\mathcal{C} = \{d_1, \ldots, d_{|\mathcal{C}|}\}$ and a query q, retrieval-augmented generation (RAG) aims to retrieve relevant documents to support answer generation. While vanilla RAG relies on vector similarity, it often treats documents as independent units, struggling with complex queries that require multi-hop reasoning across interconnected information. To overcome this limitation, graph-based RAG structure the corpus into knowledge graphs to improving multi-step information propagation. However, traditional GraphRAG methods typically depend on LLMs to extract fine-grained entity-relation triples, which incurs prohibitive computational costs and latency.

Recently, lightweight methods (e.g., LinearRAG) replace expensive relation extraction with Named Entity Recognition (NER), modeling relationships primarily through explicit structural co-occurrence. Despite their efficiency, these pairwise graph structures often fail to capture latent semantic correlations between spatially disjoint entities. To bridge this gap, we model the corpus as a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V}$ consists of unique entities extracted via lightweight NER. Unlike simple edges, a hyperedge $e \in \mathcal{E}$ connects an arbitrary subset of entities ($e \subseteq \mathcal{V}$), enabling the unified modeling of both explicit structural inclusion (entities within a sentence) and implicit semantic grouping (entities within a semantic cluster). The hypergraph is always represented by an incidence matrix $\mathbf{H} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$, where $H(v,e) = 1$ indicates entity v belongs to hyperedge e.

2.2 Lightweight Node Extraction

Efficient graph-based RAG uses lightweight Named Entity Recognition (NER) models (e.g., GLiNER or SpaCy) to extract entities directly from the raw text, thereby bypassing the computationally intensive relation extraction (RE) process. For each sentence $s_{i,j}$ in document d_i, we extract a set of entities $E_{i,j}$. The global node set $\mathcal{V}$ is the union of all unique entities: $\mathcal{V} = \bigcup_{d_i \in \mathcal{C}} \bigcup_{s_{i,j} \in d_i} E_{i,j}$. This process scales linearly with the corpus size, effectively avoiding the $\mathcal{O}(N^2)$ complexity associated with pairwise relation modeling.

3 Methodology

To overcome the limitations of existing lightweight GraphRAG methods, we propose **EHRAG**, an efficient hpyergraph-based retrieval framework. The whole process of EHRAG is shown in Figure 2. In short, EHRAG models the corpus as a hybrid hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes (i.e. entities in the corpus), $\mathcal{E} = \mathcal{E}_{str} \cup \mathcal{E}_{sem}$ is the set of hyperedges. Here $\mathcal{E}_{str}$ represents the hyperedges generated from the structure of the documents and $\mathcal{E}_{sem}$ represents the hyperedges capturing implicit semantic relations. For each question, EHRAG first finds the similar entities as anchors in the hypergraph and then activates the relevant entities to construct a subgraph. Finally, it scores the passages with different dimensions and uses PPR to refine the scores and get the final top-k results. Next, we will give the details of each component of EHRAG and analyze why it works. Due to the page limit, we put the theoretical analysis of EHRAG in Appendix A.

3.1 Hybrid Hypergraph Construction

Unlike general graphs where edges connect pairs of nodes, a hyperedge $e \in \mathcal{E}$ in a hypergraph can connect an arbitrary number of nodes, making it ideal for modeling complex group relationships in RAG. We construct two categories of hyperedges: *Structural Hyperedges* for local context preservation and *Semantic Hyperedges* for global concept linking.

Structural Hyperedges: Explicit Context Modeling. To capture the precise context, we treat each sentence as a structural unit. Let $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ be the set of sentences in the corpus. Following previous lightweight framework (Zhuang et al., 2025; Zhao et al., 2025), we employ the lightweight Named Entity Recognition (NER) module (e.g. spaCy package) to extract the entity set $\mathcal{V}$. Evidently, entities appearing in the same sentence exhibit contextual relevance. So we construct a **structural hyperedge** $e^{str}_{s_j} = \{v_i \in \mathcal{V} \mid v_i \in s_j\}$ for each sentence s_j, which contains all entities explicitly appearing within it. Then we can get the incidence matrix $\mathbf{H}^{str} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{S}|}$ of the structural hyperedges, which is defined as $\mathbf{H}^{str}_{i,j} = 1$ if $v_i \in e^{str}_{s_j}$ and 0 otherwise. This construction ensures that entities within the same context window form a fully connected clique in the primal projection, preserving the integrity of local information.

Semantic Hyperedges: Latent Topic Discovery. Since only same entities will be combined cross the documents, graphs with only structural connections will fail to link synonymous entities appearing in disjoint contexts (e.g., GPU and Graphics Card). To bridge this semantic gap, we introduce **semantic hyperedges** via density adaptive projecting in the latent space. Let $\mathbf{x}_i \in \mathbb{R}^d$ be the text embedding of entity v_i, entities belonging to the same latent topic cluster around a centroid in the embedding space. Instead of rigid partitioning, we employ a clustering algorithm (e.g., BIRCH) to dynamically identify K latent topic centroids $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ without pre-specifying K.

For each centroid $\mathbf{c}_k$, we define a semantic hyperedge e^k_{sem} that bonds the top-D semantically closest entities. To incorporate the semantic uncertainty, we assign a continuous weight to the incidence entry based on the kernel distance:

$$\mathbf{H}^{sem}_{i,k} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_k\|^2}{\tau}\right) & v_i \in \mathcal{N}_D(\mathbf{c}_k), \\ 0 & \text{otherwise}, \end{cases} \quad (1)$$

where $\mathcal{N}_D(\mathbf{c}_k)$ denotes the set of D nearest neighbors to $\mathbf{c}_k$, and τ is a temperature parameter. This topology allows information to teleport between structurally distant but semantically related entities.

3.2 Structure-Semantic Mixed Retrieval

Based on the constructed hybrid hypergraph, EHRAG includes a novel two-stage retrieval mechanism: *Diffusion-based Activation* followed by *Topic-aware Passage Scoreing*. To sum up, it first activates the nodes in the hypergraph to generate a subset of entities that are relevant with the question. Then EHRAG scores the passages based on the connection between passages and the subset of entities with different views and uses personalized pagerank (**PPR**) to refine the scores to generates the final top-k passages for retrieval. We will start from the initialization stage to present the details of the retrieval mechanism.

3.2.1 Anchors Initialization

To initiate the retrieval, we first map the user query q to the graph. Following existing methods (Zhuang et al., 2025; Edge et al., 2024; Zhao et al., 2025), we extract a set of query entities E_q using the same NER module during hypergraph construction. For each query entity $e \in E_q$, we compute its embedding similarity with all node entities in $\mathcal{V}$ and select the most similar nodes as the anchor

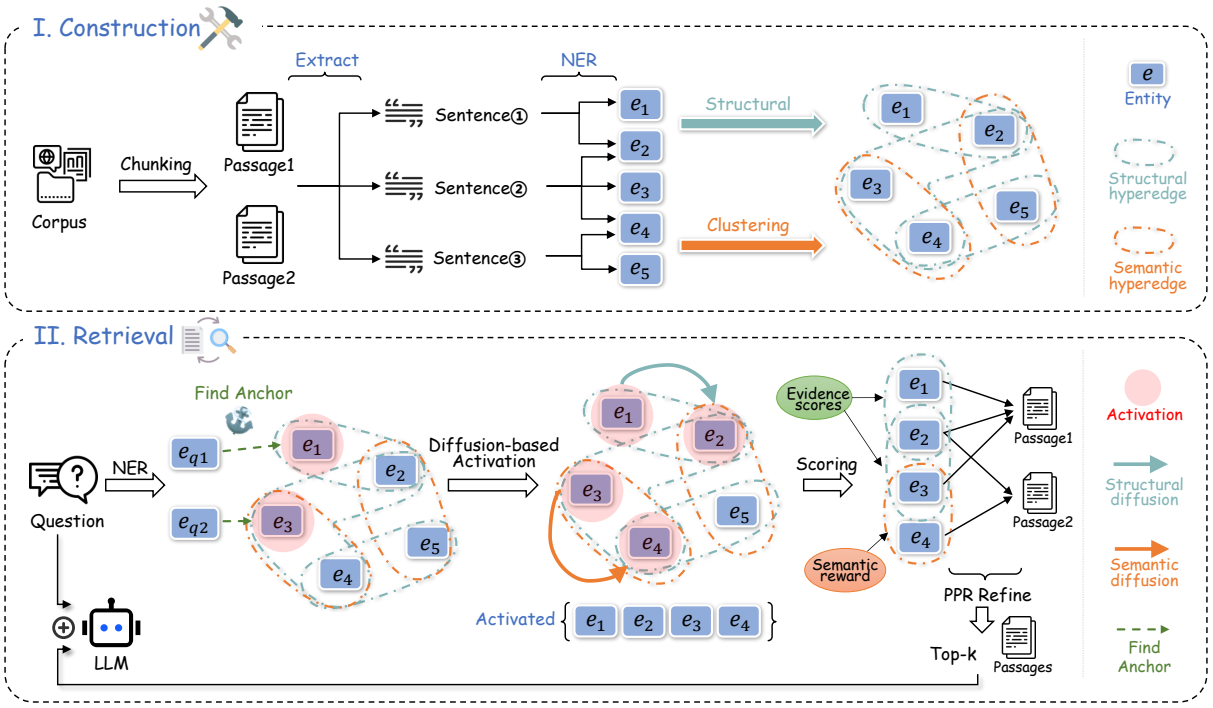

Figure 2: The overall framework of **EHRAG**. The process is divided into two phases: (1) **Offline Construction**: We extract entities using lightweight models to build structural hyperedges, while simultaneously clustering entity embeddings (e.g., BIRCH) to form semantic hyperedges. (2) **Online Retrieval**: User queries activate anchor nodes, initiating a structure-semantic hybrid diffusion process that propagates scores through the hypergraph to rank and retrieve relevant documents for generation.

to start the retrieval. We define the initial activation state $\mathbf{a}^{(0)}$ for each anchor as $\mathbf{a}^{(0)}(v) = \cos(\mathbf{x}_e, \mathbf{x}_v)$, which ensures that the retrieval starts from multiple robust entry points, tolerating extraction errors or minor morphological variations.

3.2.2 Diffusion-based Entities Activation

Starting from the seed activation $\mathbf{a}^{(0)}$, we propagate scores to identify contextually linked entities. Based on our implementation, this process is designed as a **Two-Phase Diffusion**: a one-off Semantic Expansion followed by an Iterative Structural Propagation.

Semantic Diffusion for Topic Projection. Before the iterative search, we first expand the seed set to include latent synonymous entities. This corresponds to a weighted projection through the semantic incidence matrix $\mathbf{H}^{sem}$. For the seed activation vector $\mathbf{a}^{(0)}$, we calculate the semantic expansion vector $\mathbf{a}_{sem}$ as:

$$\mathbf{a}_{sem} = \gamma \cdot \mathbf{H}^{sem}(\mathbf{H}^{sem})^{\top}\mathbf{a}^{(0)}, \tag{2}$$

where γ is a decay factor that is used to avoid the semantic shift during multi-hop diffusion. The operator $\mathbf{H}^{sem}(\mathbf{H}^{sem})^{\top}$ represents hopping from entities to their cluster centroids and then travel to the similar entities. Then we can get the initial search frontier with potential topic as $\mathbf{a}^{(1)} = \mathbf{a}^{(0)} + \mathbf{a}_{sem}$ while the initial global weight vector is $\mathbf{w} = \mathbf{a}^{(1)}$.

Iterative Structural Diffusion. We then perform a T-step structural diffusion to capture local context. Unlike standard random walks, edge weights in our graph are dynamically modulated by the query. Let $\mathbf{a}^{(t)}$ be the active frontier at iteration t. The propagation to the next hop consists of three steps:

Step 1: Entity-to-Sentence Projection. First, activation flows from entities to the sentences containing them:

$$\mathbf{s}^{(t)} = (\mathbf{H}^{str})^{\top}\mathbf{a}^{(t)}, \tag{3}$$

where $\mathbf{s}^{(t)} \in \mathbb{R}^{|\mathcal{S}|}$ represents the activation potential of each sentence.

Step 2: Query-Gated Filtering. Not all sentences containing active entities are relevant. We strictly gate the passage flow by calculating the similarity between the sentence embeddings $\mathbf{E}_S$ and the query embedding $\mathbf{x}_q$. We construct a dynamic

diagonal gating matrix $\mathbf{G}_q$:

$$\mathbf{G}_q[j,j] = \begin{cases} \mathbf{e}_{s_j}^\top \mathbf{x}_q & \text{if } s_j \in \mathbb{F}(\mathbf{E}_S \mathbf{x}_q, L), \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathbb{F}(\mathbf{x}, L)$ is the function that selects the top-L elements from $\mathbf{x}$. This step ensures that activation only flows through sentences that are semantically relevant to the user's question.

Step 3: Accumulative Update. The activation flows back to entities through the gated sentences. The new activation frontier $\Delta\mathbf{a}^{(t+1)}$ is calculated as:

$$\Delta\mathbf{a}^{(t+1)} = \mathbf{H}^{str}\mathbf{G}_q\mathbf{s}^{(t)}. \tag{5}$$

The global weight vector $\mathbf{w}$ and the frontier for the next iteration are updated as:

$$\mathbf{a}^{(t+1)} = \delta(\Delta\mathbf{a}^{(t+1)}, \epsilon), \tag{6}$$

$$\mathbf{w} \leftarrow \mathbf{w} + \mathbf{a}^{(t+1)}, \tag{7}$$

where $\delta(\mathbf{x}, \epsilon)$ is a function that only reserves the elements that are larger than ϵ in $\mathbf{x}$ and ϵ is a pruning threshold. This process repeats until convergence or maximum iterations are reached, resulting in a final weight vector $\mathbf{w}^*$ that encodes both explicit semantic similarity and implicit multi-hop contextual relevance. Then non-zero elements in $\mathbf{w}^*$ is the set of activated nodes used for scoring the related passages.

3.2.3 Topic-aware Passages Scoring

After T iterations of diffusion, we obtain a set of activated entities and topics. To retrieve the final documents, we propose a topic-aware hybrid scoring function that evaluates document relevance from three dimensions. For the question q, the score of d-th passage p_d is defined as follows:

$$\begin{aligned} S(d) = \underbrace{S_d(q,d)}_{\text{Global Context}} &+ \underbrace{\lambda_1 \sum_{v \in p_d} \log(1 + \mathbf{w}(v))}_{\text{Explicit Evidence}} \\ &+ \underbrace{\lambda_2 \cdot \log(1 + \sum_{v \in \mathcal{C}_d} S_{topic}(v))}_{\text{Semantic Reward}}. \end{aligned} \tag{8}$$

- **Global Context** $S_d(q,d)$**:** Following existing work (Zhuang et al., 2025; Zhao et al., 2025), this term is derived from a standard dense retriever (i.e. dot product of the question embeddings and the passage embeddings), ensuring that the documents broadly match the intent of the query.
- **Explicit Evidence:** It aggregates the scores of activated entities contained in the d-th passage. The logarithmic term prevents documents with simple keyword repetition from dominating the ranking.
- **Semantic Reward:** This term captures the latent thematic relevance of the document. Here, $\mathcal{C}_d$ denotes the set of unique semantic clusters (topics) appearing in document d, $S_{topic}(v)$ represents the global importance of cluster v, calculated by aggregating the semantic activation scores $\mathbf{a}_{sem}$ of all retrieved entities belonging to this cluster. This allows the system to retrieve documents that discuss the correct concept even if they lack some exact keyword matches with the query.

Scores Refinement via PPR. To enforce global consistency, we use personalized pagerank (PPR) to refine the scores on the graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ where nodes $\mathcal{V}^* = \mathcal{V}_{ent} \cup \mathcal{V}_{psg}$ include both entities and passages. The edge set $\mathcal{E}^*$ integrates structural containment (linking entity v_i to passage p_j if $v_i \in p_j$) and semantic similarity (linking entities in the same cluster). The process is defined as follows:

$$\mathbf{r}^{(t+1)} = (1-\alpha)\mathbf{M}\mathbf{r}^{(t)} + \alpha\mathbf{r}_{init}, \tag{9}$$

where $\mathbf{M}$ is the normalized adjacency matrix with both entity-passage bipartite connections and entity-entity semantic links. The restart vector $\mathbf{r}_{init}$ is initialized with S_{init} scores mapped to passage nodes and the stationary distribution $\mathbf{r}^*$ captures the final relevance of each document that can be used to determine the final top-k passages for retrieval.

3.3 Complexity Analysis

Construction Complexity. Let L be the total number of tokens in the corpus, $|\mathcal{V}|$ be the number of unique entities, and d be the embedding dimension. The construction phase consists of NER and Clustering. First, lightweight NER models (e.g., SpaCy) process text linearly with corpus length, so its time complexity is $\mathcal{O}(L)$. Next, the BIRCH algorithm constructs the CF-Tree in a single scan of the entity embeddings, whose complexity is $\mathcal{O}(|\mathcal{V}| \log(|\mathcal{V}|/B))$, where B is the branching factor. Finally, the total construction time complexity is $\mathcal{O}(L + |\mathcal{V}|d)$. Since $|\mathcal{V}| \ll L$, the whole construction takes linear time with corpus length.

Retrieval Complexity. The retrieval overhead is dominated by sparse matrix operations on $\mathcal{G}^*$, scaling linearly with the number of edges $\text{nnz}(\mathbf{A}^*)$. Physically, these edges correspond to entity mentions within the corpus. Since the number of mentions is strictly bounded by the total token count L, the complexity is $\mathcal{O}(L)$. This linear scalability ensures EHRAG remains efficient even for long-context processing.

4 Experiments

4.1 Experimental Setup

Datasets and Baselines. Following prior work (Jimenez Gutierrez et al., 2024; Zhuang et al., 2025), we evaluate our method on four datasets including three multi-hop reasoning benchmarks: **HotpotQA** (Yang et al., 2018), **2WikiMultiHop** (Ho et al., 2020), **MuSiQue** (Trivedi et al., 2022) and one domain-specific dataset **Medical** from GraphRAG-Bench (Xiang et al., 2025). Detailed dataset statistics are provided in Appendix E. We compare EHRAG against two groups of baselines: (1) direct zero-shot LLM inference, including LLaMA3-8B, LLaMA3-13B (Dubey et al., 2024), Qwen3-8B (Yang et al., 2025), GPT-3.5-trubo, GPT-4o-mini (OpenAI, 2023) and (2) retrieval-augmented-generation methods, including vanilla RAG, KGP (Wang et al., 2024), G-retriever (He et al., 2024), GraphRAG (Edge et al., 2024), RAPTOR (Sarthi et al., 2024), E^{2}GraphRAG (Zhao et al., 2025), LightRAG (Guo et al., 2024), HippoRAG (Jimenez Gutierrez et al., 2024), GFM-RAG (Luo et al., 2025), HippoRAG2 (Gutiérrez et al., 2025) and LinearRAG (Zhuang et al., 2025).

Evaluation Metric. To validate the effectiveness, we adapt two widely used metrics following existing work (Zhuang et al., 2025; Wang et al., 2025): 1. **SubEM** utilizes whether the ground truth answer is included in the response to determine the correctness for each question. 2. **LLM-Acc** uses the LLM to assess the correctness of each response. For the Medical dataset, we only evaluate the LLM-Acc metric because the ground truth answers contain multiple statements, which makes SubEM cannot validate the effectiveness.

Implementation Details. For a fair comparison, we utilize **all-mpnet-base-v2** (Song et al., 2020) as the embedding model and GPT-4o-mini (OpenAI, 2023) as the generator and evaluator for all methods. For the parameter k in the top-k retrieval, we set $k = 5$ for all methods. Further details regarding machine configuration and hyperparameters are listed in Appendix B.3.

4.2 Generation Performance

Table 1 presents the results across four datasets. We observe that EHRAG consistently outperforms all baselines. Graph-based RAG methods generally surpass zero-shot baselines and vanilla RAG, confirming the necessity of graph construction for multi-hop reasoning. Among graph-based RAG approaches, EHRAG achieves superior performance. Notably, on the **2WikiMultiHop** dataset, our method achieves a substantial gain of 3.2% in SubEM and 6.9% in LLM-Acc over LinearRAG, which is the state-of-the-art graph-based RAG method. This dataset frequently involves reasoning chains requiring entity aliasing (e.g., linking monarch of UK to the Queen). While LinearRAG relies on structural co-occurrence within sentences, EHRAG utilizes semantic hyperedges and diffusion to bridge disjoint but semantically related entities, thereby enabling more robust multi-hop retrieval. Similar gains are observed on HotpotQA ($+1.4\%$ SubEM and $+2.8\%$ LLM-Acc), indicating that incorporating latent semantic correlations improves robustness without introducing significant noise.

Furthermore, on the **Medical** dataset, we observe that several LLM-heavy graph-based methods are inferior to Vanilla RAG. This phenomenon suggests that in specialized and long-context domains, relying on low-parameter LLMs for explicit graph extraction may introduce structural noise or extraction errors that degrade retrieval quality. Conversely, EHRAG outperforms the strongest baseline by 1.6%, validating the effectiveness of our proposed semantic hypergraph construction. By leveraging implicit semantic distributions rather than relying solely on rigid, LLM-extracted edges, our method maintains robustness even facing the complex domain-specific corpus.

4.3 Ablation Study

To understand the impact of different hyperedge types, we conduct an ablation study by removing specific components. The results are summarized in Table 2.

As shown in Table 2, the full EHRAG model consistently outperforms all variants on both datasets, though the impact of specific components varies. *Str*-Diffusion proves universally critical, with its removal causing significant drops on both 2Wiki-

Method	HotpotQA		2WikiMultiHop		MuSiQue		Medical
	SubEM	LLM-Acc	SubEM	LLM-Acc	SubEM	LLM-Acc	LLM-Acc
Direct Zero-shot LLM Inference							
llama-8B	31.10	27.30	33.60	16.20	7.40	8.10	27.31
llama-13B	24.20	16.80	21.90	10.50	3.30	4.40	28.86
Qwen3-8B	25.10	34.70	29.80	27.10	6.50	19.80	58.39
GPT-3.5-turbo	33.40	43.20	28.70	31.00	10.30	21.90	45.60
GPT-4o-mini	38.90	40.20	36.30	31.40	13.60	15.80	42.10
Retrieval-Augmented-Generation Methods							
Vanilla RAG	55.70	58.60	48.60	43.00	26.10	29.60	61.68
KGP	61.50	60.90	31.60	30.00	25.60	30.10	54.22
G-retriever	42.20	40.60	46.60	27.10	14.40	15.50	50.36
GraphRAG	58.60	59.80	49.40	41.60	24.30	28.70	48.50
RAPTOR	55.90	58.30	50.10	42.10	23.30	27.40	55.75
E^2GraphRAG	61.00	63.90	54.30	38.10	23.80	26.20	58.00
LightRAG	60.30	59.50	55.20	39.00	27.40	28.60	54.36
HippoRAG	57.00	59.30	66.10	59.90	29.30	24.10	55.04
GFM-RAG	62.70	65.60	66.80	59.60	29.90	34.60	56.07
HippoRAG2	62.90	64.30	62.70	55.00	31.00	35.00	60.77
LinearRAG	<u>64.30</u>	<u>66.50</u>	<u>70.20</u>	<u>63.70</u>	<u>33.90</u>	<u>37.00</u>	<u>63.72</u>
EHRAG	**65.70**	**69.30**	**73.40**	**70.60**	**34.30**	**38.40**	**65.32**

Table 1: Result (%) of baselines and EHRAG on four benchmark datasets in terms of SubEM metric and LLM Evaluation Accuracy. The best result for each dataset is highlighted in **bold**, while the second result is indicated with an <u>underline</u>.

Variant	2WikiMultiHop	HotpotQA
EHRAG (Full)	**70.60**	**69.30**
w/o *Sem*-Diffusion	67.30(↓ 3.3%)	68.50(↓ 0.8%)
w/o Filtering	68.20(↓ 2.4%)	67.90(↓ 1.4%)
w/o *Str*-Diffusion	66.90(↓ 3.7%)	66.80(↓ 2.5%)
w/o PPR Refine	63.10(↓ 7.5%)	68.50(↓ 0.8%)

Table 2: Ablation study results (LLM-Acc). *Sem* means semantic and *Str* means structural.

MultiHop (3.7%) and HotpotQA (2.5%), confirming the necessity of iterative structural propagation. Interestingly, *Sem*-Diffusion and PPR Refine show more pronounced effects on 2WikiMultiHop (drops of 3.3% and 7.5%) compared to HotpotQA (0.8% and 0.8%). This suggests that 2WikiMultiHop contains more semantically disjoint entities and complex global dependencies, thereby relying more heavily on latent semantic bridging and global graph consistency for robust reasoning. Finally, the w/o Filtering results demonstrate that query-gated filtering consistently benefits both datasets by reducing noise. Overall, these results prove that combining structural and semantic diffusion with noise filtering and PPR refinement is essential for robust multi-hop retrieval.

4.4 Efficiency Analysis

We analyze the computational efficiency of EHRAG compared to representative baselines on the 2WikiMultiHop and HotpotQA datasets. As shown in Figure 3, we report indexing time, token consumption and overall retrieval time. EHRAG demonstrates superior efficiency compared to LLM-heavy baselines like GraphRAG and LightRAG. It maintains zero token consumption and completes indexing in just 267.5 seconds, which is comparable to the state-of-the-art lightweight method LinearRAG, confirming that the semantic hyperedges via BIRCH clustering adds negligible computational overhead while maintaining linear complexity.

In terms of retrieval efficiency, EHRAG outperforms most baselines including HippoRAG2 and LinearRAG. While E^2GraphRAG exhibits slightly lower latency, it significantly lags behind state-of-the-art graph-based RAG in generation performance and needs higher indexing overhead. Consequently, EHRAG achieves the best balance. It possesses the most advanced multi-hop reasoning capability, and its efficiency is comparable to the most lightweight baseline.

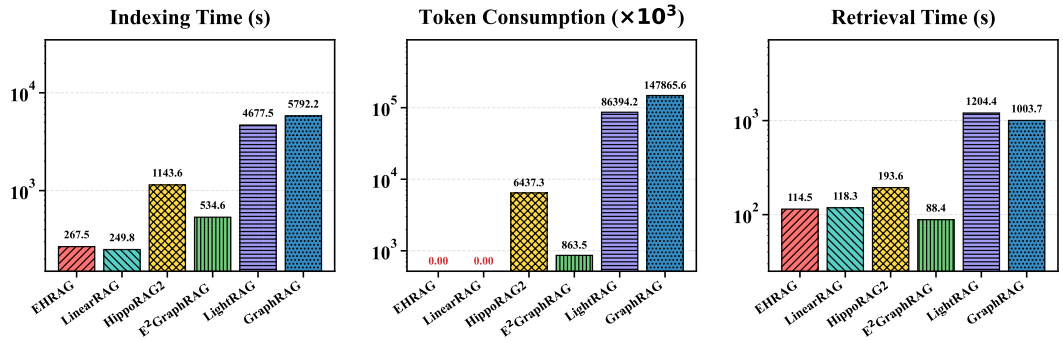

Figure 3: Efficiency comparison on 2WikiMultiHop. We report the Indexing Time, Token Consumption, and Retrieval Time for different methods. Note that the y-axis is in log scale.

Because the retrieval overhead is dominated by sparse matrix operations that scale linearly, the actual graph traversal takes only a fraction of a second per query. We profiled the average inference latency per query (in milliseconds) on the 2WikiMultiHop dataset using our standard hardware setup (NVIDIA RTX 4090). The detailed breakdown is presented in Table 3.

Retrieval Stage	Latency	Percent
1. Lightweight NER	21.4 ms	18.3%
2. Entity Embedding	28.8 ms	24.7%
3. Anchor Initialization	5.2 ms	4.5%
4. Hybrid Diffusion	4.6 ms	3.9%
5. Evidence Scoring	21.4 ms	18.3%
6. Topic Scoring	2.1 ms	1.8%
7. PPR Refinement	33.3 ms	28.5%

Table 3: Per-query inference latency breakdown on 2WikiMultiHop (NVIDIA RTX 4090).

This detailed breakdown clearly demonstrates that our core algorithmic contributions are highly efficient. The hybrid diffusion algorithm consumes merely 4.6 milliseconds per query (only 3.9% of the total time). Furthermore, while the overall passage scoring step takes time, our newly introduced topic-based scoring only takes 2.1 milliseconds (1.8%). This proves that adopting a hypergraph, performing iterative diffusion, and utilizing topic scoring does not add a heavy burden to the online inference process.

Instead, the majority of the inference time is occupied by standard pipeline components. Specifically, PPR refinement (33.3 ms), entity embedding generation (28.8 ms), standard evidence scoring (21.4 ms), and lightweight NER extraction (21.4 ms) take up the bulk of the time. Since these standard steps currently dominate the retrieval process, they represent the primary ceiling for latency. Overall, this analysis confirms that the efficiency bottleneck lies in conventional pipeline components rather than in our proposed components, demonstrating that our method achieves improved retrieval quality without sacrificing inference efficiency.

4.5 Parameter Sensitivity Analysis

To evaluate the robustness of EHRAG, we investigate the impact of four key hyperparameters on the 2WikiMultiHop and HotpotQA datasets including the number of nodes in a cluster D, the semantic propagation decay factor γ, the global context coefficient λ_1, and the semantic reward λ_2.

Impact of Cluster Size (D): The parameter D controls the amount of entity within one cluster. As shown in Figure 4(a), performance initially improves as D increases, peaking at $D = 100$ for both datasets (70.6% on 2WikiMultiHop and 69.3% on HotpotQA). Setting D larger than 100 leads to a performance decline because incorporating excessive entities introduces irrelevant noise that distracts the LLM reasoning process. Thus, we recommend set $D = 100$ initially for all datasets.

Impact of Decay Factor (γ): Figure 4(b) illustrates the effect of the semantic propagation decay factor. The model achieves optimal performance at $\gamma = 0.1$ for 2WikiMultiHop and $\gamma = 0.2$ for HotpotQA. This indicates that a moderate decay is necessary to maintain the focus of semantic expansion. For applying EHRAG on other datasets, tuning the gamma smaller than 0.5 is recommended.

Impact of Coefficients (λ_1 and λ_2): Figures 4(c) and (d) analyze the balancing coefficients. For the global context coefficient λ_1, the optimal value varies significantly between datasets (0.05 for 2WikiMultiHop while 1.5 for HotpotQA), suggesting that different datasets require varying degrees

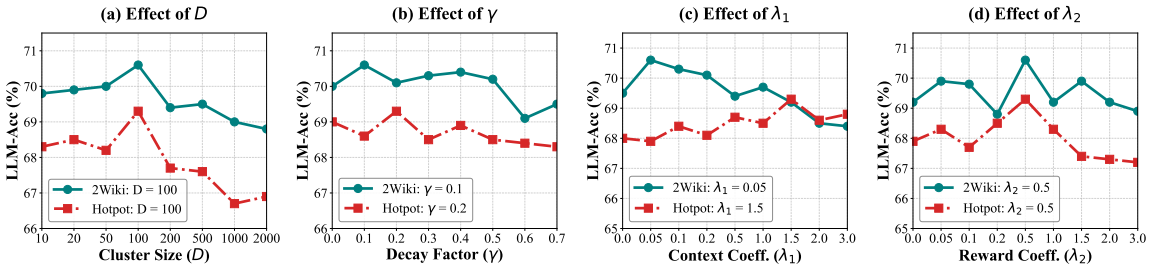

Figure 4: Parameter sensitivity analysis on 2WikiMultiHop (2Wiki) and HotpotQA (Hotpot) datasets.

of global context integration. In contrast, the semantic reward coefficient λ_2 demonstrates strong stability, with both datasets achieving their peak performance at $\lambda_2 = 0.5$.

For other hyperparameters such as threshold ϵ, sentence number L and iteration number T, we set them based on the analysis and common values in existing studies (Zhuang et al., 2025; Zhao et al., 2025; Jimenez Gutierrez et al., 2024; Gutiérrez et al., 2025). The details of all hyperparameters are listed in Appendix B.3.

5 Conclusion

In this paper, we introduced **EHRAG**, a novel lightweight graph-based RAG framework that addresses the semantic limitations of existing efficient graph-based retrieval methods. Unlike existing lightweight methods that rely solely on structural co-occurrence, EHRAG constructs a hybrid hypergraph that unifies explicit document structures with implicit semantic correlations via embedding-based clustering. This topology enables a structure-semantic hybrid diffusion process that effectively bridges disjoint but semantically related entities, facilitating robust multi-hop reasoning. Extensive experiments across four benchmark datasets demonstrate that EHRAG significantly outperforms state-of-the-art baselines while maintaining linear indexing complexity and zero token consumption. To sum up, our work offers a scalable and effective solution for knowledge-intensive tasks, demonstrating that lightweight semantic construction and semantic-based diffusion can also significantly improve the performance of graph-based RAG.

Limitations

Despite achieving state-of-the-art performance with linear indexing complexity, EHRAG remains sensitive to several key hyperparameters, such as the cluster size D and the decay factor γ, which may necessitate specific tuning for different datasets. Furthermore, while semantic hyperedges effectively bridge disjoint entities, the framework's reliance on the quality of initial entity extraction and text embeddings could potentially introduce structural noise in extremely specialized domains.

Acknowledgements

This work is partially supported by National Key R&D Program of China under Grant No. 2023YFF0725100, by the National Natural Science Foundation of China (NSFC) under Grant No. 62402410, by Guangdong Provincial Project (No. 2023QN10X025), by Guangdong Basic and Applied Basic Research Foundation under Grant No. 2023A1515110131, by Guangzhou Municipal Education Bureau (No. 2024312263), by Nansha District Project (No. 2023ZD022), and by HKUST(GZ) Kunpeng&Ascend Center of Cultivation.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*.

Yuhan Chen, Yihong Luo, Yifan Song, Pengwen Dai, Jing Tang, and Xiaochun Cao. 2024. Decoupled graph energy-based model for node out-of-distribution detection on heterophilic graphs. In *The Thirteenth International Conference on Learning Representations*.

Yuhan Chen, Yihong Luo, Jing Tang, Liang Yang, Siya Qiu, Chuan Wang, and Xiaochun Cao. 2023. Ls-gnn: towards general graph neural network in node classification by local similarity. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3550–3558.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Haoyu Han, Harry Shomer, Yu Wang, Yongjia Lei, Kai Guo, Zhigang Hua, Bo Long, Hui Liu, and Jiliang Tang. 2025. Rag vs. graphrag: A systematic evaluation and key insights. *arXiv preprint arXiv:2502.11371*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*.

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

Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. 2025. Ket-rag: A cost-efficient multi-granular indexing framework for graph-rag. *arXiv preprint arXiv:2502.09304*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Han Linghu, Qianhao Cong, Yuming Huang, Shangqi Lu, Liang Feng, and Jing Tang. 2025. Llm-powered interactive graph search: A scalable and practical approach. *Proceedings of the ACM on Management of Data*, 3(6):1–26.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2025. Gfm-rag: graph foundation model for retrieval augmented generation. *arXiv preprint arXiv:2502.01113*.

Yihong Luo, Yuhan Chen, Siya Qiu, Yiwei Wang, Chen Zhang, Yan Zhou, Xiaochun Cao, and Jing Tang. 2024. Fast graph sharpness-aware minimization for enhancing and accelerating few-shot node classification. *Advances in Neural Information Processing Systems*, 37:132364–132387.

Haitao Mao, Juanhui Li, Harry Shomer, Bingheng Li, Wenqi Fan, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang. 2024. Revisiting link prediction: a data perspective. In *The Twelfth International Conference on Learning Representations*.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

OpenAI. 2023. Gpt-4 technical report. *OpenAI Blog*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Runwen Qiu and Jing Tang. 2025. Efficient approximate nearest neighbor search via hemi-sphere centroids graph. *Proceedings of the ACM on Management of Data*, 3(6):1–26.

Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W Moore. 2011. Theoretical justification of popular link prediction heuristics. In *IJCAI proceedings-international joint conference on artificial intelligence*, volume 22, page 2722.

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

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in*

neural information processing systems, 33:16857–16867.

Yifan Song, Xiaolong Chen, Wenqing Lin, Jia Li, Chen Zhang, Yan Zhou, Lei Chen, and Jing Tang. 2024. Efficient graph embedding generation and update for large-scale temporal graph. *Proceedings of the VLDB Endowment*, 18(4):929–942.

Yifan Song, Darong Lai, Zhihong Chong, and Zeyuan Pan. 2021. Dynamic network embedding by time-relaxed temporal random walk. In *International Conference on Neural Information Processing*, pages 426–437. Springer.

Yifan Song, Fenglin Yu, Yihong Luo, Xingjian Tao, Siya Qiu, Kai Han, and Jing Tang. 2025. Ddfi: Diverse and distribution-aware missing feature imputation via two-step reconstruction. *arXiv preprint arXiv:2512.06356*.

Xingjian Tao, Yiwei Wang, Yujun Cai, Zhicheng Yang, and Jing Tang. 2024. Are llms really not knowledgable? mining the submerged knowledge in llms' memory. *arXiv preprint arXiv:2412.20846*.

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

Shu Wang, Yixiang Fang, Yingli Zhou, Xilin Liu, and Yuchi Ma. 2025. Archrag: Attributed community-based hierarchical retrieval-augmented generation. *arXiv preprint arXiv:2502.09891*.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Conference on Artificial Intelligence (AAAI)*.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2025. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *arXiv preprint arXiv:2506.05690*.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Fangyuan Zhang, Zhengjun Huang, Yingli Zhou, Qintian Guo, Zhixun Li, Wensheng Luo, Di Jiang, Yixiang Fang, and Xiaofang Zhou. 2025a. Erarag: Efficient and incremental retrieval augmented generation for growing corpora. *arXiv preprint arXiv:2506.20963*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025b. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Yanqiu Zhang, Zhen Xu, DongDong Huo, Xiaokun Guo, Qihui Zhou, and Yan Zhang. 2025c. Adagcrag: Adaptive graph-chunk retrieval for lightweight rag. In *International Semantic Web Conference*, pages 594–610. Springer.

Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. 2025. Eˆ 2graphrag: Streamlining graph-based rag for high efficiency and effectiveness. *arXiv preprint arXiv:2505.24226*.

Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang, Sicong Liang, Xilin Liu, Yuchi Ma, and 1 others. 2025. In-depth analysis of graph-based rag in a unified framework. *arXiv preprint arXiv:2503.04338*.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2025. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. *arXiv preprint arXiv:2510.10114*.

A Theoretical Analysis

In this section, we provide a rigorous justification for EHRAG using the Latent Space Model (LSM) (Mao et al., 2024). We demonstrate how constructing semantic hyperedges via BIRCH clustering explicitly bridges the gap between disjoint entities by strictly tightening the upper bound of their latent distance.

A.1 Latent Space Modeling

Following (Mao et al., 2024; Sarkar et al., 2011), we model the corpus graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in a D-dimensional latent Euclidean space $\mathbb{R}^D$. Each entity v_i has a latent position $\mathbf{z}_i$ and an influence radius r_i. The probability of a link between entities i and j is governed by their latent distance $d_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|$:

$$P(i \sim j \mid d_{ij}) = \frac{1}{1 + \exp(\alpha(d_{ij} - \tau))} \tag{10}$$

where $\alpha > 0$ is a scaling factor. A smaller d_{ij} implies a higher retrieval probability.

A.2 The Semantic Gap in Structural Retrieval

In purely structural RAG systems, links rely on explicit sentence co-occurrence. Let $\eta_{ij}^{str} = |\mathcal{N}_{str}(i) \cap \mathcal{N}_{str}(j)|$ be the number of structural common neighbors. For semantically related but spatially disjoint entities (e.g., in different documents), we have $\eta_{ij}^{str} \to 0$. According to Proposition 1 in (Mao et al., 2024), the latent distance d_{ij} is loosely bounded:

$$d_{ij} \leq 2\sqrt{r_{ij}^{\max} - \left(\frac{\eta_{ij}^{str}/N - \epsilon}{V(1)}\right)^{2/D}} \tag{11}$$

where $r_{ij}^{\max} = \max(r_i, r_j)$ and $V(1)$ is the unit hypersphere volume. As $\eta_{ij}^{str} \to 0$, the subtracted term vanishes, leaving d_{ij} close to its maximum possible value $2\sqrt{r_{ij}^{\max}}$, resulting in $P(i \sim j) \to 0$. This mathematically quantifies the *Semantic Gap*.

A.3 Bridging via Clustering-Induced Hyperedges

EHRAG overcomes this by utilizing BIRCH to construct semantic hyperedges.

Definition 1 (Cluster-Induced Connectivity)**.** Let $\mathcal{C} = \{C_1, \ldots, C_K\}$ be the clusters generated by BIRCH. For a cluster C_k with centroid μ_k and threshold radius T, any entity $v_i \in C_k$ satisfies $\|\mathbf{x}_i - \mu_k\| \leq T$ in the embedding space. We construct a semantic hyperedge e_{sem}^k connecting all $v \in C_k$.

This construction transforms feature compactness into structural connectivity. We formalize this effect as a *Feature Proximity (FP)* term β_{ij}, which acts as a "synthetic" common neighbor probability.

Theorem 1 (Cluster-Tightened Distance Bound)**.** *For any two entities i, j belonging to the same semantic hyperedge e_{sem}^k (i.e., $i, j \in C_k$), the BIRCH clustering guarantees a feature proximity lower bound $\beta_{min} \propto \exp(-4T^2)$. Consequently, the latent distance d_{ij} is tightly bounded by:*

$$d_{ij} \leq 2\sqrt{r_{ij}^{\max} - \Delta_{sem}},$$
$$\text{where } \Delta_{sem} = \left(\frac{\beta_{ij} + \mathcal{A}(r_i, r_j, d_{ij})}{V(1)}\right)^{2/D}. \tag{12}$$

Here, $\mathcal{A}(\cdot)$ is the intersection volume of influence spheres, and β_{ij} represents the probabilistic connection strength induced by the semantic cluster.

Proof. We extend the proof from (Mao et al., 2024). The existence of a semantic hyperedge e_{sem}^k containing i and j introduces a direct path in the hypergraph. In the LSM, this is equivalent to injecting a non-zero feature proximity term β_{ij} into the intersection volume. Since $i, j \in C_k$, by triangle inequality, their embedding distance $\|\mathbf{x}_i - \mathbf{x}_j\| \leq 2T$. Mapping this feature distance to the connection probability space, we obtain $\beta_{ij} > 0$.

Substituting the augmented volume $\mathcal{A}' = \beta_{ij} + \mathcal{A}(r_i, r_j, d_{ij})$ into the hypersphere packing bound:

$$\frac{\mathcal{A}'}{V(1)} \leq \left(r_{ij}^{\max} - \left(\frac{d_{ij}}{2}\right)^2\right)^{D/2}$$
$$\implies d_{ij} \leq 2\sqrt{r_{ij}^{\max} - \left(\frac{\beta_{ij} + \mathcal{A}}{V(1)}\right)^{2/D}} \tag{13}$$

Critically, for disjoint entities where $\mathcal{A} \approx 0$, the term β_{ij} (guaranteed by BIRCH clustering) ensures that the subtraction term Δ_{sem} is strictly positive. Since $f(x) = \sqrt{C - x}$ is monotonically decreasing, a larger β_{ij} strictly decreases the upper bound of d_{ij}. □

Implication: Eq. 12 proves that EHRAG guarantees a tighter latent distance bound than LinearRAG. Even if $\eta_{ij}^{str} = 0$, the semantic term β_{ij} forces the latent distance to shrink, thereby theoretically ensuring a higher retrieval probability for semantically similar entities.

Category	Content
Question	"Which film was released first, Aas Ka Panchhi or Phoolwari?"
Ground Truth	Phoolwari (1946)
Support Context	[Aas Ka Panchhi: 1961] ↔ [Phoolwari: 1946]
LinearRAG	**Retrieved context (Top 5):** 1) ✘ *Shah Muhammad... Phulwarisharif*: ...born in Phulwarisharif, Bihar... (Location noise) 2) ✘ *Pyar Ka Bandhan*: ...is a 1963 Hindi film... 3) ✘ *Student of the Year*: ...released on 19 October 2012... 4) ✘ *Hum Dil De Chuke Sanam*: ...released internationally in 1999... 5) ✘ *Phool Aur Kaante*: ...began career with Phool Aur Kaante in 1991... **Prediction:** ✘ Aas Ka Panchhi
EHRAG	**Retrieved context (Top 5):** 1) ✔ **Aas Ka Panchhi (1961)** & **Phoolwari (1946)**: ...Phoolwari is a 1946 film. Aas Ka Panchhi is a 1961 movie... 2) ✔ *Hum Dil De Chuke Sanam*: ...adaptation of Maitreyi Devi's novel... (Relatively irrelevant but correctly ranked) 3) ✘ *Vaibhavi Merchant*: ...choreography work in Bollywood films... 4) ✘ *Nitin Chandrakant Desai*: ...noted Indian art director... 5) ✔ **Aas Ka Panchhi (1961)**: ...1961 Hindi movie produced by J. Om Prakash... **Prediction:** ✔ Phoolwari

Table 4: **Detailed Case Study comparison.** Our method (EHRAG) successfully retrieves the exact release years for both movies, while the Baseline is misled by geographic entities and recent film noise.

Category	Method	Key Characteristics
Zero-shot LLM	LLaMA3 (8B/13B), Qwen3-8B	Evaluates the internal knowledge of state-of-the-art open-source LLMs.
	GPT-3.5-turbo, GPT-4o-mini	Proprietary models used to establish a performance upper bound for zero-shot inference.
Standard RAG	Vanilla RAG	Standard retrieval-augmented generation relying on vector similarity.
Graph-based RAG	GraphRAG, KGP, G-retriever	Traditional GraphRAG methods that typically utilize LLMs for entity-relation triple extraction.
	RAPTOR	Builds tree-organized indices through recursive abstractive processing.
Lightweight RAG	HippoRAG, HippoRAG2	Neurobiologically inspired methods utilizing Personalized PageRank (PPR).
	LinearRAG	A state-of-the-art lightweight framework that models document structures directly via NER.
	E2GraphRAG, LightRAG	Recent efficient frameworks designed to streamline graph-based retrieval.

Table 5: Comprehensive overview of baseline methods.

B Related Work

B.1 LLM-based GraphRAG via Triple Extraction

Graphs serve as a natural and expressive representation for encoding relational knowledge, and recent advances in graph learning (Song et al., 2021, 2024, 2025; Chen et al., 2024; Luo et al., 2024; Chen et al., 2023) have significantly improved the ability to reason over such structured information. Building on this foundation, GraphRAG transforms unstructured text into structured Knowledge Graphs (KGs) to explicitly model entity relationships (Zhang et al., 2025a; Zhou et al., 2025; Xiao et al., 2025; Zhang et al., 2025b; Xiang et al., 2025; Mavromatis and Karypis, 2024). Prominent frameworks like Microsoft's GraphRAG (Edge et al., 2024) employ Large Language Models (LLMs) for

Hyperparameter	HotpotQA	2WikiMultiHop	MuSiQue	Medical
NER Model	`en_core_web_trf`	`en_core_web_trf`	`en_core_web_trf`	`en_core_sci_scibert`
Max Iterations (T)	3	3	5	3
Pruning Threshold (ϵ)	0.5	0.4	0.4	0.5
Passage Ratio (λ_1)	1.5	0.05	2.0	1.5
Sentence Number (L)	1	1	4	1
Cluster Threshold (D)	100	100	100	100

Table 6: Detailed hyperparameter settings for EHRAG across four benchmark datasets.

Open Information Extraction (OpenIE) to construct entity-relation triples, subsequently using community detection (e.g., Leiden) to support global query answering. Similarly, RAPTOR (Sarthi et al., 2024) builds tree-structured indices via recursive clustering.

However, these methods suffer from a construction bottleneck (Edge et al., 2024; Sarthi et al., 2024; Jimenez Gutierrez et al., 2024). The reliance on LLMs for triple extraction incurs prohibitive computational costs that scale polynomially with corpus size. Furthermore, rigid Named Entity Recognition (NER) often leads to *Semantic Loss*—semantically related entities that do not physically co-occur or fail extraction remain disconnected, disrupting retrieval pathways.

B.2 Lightweight Graph Construction

To mitigate high costs, recent work explores lightweight strategies (Huang et al., 2025; Pan et al., 2024; Zhang et al., 2025c). HippoRAG (Jimenez Gutierrez et al., 2024) and its successor (Gutiérrez et al., 2025) leverage Personalized PageRank (PPR) on existing KGs to simulate associative memory. LinearRAG (Zhuang et al., 2025) introduces a tri-Graph architecture using lightweight tools to connect entities, sentences and passages, achieving high efficiency and outstanding QA performance on various datasets.

Positioning of EHRAG: While the aforementioned methods reduce costs, they often overlook deep semantic correlations, relying primarily on physical textual co-occurrence. Our proposed **EHRAG** inherits the efficiency of lightweight construction (linear complexity) while introducing **Semantic Hyperedges**. By leveraging hypergraph topology, we explicitly resolve semantic disconnects without increasing construction overhead.

B.3 Experimental Configuration

All experiments were conducted on a high-performance computing server equipped with two Intel(R) Xeon(R) Platinum 8377C CPUs , theNVIDIA RTX 4090 GPU (24GB VRAM), and 512GB of RAM. Hyperparameters were tuned specifically for each dataset to handle varying reasoning complexities. The dataset-specific configurations are summarized in Table 6.

C Case Study

To intuitively demonstrate how EHRAG bridges the semantic gap, we present a qualitative analysis in Table 4 using a comparative query from 2WikiMultiHop. EHRAG utilizes semantic hyperedges to capture latent correlations. Even though the correct movie *"Phoolwari"* (1946) does not share explicit structural neighbors with the query context, our clustering-based semantic construction successfully maps the query entity to the correct latent topic. This activates the relevant passage containing *"Phoolwari is a 1946 film..."*, enabling the model to correctly identify that *"Phoolwari"* (1946) was released before *"Aas Ka Panchhi"* (1961). This case highlights EHRAG's robustness in filtering keyword noise and retrieving semantically aligned evidence.

D Baseline Descriptions

We compare our method against two primary groups of baselines: Zero-shot LLM and RAG methods. The details are shown in Table 5.

E Dataset Descriptions

We evaluate EHRAG on four benchmark datasets, including three multi-hop reasoning benchmarks and one domain-specific dataset:

- **HotpotQA**: A widely used multi-hop reasoning benchmark that requires finding and integrating evidence across multiple documents.
- **2WikiMultiHop**: This dataset frequently involves reasoning chains that require entity aliasing, such as linking synonymous but disjoint entities. It is characterized by having semantically

disjoint entities and complex global dependencies.

- **MuSiQue**: A dataset comprised of multi-hop questions generated via single-hop question composition, testing deep logical integration.
- **Medical**: A domain-specific dataset from GraphRAG-Bench. We utilize the LLM-Acc metric for this dataset because the ground truth answers contain multiple statements, making Exact Match metrics (SubEM) less effective.