Collision to Cognition: Hash-Driven Graph Construction for Efficient RAG

Chuang Zhou[1,2], **Zheng Yuan**[1,2], **Linhao Luo**[3],
Zhaozhuo Xu[4], **Yilin Xiao**[1], **Junnan Dong**[2,*],
Siyu An[2], **Di Yin**[2], **Xing Sun**[2], **Xiao Huang**[1]
[1]The Hong Kong Polytechnic University, China; [2]Tencent Youtu Lab, China;
[3]Monash University, Australia; [4]Stevens Institute of Technology, USA
[1]{chuang-qqzj.zhou, yzheng.yuan, yilin.xiao} @connect.polyu.hk
xiaohuang@comp.polyu.edu.hk; [2]hansonjdong@tencent.com; [4]zxu79@stevens.edu

Abstract

Retrieval-Augmented Generation (RAG) has long been a promising paradigm for enhancing large language models (LLMs) with external knowledge. Current embedding-based methods can capture semantic similarity but struggle to establish fine-grained, interpretable logical connections. Recently, GraphRAG has gained increasing popularity for its capability in modeling logical relations. However, it requires substantial API token usage for triple extraction or textual summarization during graph construction, which makes the entire process inefficient and expensive. In this paper, we propose MeshRAG, a novel framework that **M**ines **E**fficient **S**tructures via **H**ashing for improved RAG. We jointly model chunk-level interactions and community organizations, with the global graph structure naturally emerges from locality-sensitive hash collisions. By replacing neural embedding search with lightweight bitwise operations, MeshRAG automates a simple and rapid graph construction process. Furthermore, the hash collision mechanism provides transparent evidence for logical connections and retrieval decisions. Experimental results show that MeshRAG outperforms existing baselines, while its graph construction requires no GPU resources or API budget and can structure over ten thousand chunks within a few minutes.

1 Introduction

Retrieval-Augmented Generation (RAG) has been widely employed as a fundamental paradigm for mitigating hallucinations and enhancing the factual grounding of large language models (LLMs) by retrieving relevant information from external knowledge sources (Arslan et al., 2024; Notarangelo et al., 2016; Dong et al., 2024a; Chen et al., 2026). However, it simply treats corpora as flattened pieces and primarily relies on dense vector similarity, exhibiting inherent limitations in handling complex queries that require multi-step reasoning or integration of interconnected facts. In response, Graph-enhanced RAG (GraphRAG) has emerged as a promising advancement (Han et al., 2024; Zhang et al., 2025). By structuring knowledge into explicit graphs, where entities serve as nodes and their relationships as edges, GraphRAG equips LLMs with a powerful mechanism to navigate and reason over pre-defined connections. This explicit representation has been proven particularly effective for complex and multi-hop question answering.

Current graph-based variants can be broadly categorized into two paradigms: the first reorganizes the corpus into a tree-like hierarchical structure by clustering text chunks or generating abstractive summaries (Sarthi et al., 2024; Edge et al., 2024); the second relies on language models to extract relational triples and forms them into a structured knowledge graph (Gutiérrez et al., 2024; Luo et al., 2025). While effective in enhancing reasoning capabilities, both paradigms introduce significant computational overhead during the graph construction phase. Approaches relying on summarization or triple extraction inevitably require substantial language model involvement, and clustering-based methods depend on computing dense vector representations. This leads to efficiency problems as the corpus size grows. Furthermore, most clustering algorithms force each text chunk into one category, overlooking the inherent multifaceted property that the same passage can serve different semantic roles or belong to multiple topics simultaneously.

To this end, we are driven to design a simple and rapid graph construction method. Hash functions present a unique opportunity due to their simplicity and fast bitwise operations, offering a lightweight alternative for capturing similarity. However, it cannot be straightforwardly applied for two reasons. (1) **Information Sparsity:** a single hash value is too coarse to capture rich semantic relationships, and (2) **Structural Deficiency:** independent hash

*Junnan Dong is the corresponding author and serves as the project lead of this work.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25224–25240
July 2-7, 2026

functions fail to naturally organize the corpora into a traversable and multi-granular graph structure.

To bridge this gap, we propose MeshRAG, a lightweight framework for **M**ining **E**fficient **S**tructures via **H**ashing to enhance RAG. Its architecture directly addresses these challenges through corresponding innovations: (1) achieves semantic-aware similarity estimation by replacing neural embeddings with locality-sensitive hashing (LSH). (2) constructs a graph that both captures broad semantic communities and preserves nuanced pairwise connections between individual chunks. Such a hash-driven framework not only rapidly builds the graph but also provides interpretable logic evidence both at the construction and retrieval stages.

Summary of Contributions:

- We pioneer the integration of LSH into automated graph construction for RAG, enabling a more effective and efficient pipeline.
- MeshRAG exploits knowledge from corpora at dual levels, simultaneously modeling fine-grained pairwise chunk relationships and discovering high-level communities, thereby allowing text to serve different roles in different contexts.
- Our framework achieves rapid graph construction within only a few minutes, eliminating the need for both LLM-driven token consumption and GPU-intensive dense vector operations.
- Beyond SOTA performance, we provide detailed interpretable construction logic and transparent retrieval pathways, offering new insights into explainable knowledge-enhanced generation.

2 Preliminaries

We represent the structured knowledge derived from a textual corpus as an attributed graph $G = (V, E, \phi)$. Here, $V = \{v_1, v_2, \ldots, v_N\}$ denotes the set of vertices, each corresponding to a distinct textual chunk c_i from the corpus $\mathcal{C}$. The set $E \subseteq V \times V$ comprises undirected edges, where an edge $(v_i, v_j) \in E$ signifies a meaningful semantic relationship or similarity between the chunks c_i and c_j. The function $\phi : V \to \mathbb{H}$ maps each vertex to a compact hash signature, which serves as its primary feature for efficient computation. The graph G naturally exhibits a two-tier structure: at the fine-grained *chunk level*, vertices are interconnected via edges E capturing pairwise semantic affinities; at the coarse-grained *community level*, densely connected subsets of V are abstracted into hyper-nodes, representing major thematic clusters.

3 Methodology

Our primary objective is to efficiently construct a reasoning graph from a textual corpus that enables the rapid and interpretable retrieval of information most relevant to a user's query. The term **Mesh** emphasizes the flexible and overlapping network induced by hash collisions. This process yields a two-level graph: a fine-grained layer of individual text chunks connected by their mutual relatedness, and a coarse-grained layer of semantic communities reflecting common clustering topics. By treating hash collisions as a robust estimate of local semantic density, the entire graph is constructed in a bottom-up manner. In the following sections, we detail the three core components of MeshRAG: (1) hash-based indexing for efficient node representation, (2) collision-driven community detection for automated graph construction, and (3) topology-aware retrieval over the resulting graph structure.

3.1 Hash-Based Local Connectivity

To identify chunk-level connections, the indexing phase transforms unstructured textual chunks into compact, comparable representations that capture semantic similarity and enable fast calculation. Let a corpus consist of N chunks, $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$. For each chunk c_i, we first extract its entity set $E_i = \{e_{i1}, e_{i2}, \ldots\}$ using a standard regular-expression based function to avoid the high cost of LLM-based extraction (Gutiérrez et al., 2024). As the following definition shows, the semantic similarity between two chunks is quantified by the Jaccard similarity of their entity sets:

$$J(c_i, c_j) = \frac{|E_i \cap E_j|}{|E_i \cup E_j|}.$$

However, directly computing $J(c_i, c_j)$ for every pair of chunks becomes computationally expensive as N grows. To enable an efficient and effective approximation, we employ the MinHash algorithm.

MinHash Signatures. We use a family of k independent hash functions, $\mathcal{H} = \{h_1, h_2, \ldots, h_k\}$, each mapping entities to distinct integers. For each chunk c_i and hash function h_l, we compute:

$$m_l(c_i) = \min_{e \in E_i} h_l(e),$$

which is the minimum hash value of all entities in c_i under h_l (Shrivastava and Li, 2014). The MinHash signature of chunk c_i is then the vector of these minima, which is presented in the format below:

$$\mathbf{s}_i = [m_1(c_i),\ m_2(c_i),\ \ldots,\ m_k(c_i)] \in \mathbb{Z}^k.$$

To achieve sub-linear search time for finding semantically similar chunks within the large corpus $\mathcal{C}$, we organize the signatures using LSH (see Appendix A.1). To amplify the similarity gap, we employ the standard **banding technique**. Specifically, we divide the k-dimensional signature $\mathbf{s}_i$ into b bands, each containing r rows ($k = b \times r$). Two chunks are considered candidate neighbors if their signatures are identical in a certain number of bands. The corresponding probability is as follows:

$$P_{\text{sim}}(c_i, c_j) = 1 - (1 - [J(c_i, c_j)]^r)^b .$$

We construct L ($L = 8$) independent LSH tables, each applying this banding scheme with a different random permutation of the signature bits. A chunk c_i is indexed by inserting its signature $\mathbf{s}_i$ into all L tables. MinHash provides a link between hash collisions and semantic similarity, as stated in the following property. For any hash function $h_l \in \mathcal{H}$,

$$P\left[m_l(c_i) = m_l(c_j)\right] = J(c_i, c_j),$$

$$\hat{J}(c_i, c_j) = \frac{1}{k} \sum_{l=1}^{k} \mathbb{I}\left[m_l(c_i) = m_l(c_j)\right],$$

where the probability is taken over the random choice of h_l. Each component provides an unbiased estimator of Jaccard similarity. The similarity between two chunks can be efficiently estimated via Hamming distance. This multi-table approach further increases the probability of capturing true near neighbors while maintaining efficient storage.

3.2 Collision-Driven Community Detection

Building upon the LSH signatures generated in the previous stage, this phase automatically organizes the textual chunks into semantically coherent communities. Passages that frequently collide (hash to the same bucket) across multiple LSH tables are more semantically similar and tend to gather to the same community in a bottom-up manner. We thus iteratively merge similar small buckets into larger, more stable communities. Unlike traditional hard clustering algorithms that force each chunk into a single category, our collision-driven approach naturally accommodates the *polysemous* nature of text, where the same chunk can play different semantic roles in different contexts. For instance, as shown in Figure 1, a passage describing *wearable devices* can simultaneously belong to both the *intelligent healthcare monitoring* community and the *smart household* community, depending on the query and

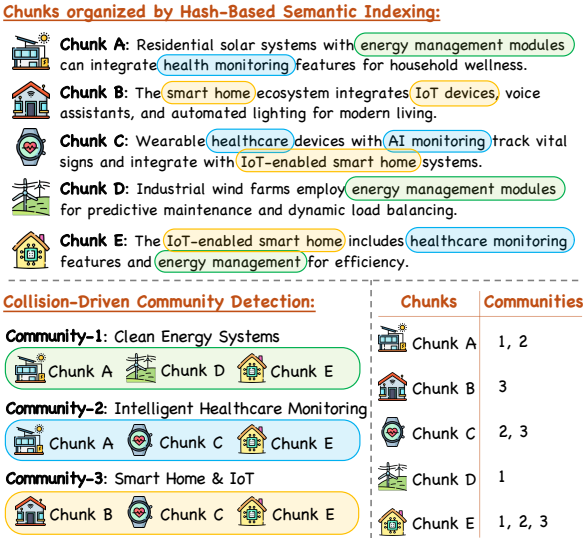

Figure 1: An illustration of *polysemous* nature. The same passage contains multiple entities can simultaneously belong to different communities when contextualized with other chunks.

reasoning focus. This flexibility is achieved by allowing chunks to participate in multiple communities based on their diverse hash collision patterns.

Candidate Community Generation. This step identifies initial grouping seeds from the LSH index structure. Each non-empty bucket in the L tables forms a raw candidate set $\mathcal{B}_{\text{raw}}^{(t,h)} = \{c_i \mid c_i \text{ hits bucket } h \text{ in table } t\}$. However, many such candidates are sparse or semantically ambiguous.

To select discriminative and meaningful candidate communities, we employ the **Bayesian Information Criterion (BIC)** as a filtering metric (Kuha, 2004). For a candidate community $\mathcal{B}$ containing n chunks with signature matrix $\mathbf{S}_{\mathcal{B}} \in \{0,1\}^{n \times k}$ (where k is the signature dimension and is set to be 12), we model the signature generation as a Bernoulli process. The BIC score is computed as:

$$\text{BIC}(\mathcal{B}) = -2 \ln \mathcal{L}(\mathbf{S}_{\mathcal{B}} \mid \hat{\boldsymbol{p}}) + k \ln n,$$

where $\hat{\boldsymbol{p}} = (\hat{p}_1, \ldots, \hat{p}_k)$ is the maximum likelihood estimate of the bit activation probabilities (i.e., $\hat{p}_l = \frac{1}{n} \sum_{c_i \in \mathcal{B}} \mathbf{s}_i[l]$) at position l within community $\mathcal{B}$, and $\mathcal{L}$ is the Bernoulli likelihood function. Candidates with lower BIC scores are retained, indicating higher internal coherence and statistical distinguishability from the distribution.

Iterative Community Consolidation. To form more coherent and stable semantic units, we consolidate the retained communities through an iterative bucket-wise merging procedure based on the actual chunk overlap between communities. Following the initial grouping induced by hash collisions, we compute the overlap between every pair of com-

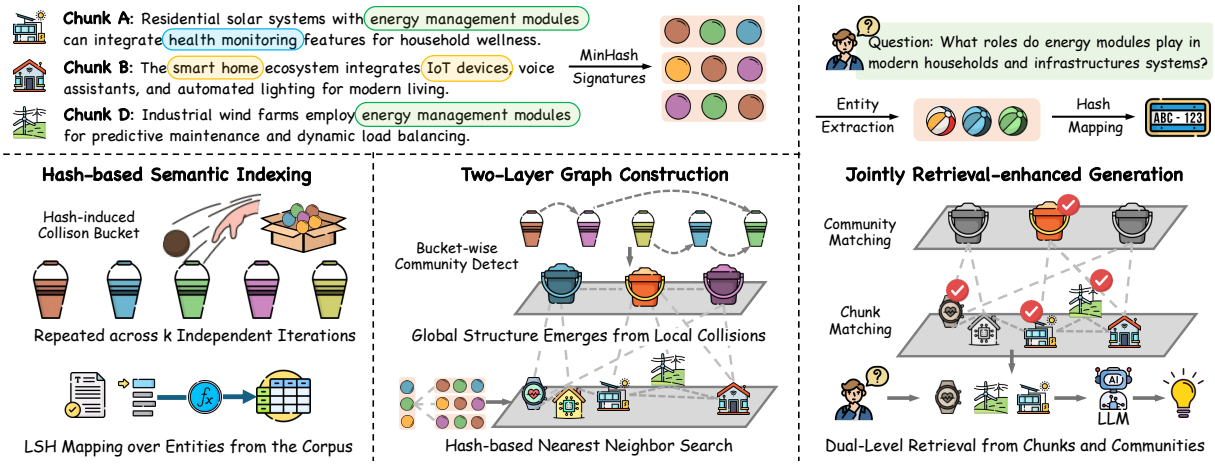

Figure 2: Schematic overview of the MeshRAG framework. Pairwise similarity between hash-based signatures induces fine-grained semantic connections at the chunk level while community structures are formed by iteratively merging chunks from candidate hash buckets. Given a query, the same hashing process produces its signature, followed by a dual-level retrieval from both chunks and communities for downstream question answering tasks.

munities as $\mathrm{O}(\mathcal{B}_u, \mathcal{B}_v) = \frac{|\mathcal{B}_u \cap \mathcal{B}_v|}{\min(|\mathcal{B}_u|, |\mathcal{B}_v|)}$ at each iteration step. All community pairs with overlap exceeding a threshold θ are then merged, forming a new community that is the union of their member chunks: $\mathcal{B}_{\text{new}} = \mathcal{B}_u \cup \mathcal{B}_v$. The prototype of the merged community is recomputed based on majority voting over the signatures of all chunks in $\mathcal{B}_{\text{new}}$. This process repeats iteratively until no community pairs satisfy the overlap condition, resulting in the final set of consolidated communities (See Appendix C.2). Importantly, because merging operates on community sets and a chunk can appear in multiple initial candidates, the final communities naturally remain overlapping, thus preserving the inherent polysemous nature of textual units.

3.3 Two-Layer Graph Construction

The figure 2 illustrates the construction workflow. Given the nearest neighbor relations discovered and the semantic communities identified in the previous stage, we construct a two-layer graph designed for efficient traversal and transparent reasoning. The vertex set consists of two distinct node types: fine-grained **Chunk Nodes**, each representing a textual chunk with its MinHash signature; and coarse-grained **Community Hyper-Nodes**, each abstracting a consolidated semantic community with its prototype signature. The edge set encodes two semantic relationships. **Chunk-Chunk Edges** connect pairs of chunks identified as LSH nearest neighbors, with each edge annotated by the exact set of common entities that caused their hash collision, $\text{entities}(u, v) = E_u \cap E_v$, providing explicit evidence for their semantic link. **Community-Member Edges** are directed links from each community into all its constituent chunk nodes, establishing the hierarchical containment. A key advantage is its inherent interpretability. At the chunk level, edge annotations directly reveal the shared entities that justified the connection. At the community level, each hyper-node is described with the most frequent entities or key phrases extracted from its member chunks, concisely summarizing the community's thematic identity. This design transforms $\mathcal{G}$ from a mere retrieval index into a readable map of the corpus's semantic landscape, allowing users to understand not only what information is retrieved but also why it is relevant.

3.4 Hash-based Retrieval & Generation

With the two-layer graph constructed, the retrieval phase efficiently locates relevant information for a user query by leveraging the hash-indexed structure to prune the search space dramatically. The query q is first processed identically to the corpus chunks: its entity list is extracted and encoded into a MinHash signature $\mathbf{s}_q$ using the same family of hash functions $\mathcal{H}$. This ensures mapping q into the same semantic space as the pre-indexed chunks. Our retrieval adopts a two-perspective search strategy to balance efficiency with comprehensiveness. First, a community-level pruning rapidly narrows the search scope by identifying the most promising thematic clusters via hash signature matching. Second, a fine-grained chunk-level search explores within and around these clusters through direct similarity comparison and graph expansion, ensuring both semantic relevance and contextual coverage.

Stage 1: Probabilistic Pruning via Bloom Filter. We first utilize a Bloom filter ($\mathcal{BF}$) pre-constructed from all community prototype signatures $\{\mathbf{p}_{\mathcal{C}_j}\}_{j=1}^{M}$. The filter $\mathcal{BF}$ uses η independent hash functions $\{g_1, \ldots, g_\eta\}$. To rapidly exclude irrelevant communities, we compute the η hash indices ($\eta = 4$) for the query signature: $\{g_1(\mathbf{s}_q), \ldots, g_\eta(\mathbf{s}_q)\}$. If *any* of the corresponding bits in $\mathcal{BF}$ is 0, the query signature is definitively not identical to any stored prototype, and the associated community can be immediately pruned. This step operates in $O(\eta)$ time and typically filters out a large majority of communities whose prototypes differ substantially from $\mathbf{s}_q$, providing a crucial first layer of efficiency.

Stage 2: Ranking via Hamming Similarity. The set of communities that pass the Bloom filter test, denoted $\mathcal{C}_{\text{candidate}}$, are those whose prototypes are likely similar enough to $\mathbf{s}_q$ to warrant closer inspection. For each community $\mathcal{C}_j \in \mathcal{C}_{\text{candidate}}$, we compute the exact Hamming similarity between its prototype and the query signature. The communities in $\mathcal{C}_{\text{candidate}}$ are then ranked in descending order of Sim_H, and the top-m communities (e.g., $m = 5$) are selected as the final relevant set $\mathcal{C}_{\text{relevant}}$ for the subsequent fine-grained chunk retrieval. This stage ensures precision by directly measuring bitwise agreement, but its cost is minimized as $|\mathcal{C}_{\text{candidate}}| \ll M$. This design leverages the advantage of hash-based representations: community prototypes and query signatures are already fixed-length bit strings, making them inherently compatible with the bitwise operations. Thus the filter serves as a lightweight discriminator that directs the query only to the most relevant regions.

To ensure coverage beyond community boundaries, we additionally retrieve globally top-20 the most similar chunks to $\mathbf{s}_q$ using the same mechanism mentioned in the previous section. These chunks serve as anchors, from which we perform a k-hop neighborhood expansion on the chunk-level graph to collect locally connected context. The union of chunks from the selected communities and the expanded neighborhoods forms the candidate pool $\mathcal{P}_q$. All chunks in $\mathcal{P}_q$ are then re-ranked and the top-5 items are selected as the retrieved content to construct the prompt for answering.

4 Experiments

We conduct comprehensive experiments on three public open-domain datasets, along with a textbook QA benchmark. Our study aims to address four key questions: **Q1:** Can our method retrieve more relevant information to effectively answer complex questions and outperform existing state-of-the-art baselines? **Q2:** Does our hash-based graph construction achieve high efficiency in both fast graph building and retrieval scope reduction during inference? **Q3:** Is the constructed graph structure intuitive, providing clear connection logics that enhance the interpretability? **Q4:** How does our method perform under different hyperparameters, and what is its sensitivity to these configurations?

Baselines. We compare our framework against a diverse set of baselines from several paradigms for a comprehensive evaluation. The compared methods are grouped into three categories: (1) direct language model approaches, which employ pre-trained models like GPT or Llama for zero-shot question answering without any retrieval; (2) retrieval-augmented LMs, which enhance language models by incorporating similarity-based retrieval mechanisms; and (3) graph-augmented generation methods, which utilize structured knowledge through common knowledge graphs or hierarchical clustering strategies. Detailed descriptions of these baselines are provided in Appendix B.1.

Datasets. Our evaluation employs three distinct datasets, each consisting of 1,000 questions from its specific domain, along with the GraphRAG benchmark dataset. More details are given in Appendix B.2. The corpus for each dataset comprises approximately ten thousand independent text chunks, providing a rich foundation of diverse contexts and query types. These datasets are designed to represent a spectrum of reasoning complexity, ranging from fundamental scientific terminology to advanced humanities subjects, thereby offering a comprehensive resource for assessing our method's proficiency in knowledge-intensive tasks. To ensure experimental consistency and comparability, we follow the same questions and corpora established in prior work (Gutiérrez et al., 2024).

Evaluation Metrics. For a fair and comprehensive comparison, we employ two complementary evaluation metrics. First, we use String-Match Accuracy, a strict lexical metric that determines correctness by checking if the key phrase or short answer from the ground truth is contained within the model's prediction. To complement this and account for semantic equivalence, we additionally employ language models as a judge. This metric captures the correctness of the prediction's meaning through checking questions, predictions and ground truths.

Table 1: Performance comparison among state-of-the-art baselines and MeshRAG on three benchmark datasets in terms of both String-Match and GPT-evaluation Accuracy. All baselines are divided into the following three groups.

Model	HotpotQA		2Wiki		Musique	
	Match-Acc.	GPT-Acc.	Match-Acc.	GPT-Acc.	Match-Acc.	GPT-Acc.
Direct Zero-shot LM Inference						
Llama3 (8b) (Touvron et al., 2023)	10.8	11.6	12.4	9.2	5.9	4.8
Llama3 (13b)	12.6	10.7	13.1	10.6	5.7	4.4
GPT-3.5	36.3	39.8	37.3	31.6	16.2	17.9
GPT-4o-mini (Achiam et al., 2023)	38.4	39.1	38.0	33.9	17.5	18.3
Retrieval-augmented Variants						
Retrieval (Top-1)	42.4	46.6	44.8	42.3	19.2	21.5
Retrieval (Top-3)	44.2	48.1	45.2	43.5	21.6	24.2
Retrieval (Top-5)	45.1	49.9	45.7	42.4	21.9	25.8
Graph-enhanced Generation Methods						
KGP (Wang et al., 2024)	54.4	57.1	48.5	40.7	24.3	27.3
G-retriever (He et al., 2024)	40.3	41.9	33.7	25.7	14.1	15.6
LightRAG (Guo et al., 2024)	50.8	52.7	49.3	43.3	25.3	27.7
E^2Graphrag (Zhao et al., 2025)	61.0	63.9	54.3	49.8	23.8	26.2
HippoRAG (Gutiérrez et al., 2024)	57.7	59.6	59.7	53.2	29.0	28.8
RAPTOR (Sarthi et al., 2024)	54.1	55.3	50.6	43.9	26.2	29.7
HippoRAG2 (Gutiérrez et al., 2025)	62.9	64.3	62.7	55.0	33.2	35.3
Youtu-GraphRAG (Dong et al., 2025)	63.5	65.3	61.1	55.4	31.8	34.0
GFM-RAG (Luo et al., 2025)	63.7	65.6	63.6	56.8	32.3	34.6
MeshRAG (Ours)	64.2	66.2	65.6	60.9	35.5	37.0

Implementation Details. The core hyperparameters of our method, the number of communities m and the search hop range k, are determined via a grid search to identify the optimal configuration for each dataset. We uniformly employ GPT-4o-mini and the number of retrieved items is 5. All baseline implementations strictly adhere to the setups and configurations described in their original publications to ensure a fair and reproducible comparison.

4.1 Main Results

The comparison of generation accuracy across three datasets between our model and other baselines is presented in Table 1. As outlined above, we categorize all compared methods into three groups.

From Zero-shot to Retrieval-Augmented. Results from the first two groups reveal a clear improvement. Direct zero-shot LLM inference establishes a baseline capability, confirming that language models can address parts of questions through their own knowledge. A significant and consistent performance gain is observed when adding retrieval modules, verifying the effectiveness of providing external contextual information.

The Graph Enhancement Advantage. The most significant trend emerges within the third group of graph-enhanced methods. These approaches consistently outperform simple retrieval-augmented models, indicating that structuring knowledge into explicit relational graphs provides a substantial benefit beyond semantic retrieval. However, the performances highly vary across this group, demonstrating that the specific paradigm for graph construction and traversal is crucical to the outcome. This variance highlights that capturing meaningful entity relationships is as important as identifying relevant entities. MeshRAG achieves the highest accuracy. This success is attributed to its informative graph construction, which effectively integrates information from both a broad community-level perspective and a fine-grained chunk-level perspective.

4.2 GraphRAG Benchmark

To further validate the generalizability of our approach across different reasoning paradigms, we also conducted experiments on the GraphBench dataset, a comprehensive collection derived from textbook knowledge that encompasses diverse question types including fill-in-the-blank, multiple-choice questions, and so on (see Appendix B.3 for details). As shown in Table 4, this result demonstrates not all graph-based RAG techniques consistently yield gains. While methods like RAPTOR demonstrate strong overall improvements, others such as DALK and G-Retriever underperform compared to the baseline GPT-4o-mini. This suggests

that improperly integrated or overly complex graph structures can potentially introduce noise or mislead the reasoning chain, thereby confusing the LLM instead of providing beneficial contextual guidance. The superior performance of our method illustrates the importance of a well-designed retrieval and integration mechanism that reliably provides relevant content from the constructed graph.

4.3 Efficiency Analysis

The core motivation for using hashing is efficiency, and we perform a detailed comparison of both time and token consumption for graph construction. As shown in Figure 3, our method requires the least running time. Hashing is fast because it performs only a small, fixed number of simple arithmetic and bitwise operations per input token or chunk; it does not need to perform iterative search, optimization, or language understanding. In contrast, most baselines consume a large number of tokens because they rely on expensive preprocessing steps such as chunk summarization or triple extraction. Importantly, our approach **constructs the graph without any dependence on language models**, avoiding costly token-based queries while still producing a compact and informative graph. The average retrieval time stays under 0.8 seconds as the graph structure helps quickly focus the search on the most promising parts. As shown in our ablation study, the best setup only needs to check around 15% of all possible items. Moreover, this implementation can be further accelerated through multiprocessing.

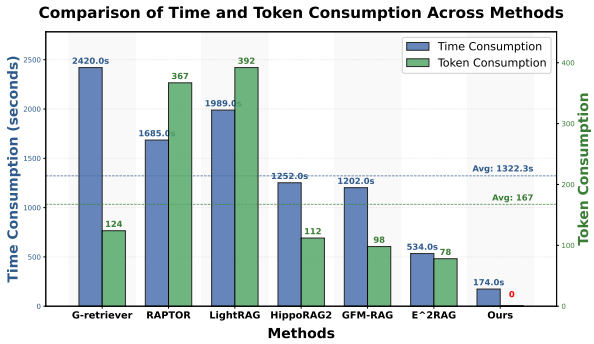

Figure 3: A histogram comparing the time and token consumptions across different Graph-based variants.

4.4 Interpretability of Hash-Based Graph

In contrast to models that rely on abstract embedding spaces, the structure of our hash-based graph is inherently interpretable. Traditional clustering methods applied to dense embeddings often result in clusters whose semantic themes are not directly apparent. The connection between chunks does not directly reveal the underlying logic. However, our method builds the graph directly upon hash collisions, where each collision serves as an explicit reference for semantic relatedness. The communities emerge from the frequent co-occurrence of specific entities across multiple chunks. Consequently, the importance of an entity within a community is directly reflected by its frequency of appearance in the collision events that defined the community. As shown in Figure 4, the t-SNE plot demonstrates clear clustering of textual chunks within sampled groups. To further illustrate the semantic coherence of our graph, Figure 5 showcases a sample of concrete community-level entities alongside representative chunk-level items. The visualization enables an intuitive explanation of the distinct topic characterizing each community, thereby verifying the interpretability of the constructed graph (A detailed analysis report is given in Appendix C.2).

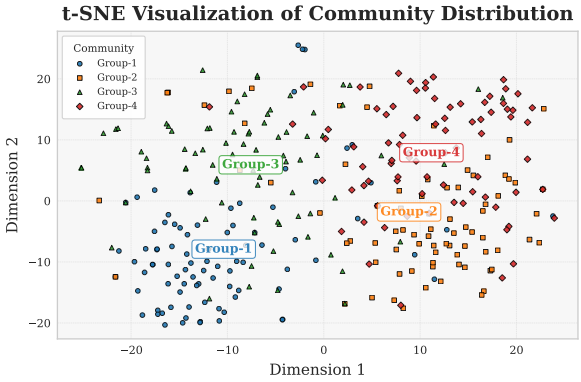

Figure 4: A t-SNE visualization demonstrates the distributions of textual chunks across different communities.

4.5 Ablation Study & Hyperparameter

We conduct an ablation study to examine the impact of different search strategies, as shown in Table 2. Our strategy first identifies the top-20 most relevant chunks as anchor nodes based on hash value similarity. Then, we expand the search scope by varying the number of hops (k) and the community integration. The *Ratio* metric reflects the proportion of total items examined under each setting. The results indicate that alternative clustering methods fail to achieve comparable performance, mainly because they enforce each chunk into a single cluster. This constraint prevents a chunk from participating in multiple contexts and often leads to highly imbalanced cluster sizes, making long-tail categories difficult to identify. Besides, simply expanding the search scope does not guarantee proportional

gains. This confirms that our hash-based method successfully identifies the most relevant content, effectively narrowing the search space. Therefore, we select the 1-hop and 5-community settings as it best balances retrieval speed and accuracy.

Method	# Community	Ratio (%)	Acc.
Dense Search	–	–	25.8
DBSCAN	10	12.5	28.9
K-means Clustering	10	14.8	29.5
MeshRAG (1-Hop)	–	3.6	22.3
	5	15.3	37.0
	10	27.1	37.4
MeshRAG (2-Hop)	–	33.8	36.8
	5	43.9	37.2
	10	57.2	38.3

Table 2: Ablation Study on Clustering Strategies and MeshRAG Variants. The table analyzes how different experimental scenarios influence generation accuracy.

The number of LSH tables (L) and the length of signatures (k) are the two primary factors controlling the the similarity estimation. We conduct a hyperparameter study by varying one parameter and comparing the results against our default setting ($L = 8, k = 12$). The analysis reveals that **our method is remarkably robust to the value selection**. As shown in Table 3, the overlapping ratio of the top-10 and top-20 items remains consistently high (> 0.90) across all configurations. More importantly, even when strictly considering the exact ranking order into account, the Spearman (ρ) and Kendall (τ) rank correlation coefficients still stay high. This study indicates the robustness and consistency of our hash-based mechanism.

Setting.	Top-10	Top-20	ρ	τ
Varying LSH tables L (with $k = 12$)				
$L = 10$	0.98	0.96	0.97	0.92
$L = 12$	0.97	0.93	0.87	0.81
$L = 14$	0.98	0.94	0.86	0.78
$L = 16$	0.96	0.93	0.83	0.77
Varying signature length k (with $L = 8$)				
$k = 8$	0.95	0.92	0.89	0.80
$k = 10$	0.96	0.96	0.94	0.85
$k = 14$	0.97	0.96	0.96	0.87
$k = 16$	0.97	0.95	0.91	0.83

Table 3: Robustness analysis of the number of LSH Tables (L) and Signature Length (k) against our setting.

5 Related Work

Retrieval-augmented generation has established itself as a powerful framework for knowledge-intensive tasks (Dong et al., 2024b), particularly in open-domain and multi-hop question answering. The core paradigm involves retrieving evidence from an external knowledge source, then conditioning a language model on this context to generate answers (Lewis et al., 2020; Dong et al., 2024a). Initial approaches like REALM (Zhu et al., 2024) and DPR (Siriwardhana et al., 2021) pioneered dense passage retrieval, learning continuous representations that enable efficient similarity search over large corpora. Recent innovations shift focus toward structured retrieval, where passages are organized not just as independent items but as interconnected units (Cuconasu et al., 2024; Han et al., 2025; Xiao et al., 2026a). Several methods adopt graph-based representations to model semantic or logical relationships between documents (Dong et al., 2024c; Fan et al., 2024; Zhou et al., 2025a; Yang et al., 2026; Dong et al., 2026; Xiao et al., 2026b). For example, HippoRAG incorporates external knowledge graphs to enrich retrieval with entity-aware context (Gutiérrez et al., 2024), while GFM-RAG builds and refines knowledge graphs from text to support more coherent multi-hop reasoning (Luo et al., 2025). LogicRAG (Chen et al., 2025) models reasoning trajectories within diverse graph types. On the contrast, RAPTOR organizes the knowledge by applying hierarchical clustering to induce a multi-level index, allowing progressive refinement of retrieval scope (Sarthi et al., 2024).

6 Conclusion

This work addresses two key limitations of RAG: the lack of interpretability in neural embedding methods and the computational cost of knowledge graph construction. To this end, We propose a paradigm shift toward a lightweight, hash-driven approach, demonstrating that semantic similarity can be effectively and transparently approximated through locality-sensitive hashing functions. Our framework autonomously builds a two-layer semantic graph by establishing both fine-grained chunk connections and overlapping semantic communities. The entire process operates without GPU-intensive computations or LLM token consumption. This design achieves not only a substantial speedup but also provides inherent interpretability, as relevant textual entities can be directly traced back to the specific hash collisions. Extensive experiments have verified the effectiveness and efficiency of our framework. We hope this core idea can offer insights to scalable and interpretable RAG systems.

Limitations

Our framework currently focuses only on textual chunks and does not explicitly model multimodal or highly-structured knowledge. We plan to extend it to include image data in our future work to achieve richer retrieval. Second, unlike mature vector search systems such as FAISS that perform retrieval in a single, optimized operation, our current graph search remains iterative. We can implement parallel retrieval mechanisms and optimize multiprocessing strategies to further improve efficiency. Besides, in some cases, different paragraphs may adopt diverse expressions to convey similar meanings, making it difficult for hash-based estimation to directly capture such synonymy. To mitigate this issue, synonym normalization can be applied as a preprocessing step, where synonym dictionaries are used to unify semantically equivalent expressions before hashing. This helps improve the consistency and reduces the impact of lexical variation.

Ethical Considerations

This study was conducted in full compliance with the ACL Ethics Policy. All datasets utilized in this research are publicly available and open-source. These datasets contain no personal information or sensitive content. We have considered the broader society impact and will continue to follow ethical standards in our future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. A survey on rag with llms. *Procedia computer science*, 246:3781–3790.

Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. 2025. You don't need pre-built graphs for rag: Retrieval augmented generation with adaptive reasoning structures. *arXiv preprint arXiv:2508.06105*.

Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. 2026. You don't need pre-built graphs for rag: Retrieval augmented generation with adaptive reasoning structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30270–30278.

Flavio Chierichetti and Ravi Kumar. 2015. Lsh-preserving functions and their applications. *Journal of the ACM (JACM)*, 62(5):1–25.

Tobias Christiani and Rasmus Pagh. 2017. Set similarity search beyond minhash. In *Proceedings of the 49th annual ACM SIGACT symposium on theory of computing*, pages 1094–1107.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.

Biplob Debnath, Sudipta Sengupta, Jin Li, David J Lilja, and David HC Du. 2011. Bloomflash: Bloom filter on flash-based storage. In *2011 31st International Conference on Distributed Computing Systems*, pages 635–644. IEEE.

Junnan Dong, Siyu An, Yifei Yu, Qian-Wen Zhang, Linhao Luo, Xiao Huang, Yunsheng Wu, Di Yin, and Xing Sun. 2025. Youtu-graphrag: Vertically unified agents for graph retrieval-augmented complex reasoning. *arXiv preprint arXiv:2508.19855*.

Junnan Dong, Qinggang Zhang, Chuang Zhou, Hao Chen, Daochen Zha, and Xiao Huang. 2024a. Cost-efficient knowledge-based question answering with large language models. *Advances in Neural Information Processing Systems*, 37:115261–115281.

Junnan Dong, Qinggang Zhang, Huachi Zhou, Daochen Zha, Pai Zheng, and Xiao Huang. 2024b. Modality-aware integration with large language models for knowledge-based visual question answering. In *ACL*, pages 2417–2429. ACL.

Junnan Dong, Chuang Zhou, Zheng Yuan, Yifei Yu, Siyu An, Di Yin, Xing Sun, and Feiyue Huang. 2026. Deep tabular research via continual experience-driven execution. *arXiv preprint arXiv:2603.09151*.

Yuxin Dong, Shuo Wang, Hongye Zheng, Jiajing Chen, Zhenhong Zhang, and Chihang Wang. 2024c. Advanced rag models with graph structures: Optimizing complex knowledge reasoning and text generation. In *2024 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*, pages 626–630. IEEE.

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

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on*

knowledge discovery and data mining, pages 6491–6501.

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

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

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

Haoyu Han, Li Ma, Harry Shomer, Yu Wang, Yongjia Lei, Kai Guo, Zhigang Hua, Bo Long, Hui Liu, Charu C Aggarwal, and 1 others. 2025. Rag vs. graphrag: A systematic evaluation and key insights. *arXiv preprint arXiv:2502.11371*.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, and 1 others. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Omid Jafari, Preeti Maurya, Parth Nagarkar, Khandker Mushfiqul Islam, and Chidambaram Crushev. 2021. A survey on locality sensitive hashing algorithms and their applications. *arXiv preprint arXiv:2102.08942*.

Jouni Kuha. 2004. Aic and bic: Comparisons of assumptions and performance. *Sociological methods & research*, 33(2):188–229.

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. *Advances in neural information processing systems*, 33:9459–9474.

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*.

Luigi D Notarangelo, Min-Sung Kim, Jolan E Walter, and Yu Nee Lee. 2016. Human rag mutations: biochemistry and clinical implications. *Nature Reviews Immunology*, 16(4):234–246.

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

Anshumali Shrivastava and Ping Li. 2014. In defense of minhash over simhash. In *Artificial intelligence and statistics*, pages 886–894. PMLR.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, and Suranga Nanayakkara. 2021. Fine-tune the entire rag architecture (including dpr retriever) for question-answering. *arXiv preprint arXiv:2106.11517*.

Sasu Tarkoma, Christian Esteve Rothenberg, and Eemil Lagerspetz. 2011. Theory and practice of bloom filters for distributed systems. *IEEE Communications Surveys & Tutorials*, 14(1):131–155.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. 9835 musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19206–19214.

Xian Wu, Moses Charikar, and Vishnu Natchu. 2018. Local density estimation in high dimensions. In *International Conference on Machine Learning*, pages 5296–5305. PMLR.

Yilin Xiao, Jin Chen, Qinggang Zhang, Yujing Zhang, Chuang Zhou, Longhao Yang, Lingfei Ren, Xin Yang, and Xiao Huang. 2026a. Logicpoison: Logical attacks on graph retrieval-augmented generation. *arXiv preprint arXiv:2604.02954*.

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

Yilin Xiao, Chuang Zhou, Qinggang Zhang, Bo Li, Qing Li, and Xiao Huang. 2026b. Reliable reasoning path: Distilling effective guidance for llm reasoning with knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*.

Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, and 1 others. 2026. Graph-based agent memory: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2602.05665*.

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. *arXiv preprint arXiv:1809.09600*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Hao Chen, Yilin Xiao, Chuang Zhou, Junnan Dong, and 1 others. 2025. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

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*.

Chuang Zhou, Junnan Dong, Xiao Huang, Zirui Liu, Kaixiong Zhou, and Zhaozhuo Xu. 2024. Quest: Efficient extreme multi-label text classification with large language models on commodity hardware. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3929–3940.

Chuang Zhou, Jiahe Du, Huachi Zhou, Hao Chen, Feiran Huang, and Xiao Huang. 2025a. Text-attributed graph learning with coupled augmentations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10865–10876.

Chuang Zhou, Zhu Wang, Shengyuan Chen, Jiahe Du, Qiyuan Zheng, Zhaozhuo Xu, and Xiao Huang. 2025b. Taming language models for text-attributed graph learning with decoupled aggregation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3463–3474.

Yinghao Zhu, Changyu Ren, Shiyun Xie, Shukai Liu, Hangyuan Ji, Zixiang Wang, Tao Sun, Long He, Zhoujun Li, Xi Zhu, and 1 others. 2024. Realm: Rag-driven enhancement of multimodal electronic health records analysis via large language models. *arXiv preprint arXiv:2402.07016*.

A Appendix

A.1 Locality-Sensitive Hashing Property

An LSH family $\mathcal{F}$ is formally defined for a similarity metric $J(\cdot,\cdot)$ with a threshold τ and approximation factors $p_1 > p_2$. For any two chunks c_i, c_j:

$$\begin{cases} \text{If } J(c_i,c_j) \geq \tau, & P_{h\in\mathcal{F}}[h(c_i)=h(c_j)] \geq p_1, \\ \text{If } J(c_i,c_j) \leq \tau-\epsilon, & P_{h\in\mathcal{F}}[h(c_i)=h(c_j)] \leq p_2, \end{cases}$$

where $\epsilon > 0$ is a small constant, and $p_1 > p_2$. In our framework, each hash function h_l in the MinHash family $\mathcal{H}$ inherently satisfies this property for the Jaccard similarity metric (Jafari et al., 2021).

A.2 Theoretical Foundation of MinHash

The core property of MinHash (Shrivastava and Li, 2014), which bridges hash collisions to semantic similarity, is given by the following theorem:

Theorem 1 (MinHash Property)**.** *For any two chunks c_i and c_j, and for any hash function $h_l \in \mathcal{H}$,*

$$P\left[m_l(c_i) = m_l(c_j)\right] = J(c_i,c_j),$$

where the probability is taken over the choice of h_l.

Let $U = E_i \cup E_j$. Under a random hash function, all permutations of U are equally likely (Christiani and Pagh, 2017; Chierichetti and Kumar, 2015). The event $m_l(c_i) = m_l(c_j)$ occurs if and only if the smallest element in U belongs to $E_i \cap E_j$. Since all elements are equally likely to be the smallest,

$$P\left[m_l(c_i) = m_l(c_j)\right] = \frac{|E_i \cap E_j|}{|U|} = J(c_i,c_j).$$

This theorem implies that a single MinHash component provides an unbiased estimate of Jaccard similarity (Zhou et al., 2024). By concatenating k such components, the signature $\mathbf{s}_i$ provides a high-probability, fixed-dimensional representation of the chunk's semantic content. The empirical similarity between two chunks can be efficiently estimated by the Hamming similarity of their signatures:

$$\hat{J}(c_i,c_j) = \frac{1}{k}\sum_{l=1}^{k} \mathbb{I}\left[m_l(c_i) = m_l(c_j)\right],$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Algorithm 1 Hash Indexing & K-NN Discovery

Require: Corpus $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$.
Ensure: MinHash signatures $\{\mathbf{s}_i\}_{i=1}^N$ and nearest neighbor set $\mathcal{N}$.

Entity Extraction and MinHash Signing
for each chunk $c_i \in \mathcal{C}$ **do**
  Extract entity set E_i.
  Compute signature $\mathbf{s}_i$ of length k:
    $\mathbf{s}_i[l] = \min_{e\in E_i} h_l(e)$ for $l = 1, \ldots, k$.
end for
LSH Table for Approximate Search
Initialize L empty hash tables, each with b bands and r rows ($k = b \times r$).
for each chunk $c_i \in \mathcal{C}$ **do**
  for each table $t = 1$ to L **do**
    Compute band keys for $\mathbf{s}_i$ in table t.
    Insert c_i into corresponding buckets.
  end for
end for
K-Nearest Neighbor Discovery
Let $\mathcal{N} \leftarrow \emptyset$ {Set of neighbor pairs}
for each chunk $c_i \in \mathcal{C}$ **do**
  $\text{Cand}_i \leftarrow \emptyset$
  for each table $t = 1$ to L **do**
    Locate bucket for c_i in table t.
    $\text{Cand}_i \leftarrow \text{Cand}_i \cup$ {in the same bucket}
  end for
  Remove c_i from Candidates_i.
  Estimate Similarity $\hat{J}(c_i,c_j)$ for each c_j:
    $\hat{J}(c_i,c_j) = \frac{1}{k}\sum_{l=1}^{k} \mathbb{I}[\mathbf{s}_i[l] = \mathbf{s}_j[l]]$.
  Select top-κ chunks with highest $\hat{J}$.
  for each selected neighbor c_j **do**
    $\mathcal{N} \leftarrow \mathcal{N} \cup \{(c_i, c_j, \hat{J}(c_i,c_j))\}$
  end for
end for
return $\{\mathbf{s}_i\}_{i=1}^N, \mathcal{N}$

A.3 Explanation of LSH-Based Counting

Given a high-dimensional dataset, we aim to estimate the number of data points within a specified angular distance A from a query vector q, denoted $|A_q|$. This problem arises naturally in similarity search and nearest neighbor applications.

Naive random sampling from the entire dataset of size n requires approximately $O\big(\frac{n}{|A_q|\epsilon^2}\ln\frac{1}{\delta}\big)$ samples to achieve a $(1\pm\epsilon)$ approximation with probability $1-\delta$. The factor $\frac{n}{|A_q|}$ represents the inverse prevalence of target points; when $|A_q| \ll n$, sampling becomes extremely expensive.

Locality-Sensitive Hashing improves efficiency by first filtering the dataset. Using hash functions that map similar points to nearby buckets, the query q is used to retrieve a candidate pool $C_q(\mathcal{I})$ consisting of points whose hash values lie within Hamming distance $\mathcal{I}$ of q's hash. This pool is typically much smaller than the full dataset and contains a higher concentration of target points. The LSH estimator achieves sample complexity

$$O\Big(\frac{\mathbb{E}(C_q(\mathcal{I}))}{\epsilon^2|A_q|\cdot\min_{x\in A_q}p(x)}\log\frac{1}{\delta}\Big),$$

where $\min_{x\in A_q}p(x)$ is the minimum probability that a target point falls within the Hamming radius $\mathcal{I}$, and $\mathbb{E}(C_q(\mathcal{I}))$ is the expected pool size. The ratio $\frac{\mathbb{E}(C_q(\mathcal{I}))}{|A_q|\cdot\min_{x\in A_q}p(x)}$ replaces the naive factor $\frac{n}{|A_q|}$ and is typically much smaller because LSH enriches the candidate pool with relevant points while excluding most irrelevant ones (Wu et al., 2018).

In summary, LSH accelerates counting by (1) constructing a small, target-rich candidate pool via locality-sensitive hashing, and (2) performing random sampling within this pool rather than across the entire dataset. This approach is especially advantageous when the set of interest is sparse within the overall population (Zhou et al., 2025b).

A.4 Probabilistic Pruning via Bloom Filter

We employ a Bloom filter $\mathcal{BF}$ for constant-time rejection of irrelevant communities (Tarkoma et al., 2011; Debnath et al., 2011). The filter is constructed from all community prototypes $\mathbf{P} = \{\mathbf{p}_1,\ldots,\mathbf{p}_M\}$ during indexing. Formally, $\mathcal{BF}$ is a bit array of length m. For each prototype $\mathbf{p}_j$, we apply η independent hash functions $\{g_1,\ldots,g_\eta\}$, each mapping a binary string to an integer in $[0, m-1]$. The corresponding bits are set to 1:

$$\mathcal{BF}[g_k(\mathbf{p}_j)] \leftarrow 1 \quad \text{for } k = 1,\ldots,\eta.$$

During retrieval, for a query signature $\mathbf{s}_q$, we check all η positions:

$$\text{If } \bigwedge_{k=1}^{\eta}\mathcal{BF}[g_k(\mathbf{s}_q)] = 1, \text{ then } \mathbf{s}_q \text{ is a } \textit{possible match}.$$

If any bit is 0, $\mathbf{s}_q$ is *definitely not* in $\mathbf{P}$, and all communities are pruned for this query. The chance of a false positive (erroneously passing an unrelated signature) after inserting M prototypes is:

$$P_{\text{FP}} \approx \left(1 - e^{-\eta M/m}\right)^{\eta}.$$

By tuning m and η, we control this probability (typically $< 1\%$), ensuring that the subsequent exact comparison stage is invoked only for a small candidate set $\mathcal{C}_{\text{cand}}$, where $|\mathcal{C}_{\text{cand}}| \ll M$.

B Experimental Details

B.1 Baselines Descriptions

Below is a detailed description of the baseline methods used for comparison in our experiments.

- **KGP**: Constructs a knowledge graph from multiple text passages using LLMs. During retrieval, it employs an LLM-based agent to perform graph traversal, progressively gathering supporting passages for the answer.
- **G-Retriever**: Integrates graph neural networks with LLMs by framing subgraph retrieval as a Prize-Collecting Steiner Tree problem. This approach aims to improve conversational QA on textual graphs while reducing hallucinations and improving scalability.
- **E^{2}GraphRAG**: Extracts entities with spaCy and uses LLMs to build a hierarchical tree of summarized groups during indexing for retrieval.
- **LightRAG**: Adopts a two-tier indexing framework that combines fine-grained entity-relation graphs with coarse-grained thematic structures, aiming to balance granularity and efficiency.
- **RAPTOR**: Builds a hierarchical tree structure over a corpus by recursively applying clustering and abstractive summarization, creating multi-granular semantic representations for retrieval.
- **HippoRAG**: A training-free retrieval method that leverages Personalized PageRank with query concepts as seeds to perform single-step or multi-hop retrieval across a document graph.
- **Youtu-GraphRAG**: Adopts a vertically unified agentic paradigm to construct and retrieve knowledge using a seed graph schema to guide entity, relation, and attribute extraction.
- **HippoRAG2**: An enhanced version of HippoRAG with improved paragraph integration and contextualization mechanisms, optimizing seed selection and PageRank parameters for better retrieval accuracy.
- **GFM-RAG**: Implements a GraphRAG pipeline by constructing KG from documents and utilizing a trained graph-enhanced retriever.

B.2 Dataset Description

We run experiments using three public datasets: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and the Musique (Trivedi et al., 2022) dataset. For each dataset, we evaluate on a subset of 1,000 questions. These benchmarks are designed for multi-hop question answering and span diverse domains such as history, science, politics, geography, and popular culture. HotpotQA emphasizes reasoning over multiple supporting documents with explicit evidence chains, while 2WikiMultiHopQA focuses on cross-document reasoning across Wikipedia articles. Musique further introduces compositional and context-dependent questions that require integrating information from multiple passages while minimizing shortcut cues. Together, these datasets provide a comprehensive evaluation of retrieval robustness and reasoning capability across varied domains and question types.

B.3 GraphRAG-Bench Description

We evaluate our model on GraphRAG-Bench, a comprehensive benchmark derived from computer science textbooks (Xiao et al., 2025). Its questions span a wide range of subfields, from fundamental machine learning to advanced deep learning areas. The benchmark thoroughly assesses reasoning capabilities through five distinct question types, with detailed descriptions provided in Table 5.

C Case Study

C.1 LSH Nearest Neighbor Search

Chunk Analysis: This case study demonstrates how LSH efficiently identifies semantically similar chunks. Chunk 1 (about Lionel Messi) is correctly matched with other football-related chunks (2, 6, 10) through shared entities like "Messi", "Barcelona", and "Real Madrid", while also retrieving historically relevant chunks (7714, 7869) through entities like "Barcelona" and "Spanish". The high similarity scores with only 8 LSH tables validate our hash-based retrieval effectiveness. Below is a detailed report showing original text and its top-k nearest neighbors based on hash collisions.

C.2 LSH Community Detection

Community Analysis: The community detection analysis reveals several key insights. First, our LSH-based approach successfully groups 72% of all chunks into semantically coherent communities with balanced sizes (average 488 chunks). Thanks to the traceable hash-collision property, figure 5 presents sampled high-frequent entities appearing in the community and chunk levels. **The case in the textbox below** demonstrates our effectiveness: chunks about diverse topics with common history and geography are clustered together through shared core entities like "European", "Germany", and "February". The hash prototype serves as a compact signature representing the community's semantic fingerprint. This shows that our method captures cross-domain semantic relationships rather than simple topical clustering.

Figure 5: A Sunburst diagram illustrating the interpretability of our hash-based graph construction. The inner ring represents community-level entity signatures, while the outer ring visualizes chunk-level common entities within each corresponding community.

Together with the t-SNE visualization shown above, these results further demonstrate the effectiveness of our hash-based community construction. The plot intuitively reveals that chunks assigned to the same community form compact and well-separated clusters, despite the communities being induced without explicit embedding-based optimization. This indicates that the hash-induced structure aligns well with underlying semantic geometry, while avoiding the rigidity of conventional clustering. Moreover, the smooth transitions observed between neighboring communities suggest that our approach preserves semantic continuity across domains, enabling chunks to participate in multiple related contexts. Overall, the visualization confirms that hash-based community induction provides a robust and flexible structural organization, supporting both clustering and interpretability.

Table 4: Comparison of performance on reasoning benchmark dataset.

Method	Accuracy					
	Fill-in-blank	Multi-choice	Multi-select	True-or-false	Open-ended	Average
GPT-4o-mini	74.29	77.05	74.54	75.95	52.23	70.81
DALK	70.40	78.34	71.62	77.22	51.49	69.81↓
G-Retriever	70.95	77.42	71.62	78.80	52.04	70.16↓
LightRAG	65.24	78.80	73.42	82.59	53.16	70.64↓
ToG	70.48	78.80	**78.38**	79.75	54.28	72.34↑
KGP	74.29	79.26	74.77	82.28	51.49	72.42↑
GFM-RAG	72.38	<u>80.65</u>	72.07	<u>82.59</u>	52.79	72.10↑
HippoRAG	70.48	80.18	74.32	81.65	<u>56.13</u>	72.56↑
RAPTOR	<u>76.67</u>	<u>80.65</u>	<u>77.48</u>	82.28	54.83	<u>74.38</u>↑
Ours	**78.95**	**81.12**	76.17	**82.96**	**60.70**	**75.98**↑

Table 5: Description of question types in the GraphRAG-Bench dataset.

Question Type	**Description**
Fill-in-the-Blank (FB)	Evaluates the model's ability to complete context-dependent statements with accurate entities or phrases. This task tests precise entity grounding and the understanding of local semantic dependencies within a graph.
Multiple-Choice (MC)	Presents a question alongside four candidate options, including semantically plausible distractors. It assesses the model's discriminative reasoning capability, requiring it to integrate entity attributes and relational edges to identify the single correct answer.
Multi-Select (MS)	Requires the selection of 2–4 correct answers from four candidates, often involving reasoning over interconnected concepts. This format tests the ability to handle complex query semantics by aggregating multi-hop evidence and resolving overlaps among distractors.
True-or-False (TF)	Involves determining the veracity of a given statement. This task measures factual accuracy and logical inference capability by verifying claims against structured knowledge.
Open-Ended (OE)	Allows for free-form, comprehensive answers, evaluating the model's capacity to synthesize and articulate knowledge across multiple subfields into coherent, long-form explanations.

Enhanced Chunk LSH Nearest Neighbor Search (Based on 8 LSH Tables)

Chunk 1: [After a year at Barcelona's youth academy, La Masia, Messi was finally enrolled in the Royal Spanish Football Federation (RFEF) in February 2002. . .]
Entity List: Lionel Messi, Barcelona, La Masia, Royal Spanish Football Federation, February, Cesc Fàbregas, Gerard Piqué, Spanish, Catalan, Arsenal, England
Top-5 Similar Chunks:

1. **[Similarity: 27/8 (337.5%)] Chunk 7714:** [Corona de Aragón (Spanish) Aragoiko koroa (Basque) Composite monarchy, confederation of kingdoms, or individual polities ruled by. . .]
   Shared Entities: Catalan, Sephardic Judaism, Spanish, La Masia, Barcelona, Sardinia, Royal Spanish Football Federation
2. **[Similarity: 25/8 (312.5%)] Chunk 2:** [Despite being the favourites and starting strongly, Barcelona finished the 2006–07 season without trophies. A pre-season US tour was later blamed for a string of injuries to key players. . .]
   Shared Entities: Lionel Messi, Messi, Barcelona, Real Madrid, Gerard
3. **[Similarity: 25/8 (312.5%)] Chunk 6:** [It was announced in summer of 2012 that Tito Vilanova, assistant manager at FC Barcelona, would take over from Pep Guardiola as manager. . .]
   Shared Entities: Lionel Messi, Spanish, Barcelona, Real Madrid, Gerard
4. **[Similarity: 25/8 (312.5%)] Chunk 10:** [Messi opened the 2015 – 16 season by scoring twice from free kicks in Barcelona's victory (after extra time) over Sevilla in the UEFA Super Cup. . .]
   Shared Entities: Lionel Messi, Messi, Barcelona, Real Madrid, Gerard
5. **[Similarity: 25/8 (312.5%)] Chunk 7869:** [Born in Barcelona on September 27, 1916, he passed the entrance exams as a pilot in December 1936 and joined Aviacion in March 1937. . .]
   Shared Entities: Arsenal, Barcelona, February, France

Community Analysis Report

Details:
Total Communities: 30
Total Chunks: 11,656
Coverage: 72.0% (8,387/11,656)
Community Size Distribution:
Largest Community: 602 chunks
Smallest Community: 451 chunks
Average Size: 487.7 chunks
Median Size: 483 chunks

Community #1 (Size: 561 chunks)
Hash Prototype:
10001-00000-10000-11000-00111-01100-11011-01101-10110-01101-11100-01010
Core Entities: English, Germany, Italian, World War, European, April, February…

1. **[Chunk 4130]** On leaving the University of Oxford, in 1676, Edmond Halley visited Saint Helena and set up an astronomical observatory…
   Entities: Saint Helena, University, Oxford, Edmond Halley, Southern Hemisphere, Saint Mathew, Church, Hutt, Gate, Longwood, Halley, Mount
2. **[Chunk 37]** Boiron is a manufacturer of homeopathic products, headquartered in France and with an operating presence in 59 countries worldwide. It is the largest manufacturer of homeopathic products in the world…
   Entities: Boiron, France, Small
3. **[Chunk 39]** Before creation of Warsaw Pact, fearing Germany rearmed, Czechoslovak leadership sought to create security pact with East Germany and Poland. These states protested strongly against re-militarization of West Germany…
   Entities: Warsaw Pact, Before, Germany, Czechoslovak, East Germany, Poland, West Germany, Soviet, European, German, Soviets, Eastern Europeans, As Soviet Union, Pact, Previously, March, German Militarism
4. **[Chunk 10288]** Mianos (in Aragonese: both "Mians") is a municipality located in the province of Zaragoza, Aragon, Spain. According to the 2004 census (INE), the municipality has a population of 45 inhabitants…
   Entities: Mianos, Aragonese, Mians, Zaragoza, Aragon, Spain, According
5. **[Chunk 6206]** The Comedy Company was an Australian comedy television series first aired from 16 February 1988 until about 11 November 1990 on Network Ten, Sunday night and was created and directed by cast member Ian McFadyen…
   Entities: The Comedy Company, Australian, February, November, Network Ten, Sunday, Ian, Jo Lane, Sketch, Australia You, Generation, Melbourne, Kylie Mole, Mary, Anne Fahey