# HyperbolicRAG: Enhancing Retrieval-Augmented Generation with Hyperbolic Representations

Linxiao Cao, *Student Member, IEEE,* Ruitao Wang, *Student Member, IEEE,* Jindong Li, *Student Member, IEEE,* Zhipeng Zhou, *Member, IEEE,* Menglin Yang

***Abstract*—Retrieval-augmented generation (RAG) enables large language models (LLMs) to access external knowledge, helping mitigate hallucinations and enhance domain-specific expertise. Graph-based RAG enhances structural reasoning by introducing explicit relational organization that enables information propagation across semantically connected text units. However, these methods typically rely on Euclidean embeddings that capture semantic similarity but lack a geometric notion of hierarchical depth, limiting their ability to represent abstraction relationships inherent in complex knowledge graphs. To capture both fine-grained semantics and global hierarchy, we propose HyperbolicRAG, a retrieval framework that integrates hyperbolic geometry into graph-based RAG. HyperbolicRAG introduces three key designs: (1) a depth-aware representation learner that embeds nodes within a shared Poincaré manifold to align semantic similarity with hierarchical containment, (2) an unsupervised contrastive regularization that enforces geometric consistency across abstraction levels, and (3) a mutual-ranking fusion mechanism that jointly exploits retrieval signals from Euclidean and hyperbolic spaces, emphasizing cross-space agreement during inference. Extensive experiments across multiple QA benchmarks demonstrate that HyperbolicRAG outperforms competitive baselines, including both standard RAG and graph-augmented baselines.**

***Index Terms*—Retrieval-Augmented Generation (RAG), Hyperbolic Space, Hierarchical Modeling, Large Language Models (LLMs).**

## I. Introduction

LARGE language models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks, including question answering, summarization, dialogue generation, and personalization [1], [2], [3]. Despite their strong generalization ability, LLMs inevitably suffer from knowledge staleness and hallucination, as their internal parameters cannot be easily updated with newly emerging facts or domain-specific information [4].

To mitigate these limitations, retrieval-augmented generation (RAG) [5] has emerged as a powerful paradigm that equips LLMs with access to external knowledge bases. By retrieving relevant documents at inference time and conditioning generation on this evidence, RAG systems can provide more up-to-date and contextually grounded responses, thereby reducing reliance on outdated or incomplete parametric knowledge. Building on this idea, graph-based RAG methods, such as G-Retriever [6], GraphRAG [7], LightRAG [8], HippoRAG [9] and HippoRAG2 [10], have organized the retrieved or corpus-level documents into graph structures. In these approaches, documents, entities, and concepts are represented as interconnected nodes linked by semantic or relational edges, enabling more structured access to knowledge. This paradigm enables multi-hop evidence aggregation through explicit graph traversal or message passing, thereby improving reasoning over linked knowledge.

However, graph-based retrieval and reasoning methods typically embed nodes in flat Euclidean spaces, which are ill-suited for representing the hierarchical dependencies that underlie complex knowledge [11], [12]. Consider the query, *"How does long-term tension (chronic stress) lead to weakened immunity?"* A standard dense retriever[1] usually returns passages about broad themes like "health" or "stress," which are superficially relevant yet too generic to reflect the underlying mechanisms. This behavior arises from the hubness inherent in high-dimensional Euclidean embedding spaces [14], [15]: semantically broad concepts occupy central regions that lie close to many queries, causing retrieval to disproportionately favor high-frequency, generic nodes. Consequently, graph traversal treats nodes as if they lie on a single semantic plane, overlooking that "cortisol release" is a specific descendant of "stress" along a causal and ontological hierarchy. In other words, graph-based propagation can connect entities while remaining largely insensitive to the hierarchical geometry that structures complex domains.

To address these challenges, and inspired by findings that human perception structures concepts in tree-like hierarchies where general concepts subsume more specific subconcepts [16], we propose integrating hyperbolic geometry into GraphRAG. Hyperbolic geometry naturally models such structures by encoding semantic depth and containment with minimal distortion [17], [18]: radial distance represents levels of specificity, and the exponential expansion of hyperbolic space accommodates large and deep hierarchies. As illustrated in Fig. 1, in Euclidean space (top), general and specific concepts co-locate on a flat surface, limiting the separation of leaf-leaf nodes and blurring hierarchical boundaries. In contrast, within hyperbolic space (bottom), general concepts are positioned near the center, while specific facts are located

Linxiao Cao, Ruitao Wang, Jindong Li and Menglin Yang are with Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. Email: {lcao950, rwang356, jli839}@connect.hkust-gz.edu.cn, menglin.yang@outlook.com.

Zhipeng Zhou is with Nanyang Technological University, Singapore. Email: zzpustcml@gmail.com

*Corresponding author: Menglin Yang*

[1]Dense retrievers encode queries and documents into continuous vector representations and retrieve results based on embedding similarity [13].

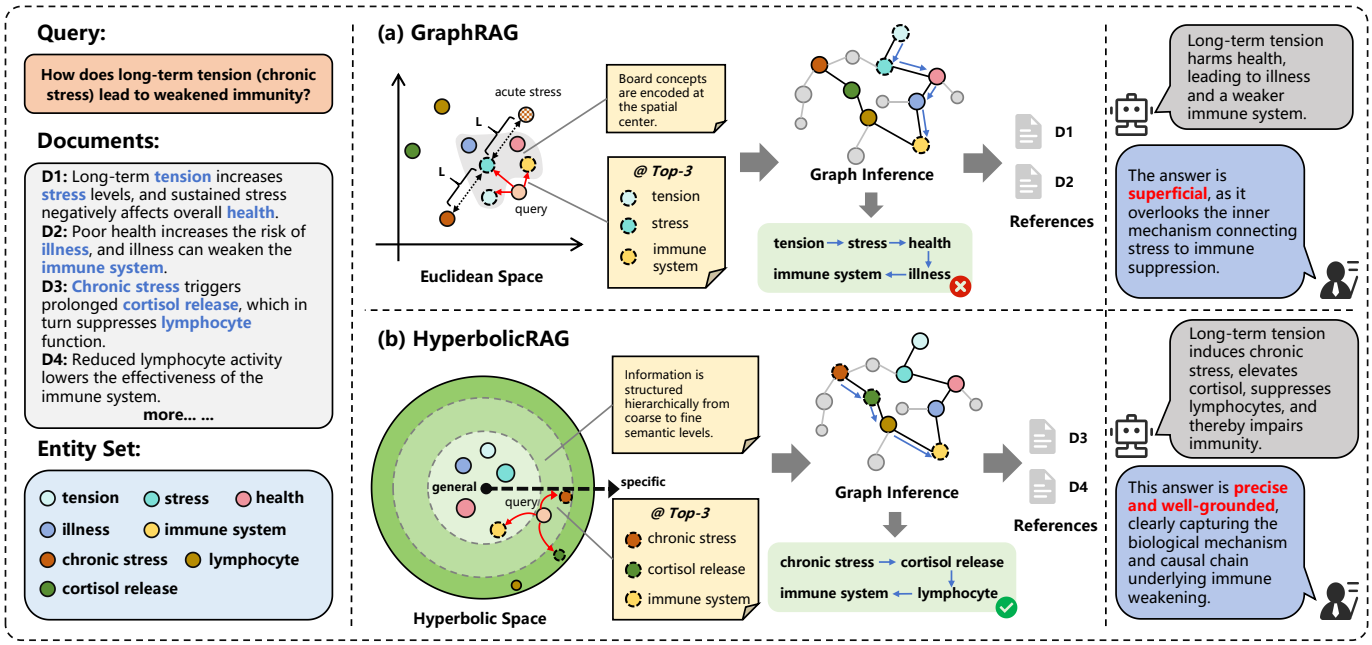


Fig. 1. Comparison of Euclidean and hyperbolic embedding effects on retrieval-augmented multi-hop reasoning. (a) In Euclidean space, embeddings reflect surface-level similarity. General concepts (e.g., stress) act as semantic hubs, making top-$k$ retrieval and graph propagation drift toward broad, generic subgraphs. (b) In hyperbolic space, hierarchical depth is radially encoded: abstract nodes lie near the center, while specific facts align near the boundary. Queries are thus aligned to relevant mechanism nodes (e.g., chronic stress, cortisol release), yielding more precise and causally focused reasoning.

toward the boundary. This arrangement exploits the exponential growth property to preserve hierarchical containment relations.

Building upon this geometric insight, we develop HyperbolicRAG, a hierarchy-aware retrieval framework that integrates hyperbolic geometry into graph-based RAG. HyperbolicRAG introduces three key components. First, it predicts a semantic depth for each textual unit and projects it into a shared Poincaré manifold, where radial distance explicitly encodes hierarchical specificity. Second, a bidirectional alignment loss enforces consistent containment relations between passages and fine-grained factual evidence. Third, at inference time, HyperbolicRAG performs retrieval jointly in Euclidean and hyperbolic spaces and fuses their rankings, thereby balancing local semantic similarity with hierarchical relevance.

Our main contributions are summarized as follows:

- **Hierarchy-aware hyperbolic Representation.** We propose a hierarchy-aware retrieval framework that predicts a scalar depth for each textual unit and performs depth-controlled projection into a shared Poincaré manifold, preserving local semantics while encoding hierarchical structure via radial positions.
- **Bidirectional containment alignment.** We introduce a bidirectional margin-based loss that aligns passages and facts, enabling the model to internalize containment relations across different granularities.
- **Dual-space retrieval fusion.** We develop a dual-space retrieval mechanism that fuses Euclidean and hyperbolic reasoning via mutual-ranking fusion, improving robustness against noisy or overly generic evidence.
- **Competitive results.** Extensive experiments on multiple QA benchmarks demonstrate that HyperbolicRAG consistently outperforms standard RAG and graph-augmented baselines, particularly on multi-hop reasoning tasks.

The remainder of this paper is organized as follows. Section II reviews related work on retrieval-augmented generation, graph-based retrieval, and hyperbolic representation learning. Section III summarizes the necessary background on hyperbolic geometry. Section IV details the proposed framework. Section V presents experimental results and analysis. Finally, Section VI concludes the paper.

## II. Related Work

### A. Retrieval Augment Generation

RAG has emerged as a key paradigm to overcome the static knowledge limitation of LLMs, decoupling parametric knowledge stored in model weights from dynamic retrieval over external corpora [5]. In its original form, RAG relies on dense vector retrieval to fetch topically relevant passages, which are then injected into the LLM context to ground generation. Recent research has explored structured extensions of RAG, with graph-based retrieval emerging as a prominent direction [19], [20], [21]. GraphRAG methods organize textual units into graphs, where edges encode semantic relationships such as entity mentions or inter-entity associations. This structured representation enables relevance propagation across connected nodes, thereby enriching contextual grounding for downstream LLM reasoning.

Despite sharing a common principle, existing GraphRAG variants diverge substantially in their design focus. Hierarchical or community-based approaches, such as Microsoft GraphRAG [7] and LazyGraphRAG [22], construct multi-level

partitions of the corpus (e.g., chapter → section → subsection) to support both local retrieval within communities and global retrieval across them, thereby balancing retrieval precision and corpus coverage. Another line of work centers on structure-optimized designs. LightRAG [8] enriches retrieval by introducing graph-enhanced indexing that combines entity–relation graph construction with key–value profiling, together with a dual-level retrieval strategy to integrate entity-level accuracy with topic-level breadth. GRAG [23] similarly focuses on structural optimization but emphasizes robustness: it employs soft pruning to suppress irrelevant nodes and incorporates graph-aware prompt tuning to reduce retrieval noise. A third direction explores task-adaptive frameworks. StructRAG [24] dynamically incorporates multiple relation types (e.g., part-of, causal) to better support complex reasoning demands, whereas KAG [25] relies on human-curated schemas to construct high-precision, domain-specific knowledge graphs that surpass fully automated extraction pipelines in specialized settings. Yet, these structured RAG variants ignore the underlying geometry where the node representation relies on Euclidean embeddings only, implying that even though nodes are connected relationally, their spatial representation fails to encode hierarchical containment. This geometric limitation motivates us to incorporate hyperbolic geometry into the retrieval process.

### *B. Hyperbolic Representation Learning*

Hyperbolic geometry has become an effective paradigm for modeling hierarchical data, addressing a key limitation of Euclidean spaces: their difficulty in embedding tree-like or nested structures with low distortion [17]. Early foundational works such as Poincaré and Lorentz embeddings [26], [27] and hyperbolic neural networks [28], [29], [30], [31], [32] demonstrate that hyperbolic manifolds provide exponentially expanding representational capacity that naturally aligns with hierarchical and scale-free structures. Besides, hyperbolic geometry has also been applied with hyperbolic metric learning [33], [34], [35], [36], and graph learning, like HyperIMBA [37], HVGNN [38], H2H-GCN [30], H2SeqRec [39]. These empirical advances align with theoretical results showing that hyperbolic spaces enable low-distortion tree embeddings [17], exhibit favorable generalization behavior for hierarchical data [40], and provide high representational efficiency for complex structures [41].

Recent advances are pushing the frontier of hyperbolic geometry into LLMs. Yang et al. [42] demonstrate that token embeddings exhibit inherent hyperbolicity and propose a hyperbolic adaptation for LLMs that enhances downstream reasoning performance. Desai et.al [43] introduce hyperbolic geometry into image-text representation. He et al. [44] propose curvature-adaptive hyperbolic LLM architectures and train hyperbolic LLMs from scratch. Despite these advances, hyperbolic representation learning has rarely been applied to RAG. Our work bridges this gap by integrating hyperbolic geometry into RAG through explicit regularization and dual-space retrieval.

## III. Preliminary

There are several isometric hyperbolic models [45], [46], [47], including the Poincaré ball model, the Lorentz model, Klein model, which show different characteristics but are mathematically equivalent. In this work, we employ the Poincaré ball model, which provides a conformal and analytically convenient formulation of hyperbolic space. Its closed-form geodesics and mappings integrate seamlessly with Euclidean neural encoders, and its radial coordinate furnishes an interpretable measure of hierarchical depth. These features make the Poincaré model well-suited to our depth-controlled projection mechanism and contribute to stable end-to-end training. Although we instantiate our method in the Poincaré ball, it is also compatible with other hyperbolic models. The core geometric concepts of Poincaré ball used in our framework are summarized below.

**Poincaré Ball Model.** The $d$-dimensional Poincaré ball with negative curvature $-c$ ($c > 0$) is defined as $\mathbb{H}^c_d = \{\mathbf{x} \in \mathbb{R}_d : c\|\mathbf{x}\|^2 < 1\}$. It is a conformal model in which angles are preserved and all points lie within a Euclidean ball of radius $1/\sqrt{c}$. The Riemannian metric is $g_{\mathbf{x}} = \lambda_{\mathbf{x}}^2 g^E$, and $\lambda_{\mathbf{x}}^c = \frac{2}{1-c\|\mathbf{x}\|^2}$, where $g^E$ denotes the Euclidean metric. The conformal factor grows rapidly near the boundary, producing the characteristic expansion of hyperbolic space and enabling compact representation of deeply nested hierarchical structures.

**Geodesic and Radial Distances.** Distances in the Poincaré ball follow the induced Riemannian geometry. For points $\mathbf{u}, \mathbf{v} \in \mathbb{H}^c_d$, the hyperbolic geodesic distance is

$$d^c_{\mathbb{H}}(\mathbf{u}, \mathbf{v}) = \frac{1}{\sqrt{c}} \operatorname{arcosh}\left(1 + 2c\,\frac{\|\mathbf{u}-\mathbf{v}\|^2}{(1-c\|\mathbf{u}\|^2)(1-c\|\mathbf{v}\|^2)}\right). \tag{1}$$

This distance reflects both the Euclidean separation of the points and their proximity to the boundary, which causes the metric to expand and naturally induces hierarchical organization: points near the origin correspond to more general concepts, whereas points near the boundary denote more specific ones. The radial distance from the origin is

$$d^c_{\mathbb{H}}(\mathbf{x}, \mathbf{0}) = \frac{1}{\sqrt{c}} \operatorname{arcosh}\left(1 + 2c\,\frac{\|\mathbf{x}\|^2}{1-c\|\mathbf{x}\|^2}\right), \tag{2}$$

which provides a direct geometric measure of hierarchical depth in the embedding space.

**Exponential and Logarithmic Maps.** To couple hyperbolic representations with Euclidean neural encoders, we use the mappings between the tangent space at the origin and the manifold. The tangent space provides a locally Euclidean parameterization that allows standard neural operations to interface with hyperbolic geometry.

The exponential map sends a tangent vector $\mathbf{v} \in T_{\mathbf{0}}\mathbb{H}^c_d$ onto the manifold:

$$\exp^c_{\mathbf{0}}(\mathbf{v}) = \tanh\left(\sqrt{c}\,\|\mathbf{v}\|\right)\frac{\mathbf{v}}{\sqrt{c}\,\|\mathbf{v}\|}. \tag{3}$$

Its inverse, the logarithmic map, returns a point $\mathbf{u} \in \mathbb{H}^c_d$ to the tangent space:

$$\log^c_{\mathbf{0}}(\mathbf{u}) = \frac{1}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\,\|\mathbf{u}\|\right)\frac{\mathbf{u}}{\|\mathbf{u}\|}. \tag{4}$$

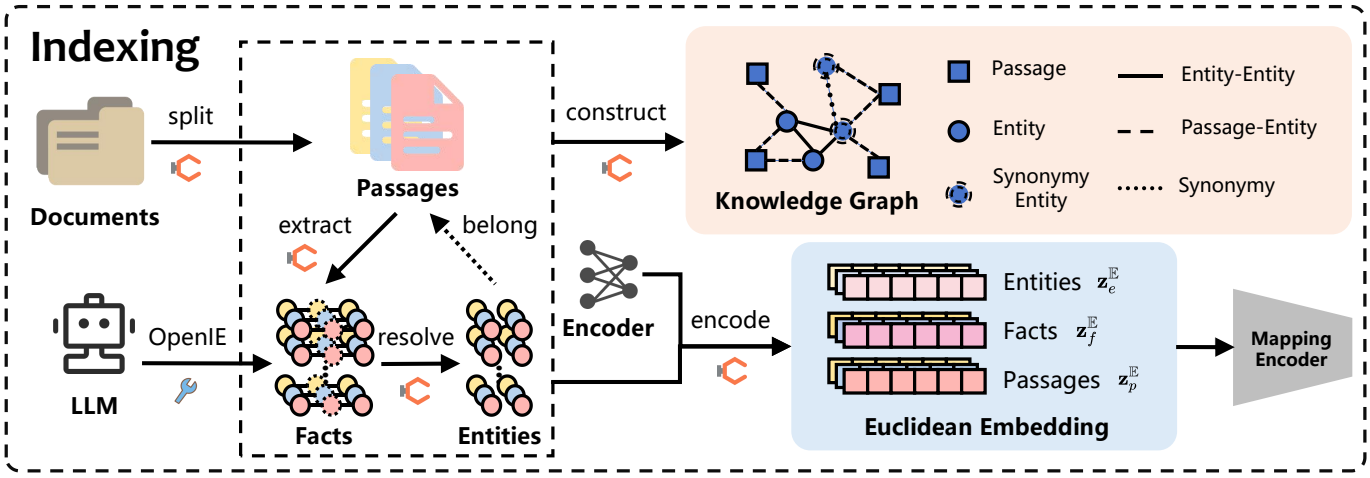


Fig. 2. **Indexing pipeline.** Given a document collection, the framework first performs *chunking* to obtain passages, from which an *OpenIE extractor* derives relational triples and normalized entity mentions. Passages, entities, and facts are then encoded into dense vectors using a pretrained encoder. Finally, a heterogeneous knowledge graph is constructed by linking (i) entity–entity pairs co-occurring in triples, (ii) passage–entity pairs grounded in text, and (iii) synonymy links between semantically similar entities.

Together, these maps provide a smooth interface between Euclidean and hyperbolic computations, enabling end-to-end training while preserving the hierarchical structure encoded by the manifold.

## IV. Methodology

### A. Overview

Our framework presents a dual-space retrieval framework that seamlessly incorporates hierarchical structural knowledge into RAG. The pipeline is organized into three complementary stages: *Indexing Process* (illustrated in Fig. 2), *Hierarchical Enhancement* (illustrated in Fig. 3) and *Dual-space Retrieval Process* (illustrated in Fig. 4).

### B. Indexing Process

The goal of the indexing phase is to transform the raw corpus $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ into a structured representation that supports semantic retrieval and hierarchical reasoning. This is achieved by constructing a heterogeneous knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\text{edge}})$, where passages and entities constitute the node set and their relationships are encoded through extracted factual and contextual connections. The overall process consists of four sequential stages: (1) *document chunking* to segment the corpus into coherent retrieval units, (2) *relational extraction* to identify canonical entities and their interrelations, (3) *representation learning* to obtain semantic embeddings as node features, and (4) *graph construction* to define the topological structure of $\mathcal{G}$.

*1) Document chunking:* Each document $D_i$ is divided into shorter, semantically coherent segments, referred to as *passages*. This segmentation balances retrieval granularity and computational efficiency: passages should preserve sufficient local context for meaningful retrieval while avoiding unnecessary redundancy. Formally, each document $D_i$ is decomposed into passages $D_i \to \{p_{i,1}, p_{i,2}, \ldots, p_{i,M_i}\}$ with $p_{i,j} \in \mathcal{P}$, where $p_{i,j}$ denotes the $j$-th passage extracted from $D_i$, $M_i$ is the number of passages derived from $D_i$, and $\mathcal{P}$ denotes the global set of all passages. Each passage later becomes a textual node in $\mathcal{G}$, serving as the retrieval unit to which entities and relational evidence are anchored.

*2) Relational extraction:* To build a relationally grounded view of the corpus, we employ a two-stage extraction pipeline guided by an LLM. First, the model identifies salient entities from each passage $p \in \mathcal{P}$ to establish the entity context. Conditioned on these entities, it then extracts relational triples that describe their interactions:

$$\mathcal{T}(p) = \{(e_s, r, e_o) \mid e_s, e_o \in \text{Entities}(p),\ r \in \text{Relations}(p)\}. \tag{5}$$

Each extracted subject or object entity $(e_s, e_o)$ is incorporated into the global entity set $\mathcal{E}$ if not already present, and every triple $(e_s, r, e_o)$ is recorded as a fact $f \in \mathcal{F}$. Facts are not represented as graph nodes; instead, they act as relational annotations linking passages and canonical entities. Accordingly, the node set of the graph is $\mathcal{V} = \mathcal{P} \cup \mathcal{E}$, where passages and entities form the structural backbone of $\mathcal{G}$, while facts enrich their semantic connectivity.

*3) Representation learning:* After identifying the structural elements, we encode their semantic content into dense vector representations using a pretrained language model encoder, denoted $\text{Enc}(\cdot)$. These embeddings serve as *node features* and provide the semantic foundation for later hierarchy-aware refinement. Specifically, we obtain three types of embeddings:

$$\begin{aligned} \mathbf{z}_p^{\mathbb{E}} &= \text{Enc}(p), & p \in \mathcal{P}, \\ \mathbf{z}_e^{\mathbb{E}} &= \text{Enc}(e), & e \in \mathcal{E}, \\ \mathbf{z}_f^{\mathbb{E}} &= \text{Enc}(f), & f \in \mathcal{F}. \end{aligned} \tag{6}$$

Passage embeddings $\mathbf{z}_p^{\mathbb{E}}$ capture contextual semantics at the text level; entity embeddings $\mathbf{z}_e^{\mathbb{E}}$ encode concept-level meaning aggregated across mentions; and fact embeddings $\mathbf{z}_f^{\mathbb{E}}$ represent relational semantics between entities. Importantly, these embeddings do not define the graph topology; rather, they serve as semantic attributes that enable similarity computation and

inform the subsequent hierarchy-aware embedding enhancement.

*4) Graph construction:* The final step of indexing is to establish edges $\mathcal{E}_{\text{edge}}$ in $\mathcal{G}$ to encode complementary structural relationships between nodes. These edges are categorized into three types, each designed to capture a specific type of semantic connection:

- **Entity–Entity edges**: Each fact triple $(s, r, o)$ (with $s$ and $o$ normalized to canonical entities $e_s, e_o \in \mathcal{E}$) induces an edge between $e_s$ and $e_o$. To quantify the strength of this relationship, we increment the edge weight by the co-occurrence frequency of $e_s$ and $e_o$ across all facts. Formally, the edge weight is updated as $w(e_s, e_o) \leftarrow w(e_s, e_o) + 1$, where $w(e_s, e_o)$ denotes the weight of the edge between $e_s$ and $e_o$. These edges capture local factual relations (e.g., "lung cancer" $\xrightarrow{\text{causes}}$ "chest pain") and support the propagation of evidence among semantically related entities during retrieval.
- **Passage–Entity edges**: Each passage $p \in \mathcal{P}$ is connected to all entities $e \in \text{Entities}(p)$ (i.e., all entity mentions in $p$ after normalization). Formally, we add an edge $(p, e)$ to $\mathcal{E}_{\text{edge}}$ for every entity $e \in \text{Entities}(p)$, i.e., $(p, e) \in \mathcal{E}_{\text{edge}}$ for all $e \in \text{Entities}(p)$. These edges anchor entities to their original textual context and allow entity-level signals (such as relevance scores of query-related entities) to influence passage scoring, thereby enhancing the model's ability to capture context-aware connections.
- **Synonymy edges**: To address lexical variability (i.e., different surface forms of the same entity, such as "U.S." vs. "United States" or "COVID-19" vs. "coronavirus disease 2019"), we connect entities whose embeddings exceed a predefined cosine similarity threshold $\tau_{\text{syn}}$. Formally, we add an edge $(e, e')$ to $\mathcal{E}_{\text{edge}}$ whenever $\cos(\mathbf{z}_e^{\mathbb{E}}, \mathbf{z}_{e'}^{\mathbb{E}}) \geq \tau_{\text{syn}}$, where $\cos(\cdot, \cdot)$ denotes cosine similarity and $e, e' \in \mathcal{E}$ are distinct entities. This construction strengthens connectivity among semantically equivalent entities expressed with different surface forms and alleviates graph fragmentation arising from lexical variation in heterogeneous graphs.

The resulting heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\text{edge}})$, together with the cached embeddings $\{\mathbf{z}_p^{\mathbb{E}} \mid p \in \mathcal{P}\}, \{\mathbf{z}_e^{\mathbb{E}} \mid e \in \mathcal{E}\}, \{\mathbf{z}_f^{\mathbb{E}} \mid f \in \mathcal{F}\}$, forms a compact yet expressive index, supporting later hierarchy-aware projection and dual-space reasoning.

### C. Hierarchical Enhancement

The heterogeneous graph constructed during indexing captures explicit relational connectivity but remains geometrically flat. Euclidean embeddings encode local semantic similarity but fail to capture how broad passages encompass fine-grained facts. This flat geometry causes relevance propagation to drift toward generic or high-degree nodes, leading to unstable multi-hop reasoning and noisy retrieval.

We address this by refining node embeddings. Specifically, Euclidean embeddings of passages, entities, and facts are projected into a shared hyperbolic space $\mathbb{H}_d^c$, whose negative curvature naturally models tree-like hierarchies. Radial distance from the origin encodes hierarchical depth: general passages are placed near the center, while specific facts are pushed toward the boundary. A learned depth predictor assigns each element a scalar specificity score that determines its radial position. This hierarchy-aware embedding reshapes the query–node similarity distribution, focusing propagation on relevant, hierarchy-consistent subgraphs without modifying the underlying topology.

The following sections describe (1) the *hyperbolic projection mechanism* and (2) *unsupervised optimization* that enforce containment consistency between passages and facts.

*1) Hyperbolic projection:* Given Euclidean embeddings $\mathbf{z}_v^{\mathbb{E}}$ for passages, entities, and facts, we project them into the hyperbolic space $\mathbb{H}_d^c$ to obtain hierarchy-aware representations $\mathbf{z}_v^{\mathbb{H}}$. The projection process integrates semantic preservation with hierarchical control and proceeds as follows.

*a) Hierarchy feature extraction:* While Euclidean embeddings $\mathbf{z}_v^{\mathbb{E}}$ encode topical similarity, they lack cues that distinguish hierarchical granularity. To capture such structure, we firstly apply a nonlinear transformation $\phi : \mathbb{E}^d \to \mathbb{E}^{d'}$, yielding $\mathbf{u}_v = \phi(\mathbf{z}_v^{\mathbb{E}})$, where $\mathbf{u}_v$ captures hierarchy-related features.

*b) Depth prediction:* A type-specific predictor ($\psi_{\text{pass}}$, $\psi_{\text{fact}}$, $\psi_{\text{ent}}$) maps the hierarchical signal $\mathbf{u}_v$ to a depth score $d_v \in [0, 1]$ via $d_v = \psi_{\text{mode}(v)}(\mathbf{u}_v)$, where smaller scores correspond to more general nodes and larger scores to more specific ones. These depth values later determine the radial placement of nodes in the hyperbolic space.

*c) Feature fusion with gating:* To jointly preserve semantic and hierarchical information, we fuse $\mathbf{z}_v^{\mathbb{E}}$ and $\mathbf{u}_v$ via a gating mechanism:

$$\tilde{\mathbf{z}}_v^{\mathbb{E}} = \rho([\mathbf{z}_v^{\mathbb{E}} \parallel \mathbf{u}_v]), \tag{7}$$

$$\mathbf{m}_v = \sigma(W_g \tilde{\mathbf{z}}_v^{\mathbb{E}}), \tag{8}$$

$$\mathbf{z}_v^* = \mathbf{m}_v \odot \mathbf{z}_v^{\mathbb{E}} + (1 - \mathbf{m}_v) \odot \tilde{\mathbf{z}}_v^{\mathbb{E}}, \tag{9}$$

where $\sigma(\cdot)$ is the element-wise sigmoid, $W_g \in \mathbb{E}^{d \times d}$ is the learnable weight matrix of the gating layer, and $\mathbf{m}_v \in (0, 1)^d$ are per-dimension gates. The result $\mathbf{z}_v^* \in \mathbb{E}^d$ is the refined Euclidean embedding used in the subsequent depth-alignment step.

*d) Radial depth alignment:* To translate the predicted depth $d_v$ into spatial structure, we regulate the $L_2$ norm of the refined embedding $\mathbf{z}_v^*$ by enforcing $\|\hat{\mathbf{z}}_v^{\mathbb{E}}\| = \alpha + \beta d_v$, where $\alpha, \beta > 0$ and $\alpha + \beta \leq 1$ to ensure all vectors remain inside the Poincaré ball. The aligned embedding is then obtained by:

$$\hat{\mathbf{z}}_v^{\mathbb{E}} = \frac{\alpha + \beta d_v}{\|\mathbf{z}_v^*\|} \mathbf{z}_v^*. \tag{10}$$

This step preserves semantic direction while encoding hierarchical depth as radial distance, where general nodes occupy inner regions and specific ones are placed closer to the boundary.

*e) Mapping to hyperbolic space:* The aligned Euclidean vector $\hat{\mathbf{z}}_v^{\mathbb{E}}$ is then projected into the Poincaré ball $\mathbb{H}_d^c = \{\mathbf{x} \in \mathbb{E}^d \mid \|\mathbf{x}\| < 1\}$ via the exponential map at the origin:

$$\mathbf{z}_v^{\mathbb{H}} = \exp_{\mathbf{0}}^c(\hat{\mathbf{z}}_v^{\mathbb{E}}). \tag{11}$$

This mapping preserves the semantic direction of the Euclidean embedding while converting its norm into a radius

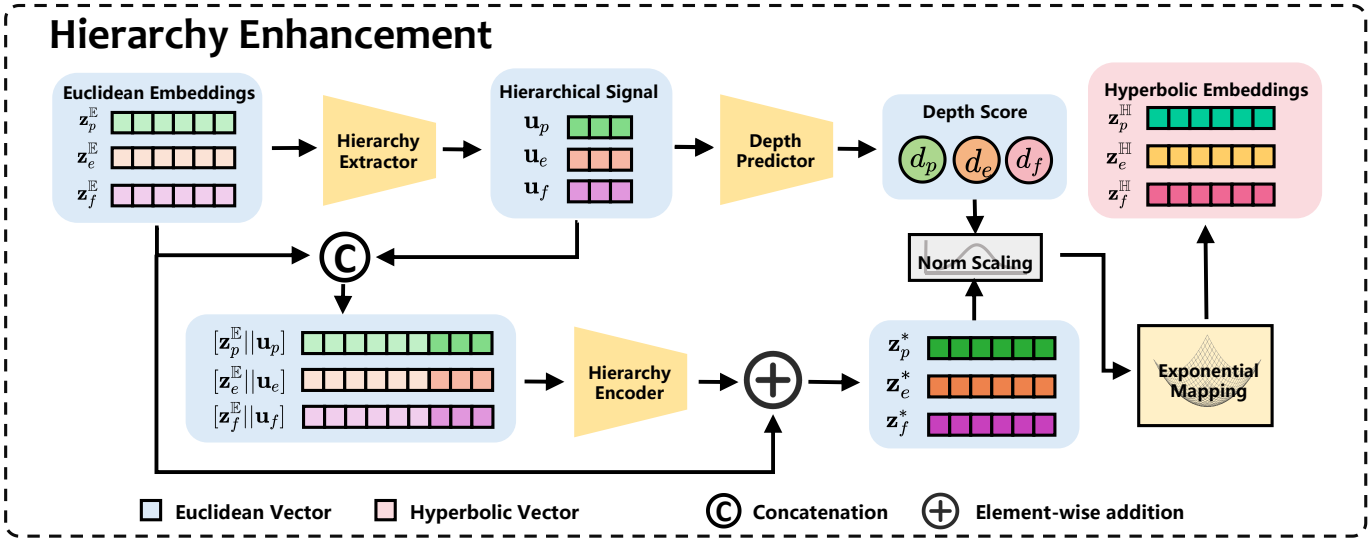


Fig. 3. **Overview of the hierarchical enhancement process.** Given Euclidean embeddings of passages, entities, and facts, the model first extracts a *hierarchical signal* $\mathbf{u}_v$. This signal serves two roles: it is concatenated with the original semantic embedding $\mathbf{z}_v^{\mathbb{E}}$ to form an enhanced Euclidean representation enriched with hierarchical cues, and it is also used to predict a depth score $d_v$ that reflects the relative granularity of each node. The predicted depth then regulates a radial rescaling of the enhanced embedding, assigning smaller norms to more generic concepts and larger norms to more fine-grained evidence. Finally, the depth-aligned vectors are projected into the Poincaré ball via the exponential map, producing hyperbolic embeddings $\mathbf{z}_v^{\mathbb{H}}$ that jointly encode semantic similarity and hierarchical structure.

on the Poincaré ball. Such curvature-aware embedding offers greater representational efficiency—capturing both local semantic similarity and global hierarchy structure.

*2) Unsupervised optimization:* The hyperbolic projection produces geometry-aware embeddings that encode hierarchical depth, yet it does not explicitly constrain the spatial relationships between passages and their contained facts. To enforce such containment consistency, we introduce a pair of margin-based contrastive objectives operating in both passage-to-fact and fact-to-passage directions.

*a) Passage-to-Fact alignment:* For each passage $p \in \mathcal{P}$, let $\mathcal{F}(p)$ denote the set of facts extracted from it. Each positive pair $(p, f^+)$ is accompanied by a randomly sampled negative fact $f^-$ not associated with $p$. We encourage $p$ to be closer to $f^+$ than to $f^-$ in hyperbolic space by at least a margin $\gamma$:

$$\mathcal{L}_{p\to f} = \sum_{f\in\mathcal{F}(p)} \left[ d_{\mathbb{H}}^c(\mathbf{z}_p^{\mathbb{H}}, \mathbf{z}_{f^+}^{\mathbb{H}}) - d_{\mathbb{H}}^c(\mathbf{z}_p^{\mathbb{H}}, \mathbf{z}_{f^-}^{\mathbb{H}}) + \gamma \right]_+, \quad (12)$$

where $d_{\mathbb{H}}^c(\cdot,\cdot)$ denotes hyperbolic distance and $[\cdot]_+ = \max(0,\cdot)$ ensures non-negative loss.

*b) Fact-to-Passage alignment:* Symmetrically, for each fact $f \in \mathcal{F}$, let $\mathcal{P}(f)$ denote its supporting passages. Each positive pair $(f, p^+)$ is contrasted with a negative passage $p^-$ not containing $f$, enforcing the reverse containment:

$$\mathcal{L}_{f\to p} = \sum_{p\in\mathcal{P}(f)} \left[ d_{\mathbb{H}}^c(\mathbf{z}_f^{\mathbb{H}}, \mathbf{z}_{p^+}^{\mathbb{H}}) - d_{\mathbb{H}}^c(\mathbf{z}_f^{\mathbb{H}}, \mathbf{z}_{p^-}^{\mathbb{H}}) + \gamma \right]_+. \quad (13)$$

The dual alignment jointly optimizes hierarchical consistency in both directions: passages act as semantic containers that aggregate multiple fine-grained facts, while facts serve as evidence grounding for passages. This bidirectional constraint stabilizes the learned geometry and prevents degenerate alignment (e.g., all nodes collapsing toward the boundary).

### *D. Dual-space Retrieval Process*

Building upon the retrieval framework of [10], we extend it to a dual-space setting that jointly exploits the complementary strengths of Euclidean and hyperbolic geometries. Specifically, our retrieval module introduces two key enhancements: 1) it performs independent relevance propagation in both the *Euclidean space*, which excels at modeling local semantic similarity and the *Hyperbolic space* which naturally preserves hierarchical containment; 2) it integrates their results through a mutual-ranking fusion strategy that emphasizes cross-space consistency while preventing interference during propagation.

This dual-space design enables retrieval to simultaneously benefit from fine-grained topical alignment and geometry-aware reasoning over hierarchical relations. An overview of the complete dual-space retrieval workflow is shown in Fig. 4

*1) Euclidean branch (semantic similarity-based retrieval):* The Euclidean branch aims to model fine-grained semantic similarity between the query and corpus elements. Its operation proceeds in three stages.

*a) Signal initialization:* Given a query $q$, we encode it into the Euclidean space using the same pretrained encoder $\mathrm{Enc}(\cdot)$ as in the indexing stage, obtaining the query embedding $\mathbf{z}_q^{\mathbb{E}}$. Two complementary types of initial relevance signals are derived:

- **Fact-level signals.** We compute cosine similarity between $\mathbf{z}_q^{\mathbb{E}}$ and cached fact embeddings $\mathbf{z}_f^{\mathbb{E}}$ from the index. The top-$k$ most similar facts are selected as initial evidence. Their scores are propagated to the corresponding subject and object entities $(e_s, e_o)$ and normalized by each entity's number of associated passages to mitigate degree bias.
- **Passage-level signals.** We directly compute cosine similarity between $\mathbf{z}_q^{\mathbb{E}}$ and passage embeddings $\mathbf{z}_p^{\mathbb{E}}$, forming

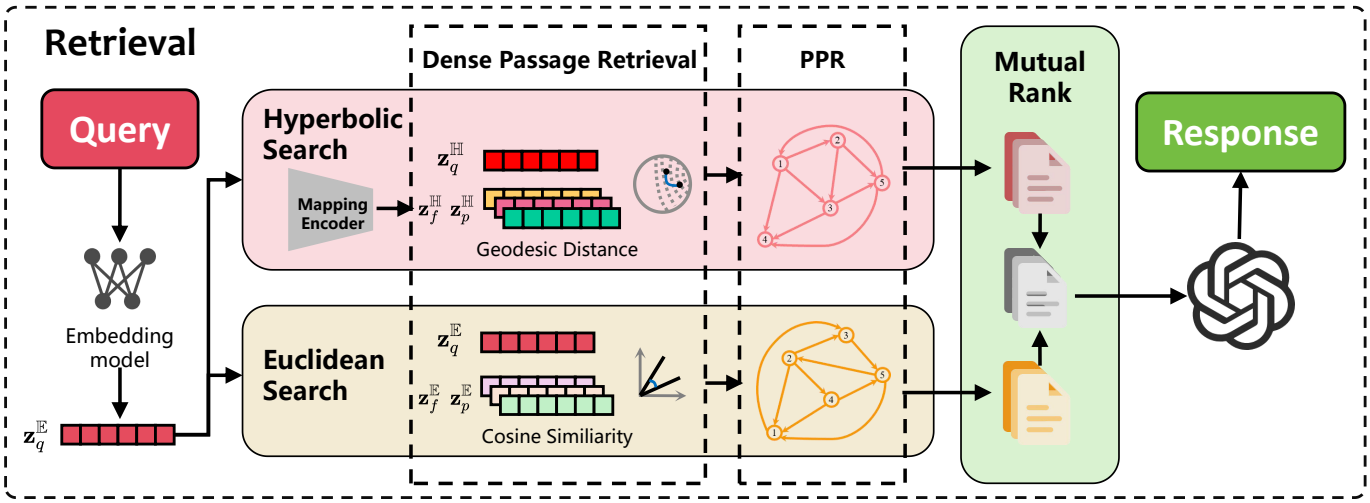


Fig. 4. **Illustration of the dual-space retrieval framework.** The query is processed in parallel Euclidean and hyperbolic spaces. Each branch computes query–fact similarities through different ways, propagates them to entities, and combines them with direct query–passage priors to form a seed distribution for PPR on the passage–entity graph, yielding space-specific rankings ($\mathcal{R}^{\mathbb{E}}$ and $\mathcal{R}^{\mathbb{H}}$). A mutual-ranking fusion then favors passages consistently ranked high in both spaces, balancing Euclidean semantic similarity and hyperbolic hierarchical structure for robust retrieval.

passage-level priors that capture topical alignment.

*b) Seed distribution construction:* The signals are merged into a unified seed distribution $\mathbf{s}_q^{\mathbb{E}}$, where each entry corresponds to the initial relevance weight of a node in the heterogeneous graph. This distribution serves as the restart distribution for the Personalized Page Rank (PPR) [48] process, ensuring propagation is centered on query-relevant regions.

*c) Graph propagation:* We apply PPR over the passage–entity graph using $\mathbf{s}_q^{\mathbb{E}}$ as the seed vector, computing $\pi_q^{\mathbb{E}} = \alpha \mathbf{s}_q^{\mathbb{E}} + (1-\alpha)\,\pi_q^{\mathbb{E}} W$, where $W$ is the row-normalized adjacency matrix and $\alpha \in (0,1)$ is the restart probability controlling the balance between local focus and global diffusion. After convergence, the stationary distribution $\pi_q^{\mathbb{E}}$ yields passage-level relevance scores, which are sorted to form the Euclidean ranking list $\mathcal{R}^{\mathbb{E}}$.

*2) Hyperbolic branch (hierarchy-aware retrieval):* The hyperbolic branch follows the same computational structure as the Euclidean branch, but performs operations in a non-Euclidean manifold, thereby contributing complementary hierarchical signals.

*a) Query projection and signal initialization:* The Euclidean query embedding $\mathbf{z}_q^{\mathbb{E}}$ is projected into the hyperbolic space $\mathbb{H}_d^c$ via the trained projection module (Section IV-C), producing $\mathbf{z}_q^{\mathbb{H}}$.

- **Fact-level signals.** We measure similarity as the *negative hyperbolic geodesic distance* between $\mathbf{z}_q^{\mathbb{H}}$ and cached fact embeddings $\mathbf{z}_f^{\mathbb{H}}$, selecting the top-$k$ closest facts as initial evidence. Their scores are propagated to associated entities $(e_s, e_o)$ and normalized by entity degree, identical to the Euclidean branch.
- **Passage-level signals.** Similarly, negative hyperbolic distance between $\mathbf{z}_q^{\mathbb{H}}$ and passage embeddings $\mathbf{z}_p^{\mathbb{H}}$ provides hierarchy-aware passage priors, emphasizing fine-to-coarse structural proximity.

*b) Seed distribution and propagation.:* The entity and passage-level signals are merged into a hyperbolic seed distribution $\mathbf{s}_q^{\mathbb{H}}$, which serves as the restart distribution for the PPR process, yielding $\pi_q^{\mathbb{H}} = \alpha \mathbf{s}_q^{\mathbb{H}} + (1-\alpha)\,\pi_q^{\mathbb{H}} W$, where $W$ and $\alpha$ are shared with the Euclidean branch to ensure consistent propagation dynamics. After convergence, the stationary distribution $\pi_q^{\mathbb{H}}$ defines the hyperbolic relevance scores for passages, producing the ranking list $\mathcal{R}^{\mathbb{H}}$.

*3) Mutual-ranking fusion (integrating complementary signals):* To combine the Euclidean ($\mathcal{R}^{\mathbb{E}}$) and hyperbolic ($\mathcal{R}^{\mathbb{H}}$) rankings, we employ a mutual-ranking fusion scheme that emphasizes passages consistently favored by both spaces, which mitigates noise from either single space and amplifies robust signals. The fusion process has three key steps:

1) **Reciprocal rank calculation.** For each passage $p$, we convert its rank in each list into a reciprocal-rank score, $s_{\mathbb{E}}(p) = 1/(\mathrm{rank}_{\mathbb{E}}(p)+1)$ and $s_{\mathbb{H}}(p) = 1/(\mathrm{rank}_{\mathbb{H}}(p)+1)$, where $\mathrm{rank}_{\mathbb{E}}(p)$ and $\mathrm{rank}_{\mathbb{H}}(p)$ denote the rank of $p$ in $\mathcal{R}^{\mathbb{E}}$ and $\mathcal{R}^{\mathbb{H}}$, respectively.
2) **Consistency bonus calculation.** We assign an additional bonus $b(p)$ to passages that appear in both rankings, rewarding cross-space consistency. The bonus is computed as $b(p) = 1/(\mathrm{rank}_{\mathbb{E}}(p) + \mathrm{rank}_{\mathbb{H}}(p) + 2)$, which gives higher values to passages that simultaneously achieve strong ranks in both lists.
3) **Hybrid score computation.** The final hybrid score for each passage $p$ is computed as $s_{\mathrm{hyb}}(p) = \big(s_{\mathbb{E}}(p) + s_{\mathbb{H}}(p)\big)\,(1 + b(p))$, and passages are subsequently re-ranked in descending order of $s_{\mathrm{hyb}}(p)$ to obtain the final retrieval result.

This late-fusion design ensures that Euclidean and hyperbolic retrieval remain independent during graph propagation, while the mutual-ranking scheme explicitly leverages cross-space consistency to enhance retrieval robustness and precision.

TABLE I
SUMMARY OF DATASET INFORMATION, EXTRACTION RESULTS, AND GRAPH STATISTICS.

| | NQ | PopQA | MuSiQue | 2Wiki | HotpotQA |
|---|---|---|---|---|---|
| ***Basic Dataset Statistic*** | | | | | |
| Number of Queries | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 |
| Number of Passages | 9,633 | 8,676 | 11,656 | 6,119 | 9,811 |
| ***Information Extraction Statistic*** | | | | | |
| Number of Facts | 115,243 | 112,990 | 140,739 | 68,840 | 129,997 |
| Number of Entities | 62,234 | 72,050 | 85,274 | 44,003 | 81,200 |
| ***Knowledge Graph Statistic*** | | | | | |
| Number of Nodes | 71,867 | 80,726 | 96,944 | 50,123 | 91,011 |
| Number of Edges | 990,057 | 954,528 | 1,399,262 | 726,330 | 1,246,677 |

## V. EVALUATION

### A. Datasets

To evaluate the effectiveness of our dual-space retrieval framework in supporting multi-hop reasoning, we follow existing work [10] categorizing datasets into two challenge types:

1) **Simple QA.** This category primarily evaluates the ability to recall and retrieve factual knowledge accurately. We randomly select 1,000 queries from Natural Questions (NQ) [49], which contains real user questions covering diverse topics. Additionally, we select 1,000 queries from PopQA [50], derived from the December 2021 Wikipedia dump. Both datasets provide straightforward QA pairs suitable for assessing single-hop retrieval performance. Notably, PopQA is more entity-centric, with entities occurring less frequently than in NQ, making it particularly useful for evaluating entity recognition and retrieval in simple QA tasks.
2) **Multi-hop QA.** Multi-hop datasets require the model to connect multiple pieces of information to answer a query, testing associative reasoning capabilities. We sample 1,000 queries from MuSiQue [51], 2WikiMultihopQA [52], and HotpotQA [53], following the setup in HippoRAG [10]. For all multi-hop datasets, long-form contexts are segmented into shorter passages while maintaining the same RAG setup, allowing our retrieval framework to aggregate evidence across multiple passages.

The statistics of the sampled datasets are summarized in Table I. Together, these datasets provide a comprehensive evaluation of retrieval models on factual memory, multi-hop reasoning, and discourse-level comprehension.

### B. Baselines

We evaluate our framework against three categories of baselines:

1) **Simple retrieval methods.** BM25 [54]: A classical lexical matching baseline. Contriever [55] and GTR [56]: Popular dense embedding retrievers that rely solely on Euclidean semantic similarity.
2) **Large pre-trained embedding models.** These baselines use state-of-the-art 7B-scale embedding models that achieve strong performance on the BEIR benchmark [57]: Alibaba-NLP/GTE-Qwen2-7B-Instruct [58], GritLM/GritLM-7B [59], nvidia/NV-Embed-v2 [60]. They provide strong semantic representations and serve as a competitive reference for dense retrieval performance.
3) **Structure-augmented RAG.** These methods leverage graph or hierarchical structures to improve multi-hop reasoning: RAPTOR [61]: Organizes the corpus hierarchically based on semantic similarity. GraphRAG [7] and LightRAG [8]: Use knowledge graphs to propagate relevance and summarize high-level concepts. HippoRAG [9]: Integrates graph-based knowledge using PPR rather than summarization. HippoRAG2 [10]: An improved variant of HippoRAG that refines both graph-based retrieval and memory aggregation, yielding stronger performance on multi-hop QA benchmarks.

### C. Metrics

We adopt two complementary sets of metrics to evaluate retrieval and downstream QA performance.

- **Retrieval evaluation.** Following HippoRAG (Gutiérrez et al., 2024), we report Passage Recall@5, which measures whether the gold evidence passages appear among the top-5 retrieved candidates.
- **QA evaluation.** For end-to-end question answering, we adopt the token-level evaluation protocol introduced in MuSiQue [51]. We report both the Exact Match (EM) and F1 scores between the predicted answer span and the ground-truth answer. EM measures the exact string match accuracy, while F1 balances precision and recall by capturing the overlap of tokens between prediction and reference.

Together, these metrics provide a comprehensive view: retrieval recall emphasizes evidence coverage, while EM and F1 measures final answer quality.

### D. Implementation Details

We follow the experimental setup of HippoRAG2 [10] to ensure fair comparison. Specifically, we use Llama-3.3-70B-Instruct [62] as the extraction model for both NER and OpenIE, and as the triple filtering model. For retrieval, we adopt NV-Embed-v2 [60] as the embedding model. For QA generation, we report the results of feeding the top-5 retrieved passages as context to an LLM (i.e., Llama-3.3-70B-Instruct).

TABLE II
COMPARISON OF RETRIEVAL METHODS IN TERMS OF RECALL@5 (%) ACROSS SIMPLE QA AND MULTI-HOP QA DATASETS.

| Retrieval Methods | Simple QA | | Multi-Hop QA | | | Avg |
|---|---|---|---|---|---|---|
| | NQ | PopQA | MuSiQue | 2Wiki | HotpotQA | |
| ***Simple Baselines*** | | | | | | |
| BM25 | 56.1 | 35.7 | 43.5 | 65.3 | 74.8 | 55.1 |
| Contriever | 54.6 | 43.2 | 46.6 | 57.5 | 75.3 | 55.4 |
| GTR (T5-base) | 63.4 | 49.4 | 49.1 | 67.9 | 73.9 | 60.7 |
| ***Large Embedding Models*** | | | | | | |
| GTE-Qwen2-7B-Instruct | 74.3 | 50.6 | 63.6 | 74.8 | 89.1 | 70.5 |
| GritLM-7B | 76.6 | 50.1 | 65.9 | 76.0 | 92.4 | 72.2 |
| NV-Embed-v2-7B | 75.4 | 51.0 | 69.7 | 76.5 | 94.5 | 73.4 |
| ***Structure-Augmented RAG*** | | | | | | |
| RAPTOR | 68.3 | 48.7 | 57.8 | 66.2 | 86.9 | 65.6 |
| HippoRAG | 44.4 | **53.8** | 53.2 | 90.4 | 77.3 | 63.8 |
| HippoRAG2 | 78.0 | 51.7 | 74.7 | 90.4 | 96.2 | 78.2 |
| **HyperbolicRAG** | **78.5** | 51.9 | **76.2** | **92.1** | **96.3** | **79.0** |

All hyperparameters (e.g., damping factor in PPR, retrieval cutoffs) follow the default values from HippoRAG2 unless otherwise stated.

### *E. Experimental Results*

We evaluate HyperbolicRAG through a comprehensive set of experiments to assess its retrieval effectiveness, end-to-end QA performance, component-wise contributions, and model-agnostic robustness. Across all results, our method is highlighted in gray, the best result is marked in **bold**, and the second-best result is underlined.

*1) Information Extraction Results and Graph Statistics:* Before evaluating retrieval performance, we first report the information extraction and graph construction results from the indexing stage. Each passage is converted into structured fact triples using Llama-3.3-70B-Instruct, forming the factual backbone of the heterogeneous passage–entity graph. Based on these facts and entities, we construct the passage–entity knowledge graphs, where passages and entities serve as nodes and edges represent factual or co-occurrence relations. The overall extraction results and graph statistics are summarized in Table I.

*2) Retrieval Results:* We evaluate retrieval performance using Recall@5. Table II reports the results on both simple and multi-hop QA datasets. HyperbolicRAG achieves the highest overall Recall@5 of 79.0%, outperforming all Euclidean and structure-augmented baselines. Compared with the strongest Euclidean retriever, NV-Embed-v2-7B (73.4%), it delivers a 5.6% absolute improvement, demonstrating the advantage of modeling hierarchical organization beyond surface-level similarity. Within structure-augmented methods, HyperbolicRAG slightly outperforms HippoRAG2 (78.2%), indicating that hyperbolic geometry provides complementary benefits even for advanced graph-based retrieval frameworks. The gains are most evident on multi-hop datasets such as 2Wiki (92.1% vs. 90.4%) and MuSiQue (76.2% vs. 74.7%), where reasoning requires integrating multiple entities and relations. On simpler datasets such as NQ and PopQA, improvements are smaller, confirming that the hyperbolic formulation enhances robustness without overfitting to specific structural patterns. Overall, these findings demonstrate that explicitly modeling relational hierarchies in hyperbolic space mitigates the hubness bias inherent in Euclidean embeddings and leads to more precise and context-aware retrieval.

*3) Generation Results:* Beyond retrieval effectiveness, we further evaluate end-to-end QA performance using EM and token-level F1 on answers generated from the top five retrieved passages. Table III summarizes the results. HyperbolicRAG achieves the highest overall performance, with an average of 51.4% EM and 63.3% F1, outperforming both Euclidean and structure-augmented baselines. Compared with the strongest competitors, HippoRAG2 (51.0% / 62.7%) and NV-Embed-v2-7B (49.0% / 60.0%), HyperbolicRAG demonstrates consistent gains across all datasets. The improvement is particularly pronounced on multi-hop QA benchmarks such as MuSiQue (39.5% / 50.6%) and 2Wiki (65.5% / 72.3%), where reasoning requires integrating multiple entities and factual relations. In these tasks, hyperbolic representations capture more coherent and hierarchically consistent evidence, enabling the generator to produce more complete and faithful answers. On simpler datasets such as NQ and PopQA, the model maintains competitive results (47.8% / 62.3% and 42.4% / 56.3%), indicating that curvature-based modeling preserves generalization on non-compositional queries. Overall, these findings show that HyperbolicRAG provides the LLM with more precise and semantically grounded evidence, leading to higher factual consistency and completeness in generated responses.

*4) Ablation Results:* To examine the contribution of each component in HyperbolicRAG, we conduct ablation studies on three representative multi-hop QA datasets, as shown in Table IV. The first variant, Euclidean Embedding Alignment, replaces the hyperbolic manifold with a flat Euclidean space while retaining the same contrastive learning objective. Although the pull–push optimization encourages hierarchical alignment, its performance (71.4% on MuSiQue, 88.4% on 2Wiki, and 94.8% on HotpotQA) falls short. This degradation

TABLE III
EM AND F1 (%) PERFORMANCE COMPARISON OF RETRIEVAL METHODS USING THE TOP-5 RETRIEVED PASSAGES.

| Retrieval Methods | Simple QA | Multi-Hop QA | | | | Avg |
|---|---|---|---|---|---|---|
| | NQ | PopQA | MuSiQue | 2Wiki | HotpotQA | |
| ***Simple Baselines*** | | | | | | |
| None | 40.2/54.9 | 28.2/32.5 | 17.6/26.1 | 36.5/42.8 | 37.0/47.3 | 31.9/40.7 |
| BM25 | 45.0/58.9 | 41.6/53.1 | 24.0/31.3 | 38.1/41.9 | 51.3/62.3 | 40.0/49.5 |
| Contriever | 44.7/59.0 | 39.1/49.9 | 20.3/28.8 | 47.9/51.2 | 52.0/63.4 | 40.8/50.5 |
| GTR (T5-base) | 45.5/59.9 | 43.2/56.2 | 25.8/34.6 | 49.2/52.8 | 50.6/62.8 | 42.8/53.3 |
| ***Large Embedding Models*** | | | | | | |
| GTE-Qwen2-7B-Instruct | 46.6/62.0 | **43.5**/56.3 | 30.6/40.9 | 55.1/60.0 | 58.6/71.0 | 46.9/58.0 |
| GritLM-7B | 46.8/61.3 | 42.8/55.8 | 33.6/44.8 | 55.8/60.6 | 60.7/73.3 | 47.9/59.2 |
| NV-Embed-v2-7B | 47.3/61.9 | 42.9/55.7 | 34.7/45.7 | 57.5/61.5 | **62.8**/75.3 | 49.0/60.0 |
| ***Structure-Augmented RAG*** | | | | | | |
| RAPTOR | 36.9/50.7 | 43.1/56.2 | 20.7/28.9 | 47.3/52.1 | 56.8/69.5 | 40.9/51.5 |
| GraphRAG | 30.8/46.9 | 31.4/48.1 | 27.3/38.5 | 51.4/58.6 | 55.2/68.6 | 39.2/52.1 |
| LightRAG | 8.6/16.6 | 2.1/2.4 | 0.5/1.6 | 9.4/11.6 | 2.0/2.4 | 4.5/6.9 |
| HippoRAG | 43.0/55.3 | 42.7/55.9 | 26.2/35.1 | 65.0/71.8 | 52.6/63.5 | 45.9/56.3 |
| HippoRAG2 | 47.1/**62.0** | 42.9/56.2 | 37.2/48.6 | 65.0/71.0 | 62.7/**75.5** | 51.0/62.7 |
| **HyperbolicRAG** | **47.8/62.3** | 42.4/**56.3** | **39.5/50.6** | **65.5/72.3** | 61.7/75.2 | **51.4/63.3** |

TABLE IV
ABLATION STUDY ON MULTI-HOP QA DATASETS (RECALL@5).

| Retrieval Methods | Multi-Hop QA Dataset | | |
|---|---|---|---|
| | Musique | 2Wiki | HotpotQA |
| Euclidean Alignment | 71.4 | 88.4 | 94.8 |
| HyperbolicRAG w/o Hyperbolic Signal | 74.7 | 90.4 | 96.2 |
| HyperbolicRAG w/o Euclidean Signal | 73.9 | 90.4 | 95.9 |
| **HyperbolicRAG** | **76.2** | **91.1** | **96.3** |

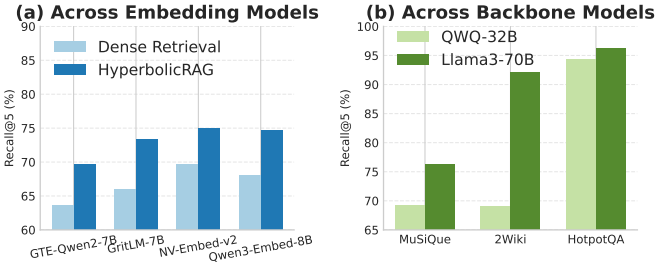


Fig. 5. Comparison of HyperbolicRAG under (a) different embedding encoders and (b) generative backbones.

reflects an inherent limitation of Euclidean space, whose isotropic geometry captures pairwise similarity but fails to express asymmetric containment among passages, entities, and facts. Consequently, hierarchical signals are compressed into a single representational layer, reducing the model's ability to distinguish between abstract and specific evidence.

The second and third variants assess the impact of the dual-space fusion mechanism by disabling one of the ranking channels. Without hyperbolic ranking signal, the model relies purely on Euclidean similarity (74.7% on MuSiQue and 90.4% on 2Wiki); conversely, removing the Euclidean ranking yields a comparable decline (73.9% and 90.4%). These results suggest that the two spaces capture complementary aspects of relevance: the Euclidean space refines local semantic consistency, whereas the hyperbolic space preserves global structural hierarchy. The complete HyperbolicRAG achieves the best overall performance (76.2%, 91.1%, and 96.3%), confirming that the proposed rank-level fusion combines hyperbolic and Euclidean signals to deliver semantically precise and structurally coherent retrieval.

*5) Model-Agnostic Effectiveness:* We assess the compatibility of HyperbolicRAG with a broad set of dense encoders and LLM backbones. As shown in Fig. 5(a), integrating our hierarchical enhancement mechanism with various dense retrievers, including GTE Qwen2 7B Instruct, GritLM 7B, NV Embed v2 (7B), and Qwen3 Embedding 8B [63], consistently yields higher Passage Recall@5 compared with their Euclidean counterparts. These results indicate that the hyperbolic representation reliably preserves hierarchical semantics across diverse embedding distributions.

In addition to retrieval encoders, Fig. 5(b) demonstrates that HyperbolicRAG provides stable performance gains when paired with different LLM backbones. Although Llama3 70B serves as our primary generator, comparable improvements are observed with QWQ 32B [64], which confirms that the advantages of hierarchical enhancement generalize across architectures of different capacities. We also find that the overall retrieval quality is influenced by the accuracy of the relational graph construction pipeline. Inaccurate or incomplete extraction of entities and relations may introduce noise and fragmentation, which in turn limits the achievable performance. Despite these constraints, HyperbolicRAG consistently enhances retrieval robustness and hierarchical sensitivity across both retriever and backbone variations.

*6) Effect of the Curvature Hyperparameter:* To assess the sensitivity of the model to geometric settings, we vary the curvature hyperparameter $c$. As illustrated in Fig. 6, $c$ has minimal influence on retrieval performance, while a moder-

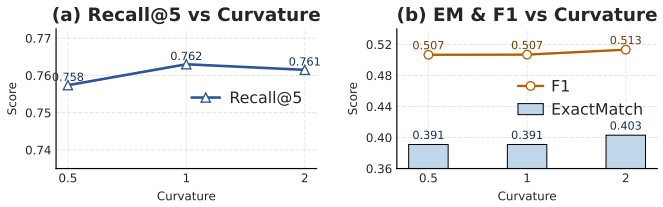


Fig. 6. Effect of the curvature hyperparameter on retrieval and generation performance.

ate curvature leads to slight but consistent improvements in generation metrics. This suggests that the model is robust to curvature variations and benefits marginally from non-Euclidean geometry.

## VI. Conclusion

In this work, we propose HyperbolicRAG, a hierarchy-aware retrieval framework that captures the intrinsic structure of entity–fact–passage relations through hyperbolic geometry. By modeling hierarchical containment within a curved space, HyperbolicRAG effectively alleviates the hubness bias inherent in Euclidean embeddings and improves the precision of evidence retrieval. A dual-space retrieval mechanism further integrates Euclidean and hyperbolic reasoning, combining fine-grained semantic similarity with global structural awareness. Extensive experiments across multiple QA benchmarks demonstrate consistent gains in both retrieval and answer generation, particularly on multi-hop reasoning tasks.

## References


[1] Z. Yi, J. Ouyang, Y. Liu, T. Liao, Z. Xu, and Y. Shen, "A survey on recent advances in llm-based multi-turn dialogue systems," *arXiv:2402.18013*, 2024.

[2] Y. Zhang, H. Jin, D. Meng, J. Wang, and J. Tan, "A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods," *arXiv:2403.02901*, 2024.

[3] J. Liu, Z. Qiu, Z. Li, Q. Dai, W. Yu, J. Zhu, M. Hu, M. Yang, T.-S. Chua, and I. King, "A survey of personalized large language models: Progress and future directions," *arXiv:2502.11528*, 2025.

[4] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A survey on rag meeting llms: Towards retrieval-augmented large language models," in *Proceedings of KDD*, 2024, pp. 6491–6501.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Proceedings of NeurIPS*, 2020, pp. 9459–9474.

[6] X. He, Y. Tian, Y. Sun, N. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," in *Proceedings of NeurIPS*, 2024, pp. 132 876–132 907.

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

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

[9] B. Jimenez Gutierrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, "Hipporag: Neurobiologically inspired long-term memory for large language models," in *Proceedings of NeurIPS*, 2024, pp. 59 532–59 569.

[10] B. J. Gutiérrez, Y. Shu, W. Qi, S. Zhou, and Y. Su, "From rag to memory: Non-parametric continual learning for large language models," *arXiv:2502.14802*, 2025.

[11] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná, "Hyperbolic geometry of complex networks," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 82, no. 3, p. 036106, 2010.

[12] Z. Pan and P. Wang, "Hyperbolic hierarchy-aware knowledge graph embedding for link prediction," in *Findings of EMNLP 2021*, 2021, pp. 2941–2948.

[13] V. Karpukhin, B. Oguz, S. Min, P. S. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering." in *Proceedings of EMNLP*, 2020, pp. 6769–6781.

[14] N. Tomasev, M. Radovanovic, D. Mladenic, and M. Ivanovic, "The role of hubness in clustering high-dimensional data," *IEEE TKDE*, vol. 26, pp. 739–751, 2013.

[15] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, "Hubs in space: Popular nearest neighbors in high-dimensional data," *JMLR*, vol. 11, pp. 2487–2531, 2010.

[16] H. Zhang, P. D. Rich, A. K. Lee, and T. O. Sharpee, "Hippocampal spatial representations exhibit a hyperbolic geometry that expands with experience," *Nature Neuroscience*, vol. 26, pp. 131–139, 2023.

[17] R. Sarkar, "Low distortion delaunay embedding of trees in hyperbolic plane," in *International Symposium on Graph Drawing*, 2011, pp. 355–366.

[18] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic entailment cones for learning hierarchical embeddings," in *Proceedings of ICML*, 2018, pp. 1646–1655.

[19] Q. Zhang, S. Chen, Y. Bei, Z. Yuan, H. Zhou, Z. Hong, J. Dong, H. Chen, Y. Chang, and X. Huang, "A survey of graph retrieval-augmented generation for customized large language models," *arXiv:2501.13958*, 2025.

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

[21] T. T. Procko and O. Ochoa, "Graph retrieval-augmented generation for large language models: A survey," in *Proceedings of IEEE AIxSET*, 2024, pp. 166–169.

[22] H. T. Jonathan Larson Darren Edge, "Lazygraphrag: Setting a new standard for quality and cost." *Microsoft Blog*, 2024.

[23] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, "Grag: Graph retrieval-augmented generation," *arXiv:2405.16506*, 2024.

[24] Z. Li, X. Chen, H. Yu, H. Lin, Y. Lu, Q. Tang, F. Huang, X. Han, L. Sun, and Y. Li, "Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization," *arXiv:2410.08815*, 2024.

[25] L. Liang, Z. Bo, Z. Gui, Z. Zhu, L. Zhong, P. Zhao, M. Sun, Z. Zhang, J. Zhou, W. Chen *et al.*, "Kag: Boosting llms in professional domains via knowledge augmented generation," in *Companion Proceedings of WWW 2025*, 2025, pp. 334–343.

[26] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proceedings of NeurIPS*, 2017.

[27] ——, "Learning continuous hierarchies in the lorentz model of hyperbolic geometry," in *Proceedings of ICML*, 2018, pp. 3779–3788.

[28] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *Proceedings of NeurIPS*, 2018.

[29] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Proceedings of NeurIPS*, 2019.

[30] J. Dai, Y. Wu, Z. Gao, and Y. Jia, "A hyperbolic-to-hyperbolic graph convolutional network," in *Proceedings of CVPR*, 2021, pp. 154–163.

[31] R. Shimizu, Y. Mukuta, and T. Harada, "Hyperbolic neural networks++," *arXiv:2006.08210*, 2020.

[32] W. Chen, X. Han, Y. Lin, H. Zhao, Z. Liu, P. Li, M. Sun, and J. Zhou, "Fully hyperbolic neural networks," in *Proceedings of ACL*, 2022, pp. 5672–5686.

[33] J. Yan, L. Luo, C. Deng, and H. Huang, "Unsupervised hyperbolic metric learning," in *Proceedings of CVPR*, 2021, pp. 12 465–12 474.

[34] A. Ermolov, L. Mirvakhabova, V. Khrulkov, N. Sebe, and I. Oseledets, "Hyperbolic vision transformers: Combining improvements in metric learning," in *Proceedings of CVPR*, 2022, pp. 7409–7419.

[35] B. Dhingra, C. J. Shallue, M. Norouzi, A. M. Dai, and G. E. Dahl, "Embedding text in hyperbolic spaces," *arXiv:1806.04313*, 2018.

[36] C. Lang, A. Braun, L. Schillingmann, and A. Valada, "On hyperbolic embeddings in object detection," in *Proceedings of DAGM GCPR*, 2022, pp. 462–476.

[37] X. Fu, Y. Wei, Q. Sun, H. Yuan, J. Wu, H. Peng, and J. Li, "Hyperbolic geometric graph representation learning for hierarchy-imbalance node classification," in *Proceedings of WWW*, 2023, pp. 460–468.

[38] L. Sun, Z. Zhang, J. Zhang, F. Wang, H. Peng, S. Su, and P. S. Yu, "Hyperbolic variational graph neural network for modeling dynamic graphs," in *Proceedings of AAAI*, 2021, pp. 4375–4383.

[39] Y. Li, H. Chen, X. Sun, Z. Sun, L. Li, L. Cui, P. S. Yu, and G. Xu, "Hyperbolic hypergraphs for sequential recommendation," in *Proceedings of CIKM*, 2021, pp. 988–997.

[40] A. Suzuki, A. Nitanda, J. Wang, L. Xu, K. Yamanishi, and M. Cavazza, "Generalization error bound for hyperbolic ordinal embedding," in *Proceedings of ICML*, 2021, pp. 10 011–10 021.

[41] G. Mishne, Z. Wan, Y. Wang, and S. Yang, "The numerical stability of hyperbolic representation learning," in *Proceedings of ICML*, 2023, pp. 24 925–24 949.

[42] M. Yang, A. Feng, B. Xiong, J. Liu, I. King, and R. Ying, "Hyperbolic fine-tuning for large language models," *arXiv:2410.04010*, 2024.

[43] K. Desai, M. Nickel, T. Rajpurohit, J. Johnson, and S. R. Vedantam, "Hyperbolic image-text representations," in *Proceedings of ICML*, 2023, pp. 7694–7731.

[44] N. He, R. Anand, H. Madhu, A. Maatouk, S. Krishnaswamy, L. Tassiulas, M. Yang, and R. Ying, "Helm: Hyperbolic large language models via mixture-of-curvature experts," *arXiv:2505.24722*, 2025.

[45] N. He, H. Madhu, N. Bui, M. Yang, and R. Ying, "Hyperbolic deep learning for foundation models: A survey," in *Proceedings of KDD*, 2025, pp. 6021–6031.

[46] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *IEEE TPAMI*, vol. 44, no. 12, pp. 10 023–10 044, 2021.

[47] M. Yang, M. Zhou, Z. Li, J. Liu, L. Pan, H. Xiong, and I. King, "Hyperbolic graph neural networks: A review of methods and applications," *arXiv:2202.13852*, 2022.

[48] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized pagerank," *arXiv:1006.2880*, 2010.

[49] Y. Wang, R. Ren, J. Li, W. X. Zhao, J. Liu, and J.-R. Wen, "Rear: A relevance-aware retrieval-augmented framework for open-domain question answering," *arXiv:2402.17497*, 2024.

[50] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," *arXiv:2212.10511*, 2022.

[51] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Musique: Multihop questions via single-hop question composition," *IEEE TACL*, vol. 10, pp. 539–554, 2022.

[52] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps," *arXiv:2011.01060*, 2020.

[53] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *arXiv:1809.09600*, 2018.

[54] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *Proceedings of SIGIR*, 1994, pp. 232–241.

[55] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, "Unsupervised dense information retrieval with contrastive learning," *arXiv:2112.09118*, 2021.

[56] J. Ni, C. Qu, J. Lu, Z. Dai, G. H. Abrego, J. Ma, V. Y. Zhao, Y. Luan, K. B. Hall, M.-W. Chang *et al.*, "Large dual encoders are generalizable retrievers," *arXiv:2112.07899*, 2021.

[57] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models," *arXiv:2104.08663*, 2021.

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

[59] N. Muennighoff, H. Su, L. Wang, N. Yang, F. Wei, T. Yu, A. Singh, and D. Kiela, "Generative representational instruction tuning," *arXiv:2402.09906*, 2024.

[60] C. Lee, R. Roy, M. Xu, J. Raiman, M. Shoeybi, B. Catanzaro, and W. Ping, "Nv-embed: Improved techniques for training llms as generalist embedding models," *arXiv:2405.17428*, 2024.

[61] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "RAPTOR: Recursive abstractive processing for tree-organized retrieval," in *Proceedings of ICLR*, 2024.

[62] AI@Meta, "Llama 3 model card," 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[63] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin *et al.*, "Qwen3 embedding: Advancing text embedding and reranking through foundation models," *arXiv:2506.05176*, 2025.

[64] Q. Team, "Qwq-32b: Embracing the power of reinforcement learning," 2025. [Online]. Available: https://qwenlm.github.io/blog/qwq-32b/