Query-Aware Knowledge Retrieval via Hyperbolic Structuring

Chuang Zhou[1,2], **Junnan Dong**[2,†], **Yilin Xiao**[1], **Shengyuan Chen**[1],
Su Dong[1], **Di Yin**[2], **Xing Sun**[2], **Zhaozhuo Xu**[3], **Xiao Huang**[1,*]
[1]The Hong Kong Polytechnic University, China; [2]Tencent Youtu Lab, China;
[3]Stevens Institute of Technology, USA
[1]{chuang-qqzj.zhou, yilin.xiao, shengyuan.chen, su.dong} @connect.polyu.hk
[2]hansonjdong@tencent.com; xiaohuang@comp.polyu.edu.hk; [3]zxu79@stevens.edu

Abstract

Retrieval-Augmented Generation (RAG) has demonstrated significant potential in enhancing large language models (LLMs) by supplementing external knowledge. However, existing approaches focus primarily on retrieving isolated factual knowledge entities while neglecting the critical reasoning relationships. To address this limitation, Graph-Augmented Generation (GraphRAG) has emerged as an effective solution, which explicitly integrates structured knowledge graphs to support complex reasoning tasks. Although diverse graph construction methods have been explored, they typically rely on static, query-agnostic graphs constructed via fixed heuristics. We are thereby motivated to propose a query-centric retrieval framework that adaptively constructs a graph tailored to each query. However, it is challenging to accurately identify these latent relationships from queries to the corpus. Moreover, unifying multiple local-perspective connections into a globally coherent structured corpus introduces additional complexity. To this end, we introduce HyperRAG, a novel framework in the Hyperbolic space that captures both explicit entity-based links and implicit query-aware connections. Extensive experiments on three benchmark datasets demonstrate that our framework consistently outperforms existing baselines.

1 Introduction

Recent advances in retrieval-augmented generation (RAG) have demonstrated its importance by integrating external knowledge to improve the downstream performance of large language models (LLMs) (Gao et al., 2024; Lewis et al., 2020a). While effective for fact-based queries, conventional RAG systems only retrieve fragmented passages without capturing the rich logical relations (Dong et al., 2023; Liang et al., 2024). This limitation becomes particularly apparent in complex domains requiring multi-hop reasoning (Zhou et al., 2024).

The emergence of graph-augmented generation (GraphRAG) represents an effective solution, enhancing retrieval with structured knowledge representations as a graph (Zhang et al., 2025; Xiao et al., 2025; Yang et al., 2026). Current graph construction methods in GraphRAG systems can generally be grouped into two paradigms. Hierarchical tree structures provide multi-scale organization of textual corpora, but typically rely on relatively rigid hierarchies (Sarthi et al., 2024). Knowledge graph-based approaches, on the other hand, are effective at modeling explicit factual relations, yet often depend on predefined schemas or entity extractions (Edge et al., 2024; Luo et al., 2025). Despite their strengths, both paradigms commonly adopt static, query-independent graph construction strategies, frequently based on heuristics such as entity co-occurrence or clustering. As a result, the constructed graph primarily captures global knowledge and may not fully reflect the specific reasoning paths required by individual queries. As shown in Figure 1, a typical graph built on a Disney-related corpus often groups entities based on general categorical features (e.g., grouping animated films by the topic). Such structures fail to answer query-specific questions such as *"Which film was released right after the Disney Renaissance period by Disney?"* since they do not capture latent relations.

Therefore, we aim to model latent relations among concepts by adequately exploiting the query-specific relations that guide graph construction. However, this remains challenging for two major reasons. First, complex queries often involve multiple implicit components, which complicates grounding them in concrete knowledge. Second, it is difficult to integrate locally induced query-specific knowledge into a unified graph while maintaining efficiency. To this end, we introduce HyperRAG, a query-centric framework that dynami-

*Xiao Huang is the corresponding author. †Junnan Dong serves as the project lead.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 21601–21614
July 2-7, 2026

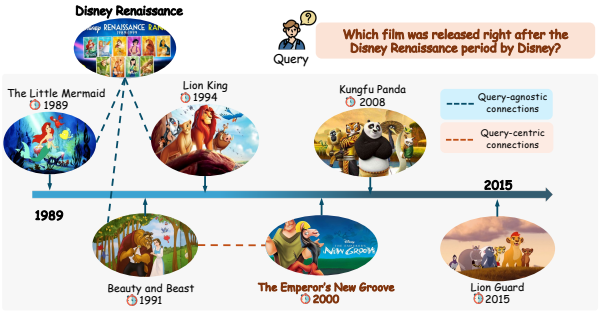

Figure 1: An illustration of query-agnostic versus query-centric graph construction. The former connects passages based on shared themes, while the latter links passages guided by the specific query. This demonstrates our motivation to coherently fuse both relationship types.

cally constructs a graph tailored to each query. Our method adopts a dual-stage strategy to first identify explicit entity-level connections and then decompose the query into fine-grained relational cues to induce implicit connections. Each query-specific graph is modeled as a minimal subtree, which is subsequently integrated into a unified structure. To preserve hierarchical properties during learning, we embed the graph in hyperbolic space, which is well-suited for representing tree-structured data.

Contributions are summarized as follows:

- We propose a novel query-centric graph construction paradigm that leverages both corpus-level knowledge and implicit relations induced by the query itself, enabling adaptive reasoning.
- Our framework adopts a hierarchical graph unification mechanism that integrates minimal subtrees into a global graph through localized training and conducts learning in hyperbolic space to better capture hierarchical and logical structures.
- Extensive experiments show HyperRAG consistently outperforms existing methods without incurring significant additional computational cost.

2 Preliminary

Let $\mathcal{Q} = q_1, q_2, \ldots, q_N$ denote the set of input questions, and let $\mathcal{C} = p_1, p_2, \ldots, p_M$ represent the corpus containing M textual passages. For each question $q_i \in \mathcal{Q}$, we construct a query-specific graph $G_i = (V_i, E_i, S_i)$, where $V_i \subseteq \mathcal{C}$ is the set of relevant passages (nodes), $E_i \subseteq V_i \times V_i$ is the set of undirected edges connecting semantically related passages, and $S_i : V_i \to \mathbb{T}$ maps each node to its corresponding textual content. A text encoder $f : \mathbb{T} \to \mathbb{R}^d$ maps both passages and questions into a d-dimensional embedding space. These embeddings are further projected into a Poincaré ball hyperbolic space $\mathbb{H}^d$ for downstream distance-based retrieval.

3 Methodology

Our primary goal is to construct a query-guided graph over a textual corpus to enhance retrieval-augmented generation performance. Instead of relying on a predefined knowledge graph, our method dynamically builds a graph that is tailored to each input question. This allows the structure of the graph to reflect the dependency relationships most relevant to the query. Existing approaches based on knowledge graphs often suffer from limited coverage and irrelevant connections, where relationships are predefined and typically represented in the form of triples. They often result in information loss by compressing rich textual content into sparse symbolic facts. In contrast, our approach preserves the full textual context. As shown in Figure 2, our hyperbolic-space framework provides a flexible and expressive structure that supports both query-centric and semantically-aware retrieval, enhancing the downstream question-answering tasks.

3.1 Graph Construction

HyperRAG constructs its reasoning graph through three sequential operations: (1) extracting static, corpus-derived explicit connections; (2) dynamically building query-aware implicit edges via structured reasoning decomposition; and (3) fusing both connection types into an integrated graph that is prepared to be embedded in hyperbolic space.

Explicit Knowledge Connection. We incorporate an explicit knowledge connection module that simulates the relational structure found in traditional knowledge graphs. In knowledge graphs, entities such as "*Arthur Conan Doyle*", "*Sherlock Holmes*", and "*Dr. Watson*" are linked through well-defined relations like writes, assistant of, and so on. While symbolic triples (head, relation, tail) enable structured reasoning, they often rely on external rules. For simplicity, we establish explicit edges between passages based on pairwise entity co-occurrence. For each passage p_i, we extract a set of key terms $\mathcal{T}(p_i)$ using lightweight keyword extraction. An explicit edge is created between p_i and p_j if they share a significant proportion of key terms:

$$e_{ij} = \mathbb{I}\left[\frac{|\mathcal{T}(p_i) \cap \mathcal{T}(p_j)|}{\min(|\mathcal{T}(p_i)|, |\mathcal{T}(p_j)|)} \geq \theta\right],$$

where θ is a similarity threshold (set to 0.15 in practice) and $\mathbb{I}[\cdot]$ denotes the indicator function. To avoid over-linking, each passage is restricted to connect to at most five neighbors via such edges.

These connections form an entity-aware backbone graph that captures static and corpus-level associations independent of any specific query.

Query-guided Implicit Connection. While explicit connections provide basic entity-level linkage across the corpus, they remain independent of any specific information need. To introduce query relevance into the graph structure, we propose a query-guided implicit connection mechanism. The key idea is to leverage the reasoning capabilities of LLMs to infer logical relationships between passages in the context of a specific question. Given a query q, we first conceptualize it as a high-level reasoning task that can be decomposed into a set of atomic reasoning units $\mathcal{U}_q = \{u_1, u_2, ..., u_n\}$. Each unit u_i represents a minimal reasoning operation and generates a precise retrieval directive to fetch the corresponding evidence passages required for its resolution. By resolving each unit with its specifically retrieved evidence and propagating these intermediate results, this module bridges the gaps left by purely semantic retrieval, thereby complementing explicit-edge approaches and implicitly establishing query-oriented functional links.

Fusion of Explicit and Implicit Edges. To construct a unified hierarchical structure that combines both factual and query-specific relationships, we propose a tree fusion mechanism that merges the **explicit** and **implicit** connections into a single tree structure. The explicit tree is derived from surface-level cues, such as co-occurring entities or shared keywords among corpus chunks, capturing predefined and globally relevant knowledge relations. In contrast, the implicit tree is constructed dynamically for each query q by leveraging a language model to identify semantically useful passages and connect them based on their contextual relevance to q. Given two sub-trees that exhibit explicit and implicit dependency relationships between textual segments, we perform fusion by first identifying shared nodes across both trees, i.e., nodes that reference the same passage or contain the same key entity. These shared nodes are unified into a single node in the resulting graph $\mathcal{G}$, and their respective subtrees are recursively merged while preserving the original parent-child directionality. To ensure that the resulting structure remains a valid tree, we prevent the formation of any cycles by discarding redundant edges that introduce loops. The ultimate tree thus integrates both global knowledge priors and query-dependent reasoning chains, offering a rich and hierarchically coherent structure suitable for downstream retrieval tasks. This mechanism enables the discovery of non-obvious semantic connections by translating high-level reasoning steps into concrete retrieval operations.

3.2 Learning Hyperbolic Representations

To faithfully encode the hierarchical and query-sensitive structure of our fused graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a key design choice is how to define the distance between nodes that capture both semantic and structural information. To this end, we introduce a hyperbolic embedding strategy specifically tailored for Retrieval-Augmented Generation over structured corpora. We embed all nodes (i.e., passages) into the Poincaré ball $\mathbb{B}^d = \{x \in \mathbb{R}^d : \|x\| < 1\}$, where distances increase exponentially with radius, allowing for separation of semantically close versus distant nodes (Nickel and Kiela, 2017).

Motivation for Hyperbolic Distance. Common L^p norms, each with distinct properties (Coghetto, 2016; Debeye and Van Riel, 1990), universally satisfy the triangle inequality. As the number of hops increases, passages with similar semantic embeddings have endpoints p_0 and p_k that remain close in the metric space. As k increases, the global semantic distance between distant nodes in the chain is underestimated. This becomes a problem in reasoning tasks where longer chains should reflect logical difference or inferential effort. For instance, passages separated by multiple reasoning steps shouldn't be treated as nearly the same as directly connected ones. Therefore, we choose hyperbolic geometry due to its **exponential expansion** property.

$$D = \sum_{i=0}^{k-1} \delta(p_i, p_{i+1}),$$

$$\mathrm{Vol}_{\mathbb{R}^n}(r) = \frac{\pi^{n/2}}{\Gamma(n/2+1)} r^n,$$

$$\mathrm{Vol}_{\mathbb{H}^n}(r) = \int_0^r \sinh^{n-1}(t)dt \sim C \cdot e^{(n-1)r}.$$

Traditional Euclidean space's linear growth cannot accommodate all nodes in deep trees. As the number of passages grows, embeddings of passages belonging to a long reasoning chain tend to cluster closely together. This is problematic for logical inference because a longer reasoning path should reflect greater inferential effort or more logical difference. For instance, passages separated by several hops should not be nearly as similar as those directly connected. In contrast, hyperbolic

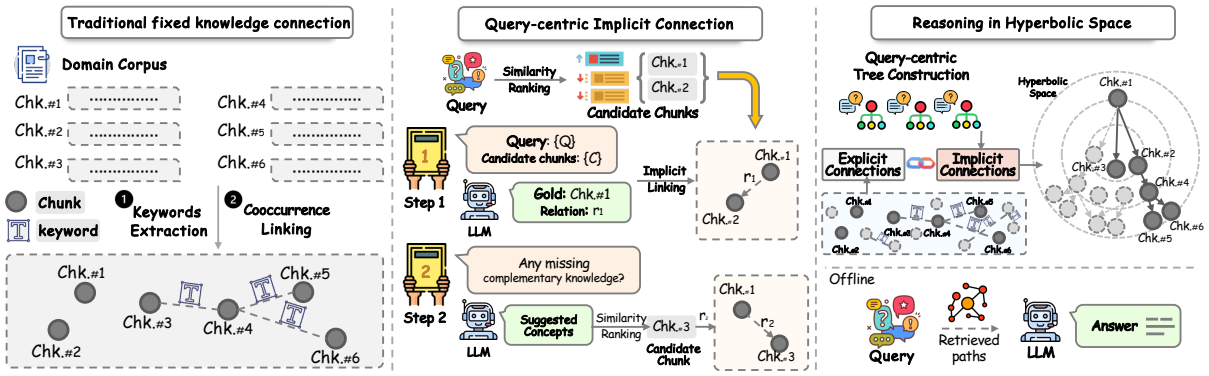

Figure 2: The overall framework of HyperRAG. We begin by preprocessing the corpus using keyword extraction techniques to connect similar passages, often grouped by shared topics or entities. Then, for each question, we identify the underlying logical connections among passages. These connections are embedded into a hyperbolic space to reflect the hierarchical tree-structured nature. Finally, we retrieve the top-K items for question answering.

space exhibits exponential volume growth with radius. This property allows it to separate nodes more aggressively as their distance from a conceptual root increases, **thus preserving hierarchical and multi-hop relational structures** inherently during graph learning. In our framework, direct logical edges are encoded with short hyperbolic distances, capturing immediate inferential relationships. Please refer to Appendix A.1 for detailed discussion and mathematical illustrations.

Semantic and Structural Learning. We initialize the Bert-based embeddings and optimize them such that connected nodes in the graph are mapped closer in hyperbolic space. The loss function numerator minimizes the sum of pairwise Poincaré distances between adjacent nodes: $\sum_{(u,v)\in\mathcal{E}} d_{\mathbb{B}}(x_u, x_v)$, where $x_u, x_v \in \mathbb{B}^d$ are the hyperbolic embeddings of nodes u and v, and $d_{\mathbb{B}}(\cdot,\cdot)$ denotes the Poincaré distance. Intuitively, this distance measures how far apart two points are in a curved hyperbolic space. It can be viewed as a projection of a negatively curved space onto the interior of a Euclidean ball. As points move closer to the boundary, they become exponentially farther away, which is shown below:

$$d_{\mathbb{B}}(x,y) = \operatorname{arcosh}\left(1 + \frac{2\|x-y\|^2}{(1-\|x\|^2)(1-\|y\|^2)}\right).$$

This loss encourages embeddings of connected passages to move closer together while naturally preserving the hierarchical structure due to the curvature of the space. By jointly considering both explicit and implicit edges, the model learns a unified geometric representation that captures not only surface-level semantic similarity but also the deeper logical structure through query-centric reasoning.

3.3 Hybrid Graph Update & Retrieval

To enable efficient and sustainable knowledge integration, we design a hybrid update strategy that balances immediate adaptability with long-term representational consistency. Our method operates in two complementary phases: a lightweight *incremental fine-tuning* phase for immediate knowledge absorption, and a periodic *global recalibration* phase for maintaining geometric coherence.

Phase 1: Incremental Fine-tuning. For each query q_t arriving at time step t, let ΔG_t denote the implicit sub-graph derived through query-centric reasoning. Instead of retraining the entire embedding matrix, we perform localized optimization that focuses exclusively on the newly introduced components. Specifically, we freeze the embeddings of all nodes not incident to edges in ΔG_t, and update only the embeddings of nodes $\mathcal{V}_\Delta = \{v \mid (u,v) \in \Delta G_t\}$ and their immediate neighbors $\mathcal{N}(\mathcal{V}_\Delta)$ within a one-hop radius. The loss for this incremental step is defined as:

$$\mathcal{L}_{\text{inc}}(t) = -\sum_{(u,v)\in\Delta G_t} \log \frac{e^{-d_{\mathbb{D}}(z_u,z_v)}}{\sum_{v'\in\mathcal{N}(u)} e^{-d_{\mathbb{D}}(z_u,z_{v'})}},$$

where $\mathcal{N}(u)$ denotes the set of negative samples for node u. This localized update completes in milliseconds, allowing the system to immediately reflect new logical connections while preserving the main part of the existing structure.

Phase 2: Periodic Global Recalibration. Since repeated local updates can gradually introduce geometric distortion throughout the embedding space. we periodically perform full-graph retraining after a fixed number of queries (empirically every 100 queries) to preserve long-term coherence and

integrate knowledge holistically. In this stage, all recently accumulated implicit subgraphs alongside the original static graph are jointly re-embedded in hyperbolic space. This global recalibration resets the structural drift introduced by incremental updates and reinforces the consistency of the overall geometric representation. The process runs asynchronously, ensuring that the real-time retrieval system remains responsive while the knowledge graph evolves in a stable and integrated manner.

This streaming paradigm adapts our model to evolving real-world environments. We leverage the learned geometry to retrieve relevant information for question answering. Specifically, given a query q, we first encode it into a hyperbolic embedding $\mathbf{z}_q \in \mathbb{B}^d$ using the same sentence encoder followed by projection into the Poincaré ball. The goal is to identify leaf nodes in the graph, i.e., raw passages from the corpus that are closest to the query in hyperbolic distance. At inference time, we compute the hyperbolic distance between $\mathbf{z}_q$ and all leaf node embeddings $\{\mathbf{z}_i\}$, where each $\mathbf{z}_i \in \mathbb{B}^d$ represents a passage. According to the experimental trials, we typically select three candidate passages for retrieval and then feed these along with the question into the language model in the form of a prompt to generate the answer. By retrieving in hyperbolic space, we benefit from the hierarchical structure of the graph, which allows for efficient identification of semantically and structurally relevant content, even across multi-hop relationships.

4 Experiments

We conduct experiments on three publicly open-source datasets: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and the Musique (Trivedi et al., 2022) dataset. Each dataset provides different query structures and reasoning depths, enabling a comprehensive analysis of how well our hyperbolic graph construction perform across varying levels of complexity and domain specificity. Our study aims to address the following questions: (1) How does HyperRAG compare to existing state-of-the-art answer generation pipelines in terms of performance? (2) How does our method demonstrate its time and token-level efficiency compared to other baselines? (3) How do different settings influence the performance of our framework? (4) How can we intuitively distinguish between hyperbolic and Euclidean distances, and how are these patterns reflected in empirical data?

4.1 Experimental Settings

Datasets. We utilize three datasets for our experiments, each comprising 1,000 questions sourced from their respective domains. Each dataset's corpus contains approximately ten thousand independent chunks of information, allowing for a diverse range of queries and contexts. These datasets collectively represent progressively complex reasoning scenarios, spanning diverse domains from basic scientific terminology to humanities subjects, providing a comprehensive evaluation of our method's capability to handle various complexity levels in knowledge-intensive tasks. To ensure the consistency and fairness of the experiments, we strictly maintain the same questions and corpus as utilized in previous studies (Gutiérrez et al., 2024).

Implementation Details. Our experiments were consistently conducted using GPT-4o-mini as the language model. During the training stage of the model within the hyperbolic space, we utilize the Adam optimizer with an initial learning rate of 0.01. The hyperparameters, such as the number of epochs and retrieval items, are tuned through a grid search, with the best values chosen for each dataset. We also record the time and token consumption.

Baselines. To comprehensively evaluate the effectiveness of our proposed framework, we compare it with a diverse set of baselines spanning several paradigms. These methods are categorized into three groups: the first includes direct language model approaches that use pre-trained models like GPT or Llama to answer questions through zero-shot prompting without additional retrieval mechanisms; the second group consists of enhanced LM methods that incorporate auxiliary techniques such as similarity-based retrieval and chain-of-thought prompting to improve reasoning capabilities; the third category involves graph-based augmented generation approaches that leverage structured knowledge graphs, including methods based on common knowledge graphs and hierarchical clustering strategies.

Evaluation Metrics. For evaluation metrics, existing methods primarily rely on string matching techniques to assess answer accuracy. The most commonly adopted metrics include Exact Match, which requires the predicted answer to exactly match the ground truth string, and Answer Containment, which checks whether the ground truth answer is contained within the model's prediction. However, the correctness of answers can sometimes

Table 1: The following main results table presents a comprehensive performance comparison between state-of-the-art baselines and HyperRAG on three benchmark datasets in terms of both String-Match and GPT-evaluation Accuracy.

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	3.9	4.8
Llama3 (13b)	10.6	11.7	13.1	10.6	4.7	5.4
GPT-3.5	28.3	39.8	27.3	31.6	13.2	17.9
GPT-4o-mini (Achiam et al., 2023)	30.4	32.1	31.0	33.9	12.5	18.3
Retrieval-augmented Variants						
Retrieval (Top-1)	38.4	42.6	34.8	37.3	16.2	20.5
Retrieval (Top-3)	43.2	45.1	41.2	43.5	20.6	23.2
Retrieval (Top-5)	44.1	45.9	40.7	42.4	20.9	23.8
Graph-enhanced Generation Methods						
KGP (Wang et al., 2024)	46.4	47.1	41.5	43.7	23.3	27.3
G-retriever (He et al., 2024)	41.3	40.9	26.7	25.7	14.1	15.6
LightRAG (Guo et al., 2024)	47.8	52.7	46.3	43.3	28.3	27.7
DALK (Li et al., 2024)	45.8	50.6	45.6	42.6	22.4	25.3
HippoRAG (Gutiérrez et al., 2024)	46.5	51.6	43.9	41.4	21.8	24.3
RAPTOR (Sarthi et al., 2024)	48.1	55.3	47.7	43.9	28.2	29.7
GFM-RAG (Luo et al., 2025)	55.1	56.2	49.1	48.1	31.3	32.6
HippoRAG2 (Gutiérrez et al., 2025)	53.7	55.6	49.7	50.8	32.0	33.8
HyperRAG	**57.4**	**58.9**	**50.8**	**52.3**	**34.7**	**35.4**

be overlooked if the phrasing differs, even when the answer is correct. To address this, GPT-based evaluation methods have gradually been adopted. We utilize a dual evaluation strategy comprising both string-matching and LLM-based judgment to achieve a fair and clarifying comparison. This is motivated by the need to overcome the limitations of any single metric, balancing string-based strictness with flexibility of semantic judgment.

4.2 Main Results

The overall comparison between HyperRAG and other baselines is presented in Table 1. As mentioned above, the baselines are categorized into three groups. From the zero-shot performance, it is evident that advanced LLMs already possess considerable answering capability, reflecting a strong storage of background knowledge. Additionally, similarity-based retrieval and chain of thought prompting significantly improve performance, demonstrating that incorporating supplementary information through multi-round interaction effectively enhances answer quality. Thirdly, overall, graph-based variants tend to outperform purely semantic similarity-based methods. This improvement stems from the graph structure's ability to enhance reasoning capacity by explicitly modeling relationships and dependencies. However, not all graph RAG approaches are equally effective. Sometimes, they do not surpass direct retrieval methods. This highlights the importance of how the graph is constructed: the quality and relevance of the graph structure are crucial factors that determine whether the graph-based approach will yield better results. It can be observed that our method consistently outperforms existing baselines, confirming the validity and effectiveness of the constructed graph. The improvements can be attributed to the combination of corpus-level knowledge and query-centric connections. Empirical results have shown that incorporating additional information retrieved in hyperbolic space significantly improves the model's performance, as it provides richer, more relevant context that helps the model better understand complex queries.

4.3 Efficiency Analysis

Our HyperRAG framework introduces two major sources of computational cost. The first occurs during query-aware graph construction, where we reason over the relationships between the input question and its retrieved candidate passages. For a dataset of 1,000 queries and around ten thousand passages, the complete graph can be constructed in a few minutes. Besides, the inference phase demonstrates significantly higher efficiency especially when employing a single-round approach. Once the graph is constructed, the system only re-

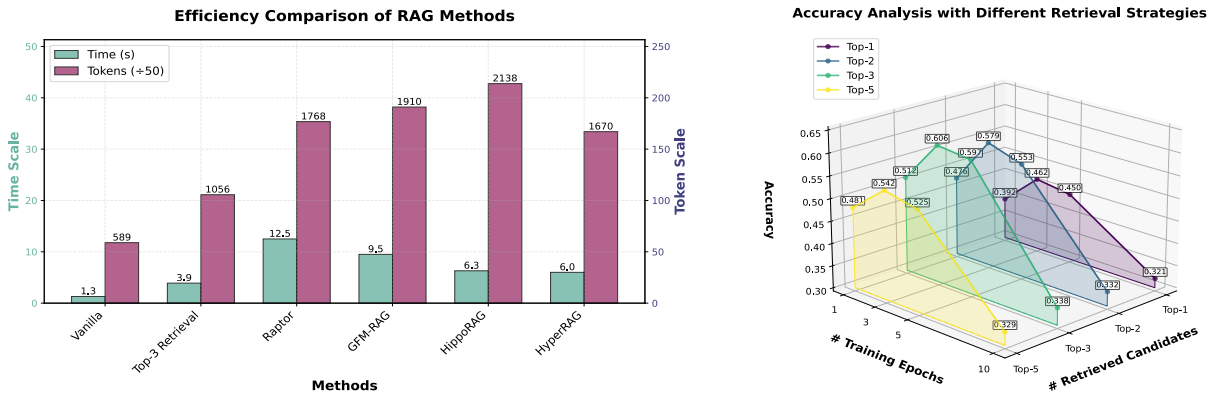

Figure 3: (Left) Time and token consumption across different methods; (Right) Effect of hyperparameter settings.

quires a lightweight search within the hyperbolic space, making it suitable for practical real-life use. In this optimized setup, each query processing time is reduced to approximately 1.5 seconds, encompassing both retrieval and answer generation. Since the current implementation processes tasks iteratively, the framework can be further accelerated through integrating multi-processing techniques.

For a more comprehensive evaluation, we compare HyperRAG against a range of baselines, including zero-shot LLM, similarity-based dense retrieval, and several graph-based retrieval variants on the left of Figure 3. As expected, the zero-shot method is the fastest since it bypasses retrieval entirely. Dense retrieval approaches, such as BERT-based retrievers, offer a good tradeoff with slightly higher latency but much better answer quality. GraphRAG models exhibit the most variability in runtime due to differences in graph construction strategies, the number of retrieval hops, and the multi-step inference. HyperRAG incurs only a modest computational cost compared to simple retrieval while being efficient than other graph-based counterparts. By leveraging a fast hyperbolic retrieval technique and capped-round reasoning, our model achieves highly competitive efficiency.

4.4 Semantic & Structural Learning Impacts

In our framework, the ultimate passage representations result from a combination of initial semantic embeddings and structural refinement through hyperbolic space learning. We analyze how the number of training epochs in hyperbolic representation learning affects performance on the right of Figure 3. This process essentially controls the tradeoff between semantic fidelity and structural alignment. When the number of training epochs is too small, the learned embeddings remain largely similar to the initial BERT-based representations, preserving strong semantic similarity but failing to encode deeper relational structure across the reasoning graph. In contrast, when trained for too many epochs, the representations become dominated by the graph's structural connections. Our experiments show that setting the number of hyperbolic training epochs to 3 provides a good balance. Another important factor affecting performance is the number of retrieved passages during inference. Interestingly, we observe that retrieving more candidates does not always help. When the number of retrieved passages exceeds 5, the overall performance tends to decrease. This suggests that adding irrelevant content may hinder the model's ability to focus on the most useful evidence.

Method	HotpotQA	Wiki	Musique
Semantic retrieval	45.1	43.5	23.2
Explicit edges	49.8	46.9	26.3
Implicit edges	55.2	50.0	31.9
HyperRAG	58.9	52.3	35.4

Table 2: Ablation study on effects of connection strategies based on different components of our framework.

Besides, we also test how important the explicit and implicit edges are by conducting an ablation study. Based on the results shown in Table 2, we observe that explicit edges contribute to a modest performance improvement across all datasets, while implicit edges significantly enhance the model's effectiveness. These results confirm that explicit and implicit connections play distinct roles in the reasoning process. Explicit edges effectively capture surface-level topical associations and structural relationships, providing a foundational context. Conversely, implicit edges are crucial for tracing the underlying logical pathways required to answer complex questions, enabling deeper inference. answer complex questions, enabling deeper inference.

4.5 Hyperbolic and Euclidean Analysis

To gain deeper insights into the properties of hyperbolic geometry in embedding spaces, we randomly sampled queries from the dataset and systematically compared their distance distributions to the corpus embeddings in both hyperbolic and Euclidean spaces. The phenomenon of larger hyperbolic distance values can be explained by the exponential expansion property of hyperbolic geometry. Under the same embedding dimensionality, hyperbolic space provides a more expansive representation space, enabling pushing unrelated entities to greater distances. This distance distribution characteristic offers two key advantages. **Enhanced Discriminative Power**: The larger distance numerical range provides richer gradient signals for the model, facilitating learning of more precise similarity decision boundaries. **Improved Ranking Stability**: The amplification effect of distance differences makes retrieval rankings more stable, reducing sorting uncertainties in borderline cases.

Metric	Hyperbolic Distance			Euclidean Distance		
	Max	Min	Mean	Max	Min	Mean
Top-1	2.22	1.59	1.87	1.11	0.29	0.71
Top-3	2.23	1.56	1.97	1.33	0.29	0.87
Top-10	2.29	1.45	2.10	1.44	0.27	1.06

Table 3: Statistical comparison of two distance metrics.

A critical challenge in high-dimensional Euclidean space is the distance concentration phenomenon (François et al., 2007). The squared Euclidean distance asymptotically follows a normal distribution. This leads to poor discrimination between the nearest and furthest neighbors. In contrast, the logarithmic growth of the arcosh function for large arguments and the amplification of small coordinate differences near the boundary induce a heavy-tailed distance distribution. This distribution is right-skewed, which maintains a high level of relative variance and enhances the separation ability of hierarchical levels. By amplifying distinctions between close and distant points, it facilitates more effective retrieval and reasoning in hierarchical graph structures, as evidenced in Table 3.

5 Related Work

Retrieval-augmented generation (RAG) has become a prominent paradigm for open-domain and multi-hop question answering, where an external corpus is indexed and queried to retrieve relevant documents that are then fed into a generative model (Lewis et al., 2020b). Early approaches such as REALM (Guu et al., 2020) and DPR (Sachan et al., 2021) focus on encoding large text corpora into dense embeddings, enabling scalable and differentiable retrieval. Subsequent work improved retrieval-augmented generation by incorporating fusion mechanisms or editable memory (Bajaj et al., 2022; Hofstätter et al., 2023). To better capture the semantic and logical relationships between passages, some studies explored graph-based structures that model paragraph connectivity through co-reference, discourse, or logical links (Zhou et al., 2025b; Chen et al., 2026; Zhou et al., 2025a). Building on this intuition, GraphRAG approaches explicitly incorporate graph structures to enrich the retrieval process, offering additional reasoning paths or relational priors to assist generation (Edge et al., 2024; Dong et al., 2025; Zhou et al., 2025c; Xiao et al., 2026). Among them, HippoRAG (Gutiérrez et al., 2024) and GFM-RAG (Luo et al., 2025) are examples of KG-based methods. HippoRAG leverages external knowledge graphs to enhance context understanding during retrieval and generation. GFM-RAG goes further by constructing and completing knowledge graphs from text, aiming to improve downstream generation through enhanced graph learning. In contrast, RaptorRAG (Sarthi et al., 2024) avoids reliance on external knowledge bases by applying hierarchical clustering over the corpus. This yields a multi-level document structure, allowing queries to navigate the corpus in a coarse-to-fine manner. In summary, most existing methods construct static graphs either through predefined relations between passages.

6 Conclusion

We observe that knowledge passages are connected not only through explicit relationships but also through implicit logical relationships that emerge only under specific questions. To this end, we introduce a novel approach that fuses both perspectives into a single graph, enabling a query-centric retrieval. Moreover, by recognizing the inherent tree-like structure of multi-hop reasoning, we get rid of traditional Euclidean representations and instead learn embeddings in a hyperbolic space, which better preserves hierarchical relationships and logical distance. Compared with traditional models, our method achieves superior performance, highlighting its potential for deployment and future research.

Limitations

Although computing hyperbolic distances is not inherently complex, unlike traditional norm-based embeddings, it cannot directly leverage highly optimized search libraries such as FAISS. As a result, retrieval efficiency is slightly lower. In future work, we plan to address this limitation by incorporating parallelization and multi-process search strategies to accelerate hyperbolic distance computations.

Ethical Considerations

We strictly adhere to ACL's ethical guidelines in conducting this research. All datasets used in our experiments are publicly available, and there is no personally identifiable information included.

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*.

Payal Bajaj, Chenyan Xiong, Guolin Ke, Xiaodong Liu, Di He, Saurabh Tiwary, Tie-Yan Liu, Paul Bennett, Xia Song, and Jianfeng Gao. 2022. Metro: Efficient denoising pretraining of large scale autoencoding language models with model generated signals. *arXiv preprint arXiv:2204.06644*.

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.

Roland Coghetto. 2016. Chebyshev distance.

HWJ Debeye and P Van Riel. 1990. Lp-norm deconvolution1. *Geophysical Prospecting*, 38(4):381–403.

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, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. 2023. Hierarchy-aware multi-hop question answering over knowledge graphs. In *Proceedings of the ACM web conference 2023*, pages 2519–2527.

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*.

Damien François, Vincent Wertz, and Michel Verleysen. 2007. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

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*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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*.

Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1437–1447.

Bernard S Kay. 1992. The principle of locality and quantum field theory on (non globally hyperbolic) curved spacetimes. *Reviews in Mathematical Physics*, 4(spec01):167–195.

Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. 2011. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 12:953–997.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a.

Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Dawei Li, Shu Yang, Zhen Tan, Jae Young Baik, Sukwon Yun, Joseph Lee, Aaron Chacko, Bojian Hou, Duy Duong-Tran, Ying Ding, and 1 others. 2024. Dalk: Dynamic co-augmentation of llms and kg to answer alzheimer's disease questions with scientific literature. *arXiv preprint arXiv:2405.04819*.

Xun Liang, Simin Niu, Sensen Zhang, Shichao Song, Hanyu Wang, Jiawei Yang, Feiyu Xiong, Bo Tang, Chenyang Xi, and 1 others. 2024. Empowering large language models to set up a knowledge retrieval indexer via self-learning. *arXiv preprint arXiv:2405.16933*.

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*.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30.

Érick O Rodrigues. 2018. Combining minkowski and chebyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier. *Pattern Recognition Letters*, 110:66–71.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. *arXiv preprint arXiv:2101.00408*.

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*.

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.

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. 2026. 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, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

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.

Huachi Zhou, Jiahe Du, Chuang Zhou, Chang Yang, Yilin Xiao, Yuxuan Xie, and Xiao Huang. 2025c. Each graph is a new language: Graph learning with llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17548–17559.

A Appendix

A.1 Distance Choice for Graph Reasoning

After constructing a passage graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a key design choice is how to define the semantic distance between nodes. Common metrics include L^p norms, each with different characteristics (Kloft et al., 2011). Let $x, y \in \mathbb{R}^n$ be embeddings of two passages. The general L^p norm is defined as:

$$d_p(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}. \quad (1)$$

There are some popular variants of the general L^p norm family, each offering a different geometric intuition and sensitivity to feature dimensions. The L^1 norm (Manhattan distance) emphasizes coordinate-wise differences, the L^2 norm (Euclidean distance) captures straight-line distance in space, and the L^∞ norm (Chebyshev distance) reflects the maximum single-coordinate deviation. These distances are commonly used for computing similarity between item embeddings. d_1, d_2, and d_∞ denote the Manhattan, Euclidean, and Chebyshev distances respectively (Rodrigues, 2018), which are presented in the following:

$$\begin{aligned} d_1(x, y) &= \sum_{i=1}^{n} |x_i - y_i|, \\ d_2(x, y) &= \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{\frac{1}{2}}, \\ d_\infty(x, y) &= \max_i |x_i - y_i|. \end{aligned} \quad (2)$$

While desirable in many metric spaces, these metrics universally satisfy the triangle inequality. As the number of hops increases (i.e., $p_1 \to p_2 \to \cdots \to p_k$), passages with similar semantic embeddings have endpoints p_0 and p_k that remain close in the metric space. As k increases, the global semantic distance between distant nodes in the chain is underestimated.

$$d(x, z) \leq d(x, y) + d(y, z).$$

$$d(p_0, p_k) \leq \sum_{i=0}^{k-1} d(p_i, p_{i+1}).$$

This becomes a problem in reasoning tasks where longer chains should reflect increased difference or inferential effort. For instance, passages separated by multiple reasoning steps shouldn't be treated as nearly the same as directly connected ones. Euclidean-like metrics end up "flattening" the structure, losing sight of the underlying hierarchy or compositional relationships within the information. Besides, hyperbolic space enables a large number of nodes per parent, reflecting the branching nature of trees. Additionally, distances in hyperbolic space show logarithmic growth with respect to similarity, quantified by

$$d(u, v) \approx -\log(\text{similarity}(u, v)).$$

Furthermore, hyperbolic spaces can effectively embed high-dimensional trees, where a tree with a branching factor b and depth h contains

$$N = \frac{b^{h+1} - 1}{b - 1} \quad \approx \quad \frac{b^{h+1}}{b - 1} \sim O(b^h)$$

nodes. The constant negative curvature of hyperbolic spaces facilitates the efficient packing of nodes, allowing for more nodes in a given volume while preserving hierarchical relationships. For instance, the Poincaré disk model visually represents these properties, illustrating how distances correspond to relationships in a tree structure. Together, these features underscore hyperbolic space's suitability for accurately capturing the complexity of hierarchical tree structures.

Logically Aligned Proximity: In multi-hop QA, we often reason over chains of passages $\{p_0, p_1, \ldots, p_k\}$. Nodes in a parent-child logical relation (e.g., $p_i \to p_{i+1}$) can remain at small distances, while nodes far apart in reasoning hierarchy (e.g., $p_0 \to p_k$) can be naturally separated.

$$\begin{aligned} &d_{\mathbb{H}}(p_0, p_1) \approx d_{\mathbb{H}}(p_1, p_2) \approx \cdots \\ &\approx d_{\mathbb{H}}(p_{k-1}, p_k) \ll d_{\mathbb{H}}(p_0, p_k). \end{aligned}$$

Thus, logical edges are modeled with short hyperbolic distances, preserving the immediate inferential relationship between supporting facts. In contrast, nodes that are farther apart in the reasoning hierarchy, which may only share indirect or topic-level connections, are naturally pushed apart due to the exponential expansion of hyperbolic space. This creates a geometric alignment between reasoning depth and embedding distance, which is essential to logic-driven passages.

A.2 Hyperbolic Locality

The hybrid training strategy is motivated by the intrinsic geometry of hyperbolic space (Kay, 1992). Owing to its exponential volume growth, regions

that are sufficiently separated in the manifold exhibit weak mutual influence. As a result, local modifications to embeddings primarily affect their immediate neighborhood, while having negligible impact on distant regionsa property which is referred to as *hyperbolic locality*. This geometric characteristic enables efficient incremental updates: query-specific adjustments can be performed locally without disrupting the global hierarchical structure. To prevent the accumulation of drift over time, a periodic global training phase is introduced, which acts as a regularizer and restores consistent hierarchical separation across the entire graph. Formally, for any two nodes u and v with tree distance l in the reasoning hierarchy, their hyperbolic distance after T incremental updates satisfies

$$d_{\mathbb{D}}^{(T)}(z_u, z_v) \geq \alpha \cdot l - \beta \cdot \epsilon(T),$$

where $\alpha, \beta > 0$ are constants, and $\epsilon(T)$ is a bounded drift that is reset to zero after each global retraining. This guarantees that the embedding quality does not degrade indefinitely over time.

A.3 Hierarchical Properties

For each query, HyperRAG constructs a query-specific subgraph by incrementally retrieving and attaching relevant passages according to decomposed relational cues. Each newly introduced node is connected to an existing node via a unique parent relation, and no backward or cross-branch edges are allowed during construction. As a result, every query-specific subgraph is a tree rooted at the query, ensuring acyclic and hierarchical organization.

When multiple query-specific subgraphs are merged, the resulting unified graph is no longer a strict tree. However, due to the localized nature of query-driven expansion and the absence of arbitrary global connections, the merged structure largely preserves tree-like hierarchical properties. To quantitatively assess how close the unified graph remains to a tree, we evaluate several structural metrics on randomly sampled subgraphs:

$$\text{Excess Edge Ratio} = \frac{|E| - (|V| - c)}{|V|},$$

$$\text{Cycle Ratio} = \frac{|E| - |V| + c}{|E|},$$

$$\text{Average Branching Factor} = \frac{1}{|V|} \sum_{v \in V} \deg^+(v).$$

Across sampled merged graphs, we observe an average excess edge ratio of 0.06 and the cycle ratio remains below 0.05, indicating that only a small fraction of edges deviate from an ideal tree structure. We further analyze the average branching factor, defined as the mean out-degree across nodes. The observed branching factor is 1.3 ± 0.4, suggesting a stable and shallow hierarchical expansion rather than dense or flat connectivity.

B Experimental Details

B.1 Baseline Descriptions

Here we provide an overview of the baseline methods included in our experimental results.

- **KGP**: Builds a knowledge graph from text passages with the help of LLMs. During retrieval, an LLM-driven agent navigates the graph to progressively collect passages that support the answer.
- **G-Retriever**: Combines graph neural networks with LLMs by formulating subgraph retrieval as a Prize-Collecting Steiner Tree problem. This method targets improved conversational QA over textual graphs, aiming to reduce hallucinations.
- **RAPTOR**: Constructs a hierarchical tree over the corpus via recursive clustering and abstractive summarization, producing multi-level semantic representations suitable for retrieval.
- **DALK**: Designed to tackle domain-specific knowledge gaps, DALK dynamically integrates LLMs with an evolving knowledge graph. LLMs are used to construct KG from relevant literature combined with a coarse-to-fine sampling.
- **LightRAG**: Implements a two-layer indexing scheme that integrates fine-grained entity-relation graphs with coarse-grained thematic structures, balancing precision and efficiency.
- **HippoRAG**: A training-free approach that utilizes Personalized PageRank with query concepts as seeds, enabling single-step or multi-hop retrieval over the document graph.
- **GFM-RAG**: Constructs document-level knowledge graphs and uses a trained graph-aware retriever to locate relevant content efficiently.
- **HippoRAG2**: Improves performance on more basic factual memory tasks on the basis of the first version. It remains employing PPR algorithm and enhance its integration ability.

C Case Study

C.1 Implicit Connection

As shown below, while passage #1 explicitly defines the temporal boundary of the Disney Renaissance period (1989–1999), it does not itself mention any subsequent films. However, passages #4, #5, and #6 provide information about films and series released after this era. The connection between passage #1 and these later passages is implicit, as they are linked not by surface-level semantic similarity but by a shared temporal and logical context, i.e. the transition from the Disney Renaissance into the post-Renaissance period. In particular, passage #5 mentions The Emperor's New Groove (2000), which is directly situated after the time frame established in passage 1, thus forming a reasoning chain across the two. Similarly, passage #6 extends this timeline further, showcasing the evolution of Disney's productions.

Query

Q: Which film was released right after the Disney Renaissance period by Disney?

Retrieved Passages

Round-1 Retrieval:

1. In the history of The Walt Disney Company, the Disney Renaissance refers to the era from 1989 to 1999 during which Walt Disney Feature Animation (renamed Walt Disney Animation Studios in 2006) experienced a creative resurgence in producing successful animated films based on well-known stories, which restored public and critical interest in The Walt Disney Company as a whole.

2. Disney's The Little Mermaid is an American animated television series produced by Walt Disney Television Animation based on the 1989 Disney film of the same name.

3. Corso is a New York Foundation for the Arts Poetry Fellow and Sherwood Anderson Fiction Award Winner, and is the author of "Death by Renaissance" (2004).

Query Decomposition

The first paragraph is relevant and it's insufficient to answer the question. We need to know which Walt Disney films were released after 1999.

Retrieved Passages

Round-2 Retrieval:

4. The Apple Dumpling Gang is a 1975 American comedy-western film produced by Walt Disney Productions about a slick gambler named Russell Donovan (Bill Bixby) who is duped into taking care of a group of orphans who eventually strike gold during the California Gold Rush.

5. The Emperor's New Groove is a 2000 American animated buddy comedy film created by Walt Disney Feature Animation and released by Walt Disney Pictures. It is the 40th Disney animated feature film.

6. The Lion Guard is an American animated television series developed by Ford Riley based on Disney's 1994 film "The Lion King." The series was first broadcast with a television movie, titled The Lion Guard: Return of the Roar on Disney Channel on November 22, 2015, and began airing as a TV series on January 15, 2016, on Disney Junior and Disney Channel.

C.2 Explicit Connection

Passages #4, #8, and #9 can be explicitly connected as they all refer to the same work, The Apple Dumpling Gang. Passage #4 introduces the original film, while Passage #8 mentions a drama produced by Walt Disney Productions, and Passage #9 references the film's composer. These connections are straightforward and fact-based, forming an explicit linkage through shared entities and production history. In our graph construction, such connections form strong edges that preserve concrete semantic relationships within the corpus.

Retrieved Passages through Keyword Extraction

7. *"Down in New Orleans"* is a jazz song from Disney's 2009 animated film *"The Princess and the Frog"*, written by Randy Newman. Several versions of the song were recorded for use in different parts of the film and other materials. The song was nominated for Best Original Song at the 82nd Academy Awards but lost to *"The Weary Kind"* from *"Crazy Heart"*.

8. *Gun Shy* is an American sitcom that was shown on CBS from March 15 to April 19, 1983. The series, produced by Walt Disney Productions, was based on its popular comedy-western films: *"The Apple Dumpling Gang"* and *"The Apple Dumpling Gang Rides Again"*.

9. Norman Dale "Buddy" Baker (January 4, 1918 – July 26, 2002) was an American composer who, together with Paul J. Smith, scored many Disney films, such as *"The Apple Dumpling Gang"* in 1975.