PanoramaRAG: Enabling Consistent Global Topic Awareness in Graph-Based RAG

Ding Deng[1], Xiang Li[1], Yaqing Zhang[2], Meng Li[1,*], Xiting Wang[1,3,4,*]

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Huawei Poisson Lab
[3]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education
[4]Beijing Key Laboratory of Research on Large Models and Intelligent Governance
{ddya, mengli.24, xitingwang}@ruc.edu.cn

Abstract

Graph-based Retrieval-Augmented Generation (RAG), which models relationships between fine-grained semantic units as a graph, effectively facilitates multi-hop reasoning to enhance large language model generation. However, its design focuses on local relationships, resulting in suboptimal performance for tasks that require global context, and the separation of query refinement from indexing limits the system's ability to capture high-level implicit relationships within the graph. This paper proposes a **Panorama**-guided **RAG** paradigm (PanoramaRAG) that integrates a light yet comprehensive "panorama" of the corpus to guide all stages of the retrieval process. This hub bridges the knowledge graph, language models, and queries in a computationally efficient manner, applicable to both open-source and closed-source models. Experimental results demonstrate that our method exhibits strong performance across five datasets and a variety of tasks. Our code is available at https://github.com/Deng-Dean/PanoRAG.

1 Introduction

Retrieval-Augmented Generation has emerged as a powerful paradigm for enhancing Large Language Models (LLMs) with external knowledge, particularly for integrating up-to-date information and private knowledge sources (Izacard et al., 2022; Ram et al., 2023; Fan et al., 2024; Chen et al., 2024; Yu et al., 2025; Zhang et al., 2025). Among the various RAG approaches, graph-based methods have attracted increasing attention due to their ability to represent knowledge in a structured, relational manner. In these methods, the knowledge base is modeled as a graph, where nodes represent a minimal semantic unit and edges encode the relationships between them (Edge et al., 2024; Jimenez Gutierrez et al., 2024; Soman et al., 2024). This graph structure enables the discovery of complex relationships among entities, making it particularly well-suited for answering multi-hop queries, which require retrieving and reasoning over multiple, interconnected pieces of supporting evidence (Tang and Yang, 2024; Zhang and Zhang, 2025). By leveraging the graphs ability to model intricate dependencies, graph-based RAG methods can provide more coherent and contextually accurate responses in tasks requiring deep reasoning and evidence synthesis.

Although graph structures are effective at modeling relationships between different semantic units, they often fall short in capturing global semantics, resulting in suboptimal performance on query-focused summarization tasks and high-level queries (see Appendix B). Prior work on integrating global data information typically improves only a single stage of the retrieval process, either refining the query using global information (Qian et al., 2024) or optimizing the keyword indexing without influencing the query or keyword generation itself (Edge et al., 2024; Guo et al., 2024). This decoupling between query refinement and retrieval index construction hinders the ability to comprehensively capture high-level relationships implicit in the graph, leading to suboptimal retrieval results (Sec. 4.4, Table 1). Moreover, existing approaches that incorporate global information often incur significant additional costs. For instance, MemoRAG modifies the internal parameters of large language models to enable global-aware keyword generation, which requires additional training and is incompatible with closed-source models.

To address the aforementioned challenges, we propose a **Panorama**-guided **RAG** framework (PanoramaRAG) that integrates a light yet comprehensive "panorama" of the corpus to guide all stages of the retrieval process. This panorama serves as a model-agnostic bridge between the Knowledge Graph (KG), LLMs, and queries, pro-

Findings of the Association for Computational Linguistics: ACL 2026, pages 40205–40217

viding global semantic guidance without substantial computational overhead. By abstracting core content through hierarchical keywords, the panorama is seamlessly compatible with both keyword-based indexing and query expansion. We integrate this global perspective throughout the entire retrieval lifecycleencompassing offline indexing, online query construction, and document retrieval. Specifically, we introduce the following technical innovations:

- **PanoramaRAG**: A novel RAG framework with a panoramic view of the corpus, which integrates global semantics into both the indexing and retrieval stages at minimal computational cost. By embedding corpus-level structure into retrieval-aware representations, PanoramaRAG improves semantic consistency without sacrificing efficiency.
- **Panorama-aware keyword construction**: A method for refining queries and keywords conditioned on the panorama, ensuring that keywords consider the full context of the data before retrieval, while not limiting the types of large models (open-source or closed-source) that can be used.
- **Panorama-aware document retrieval**: A technique that utilizes a panorama-enhanced graph and expansion mechanism to perform efficient retrieval. It improves both retrieval and generation effectiveness without adding significant overhead, leading to a notable improvement in retrieval performance.

2 Related Work

2.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation(RAG) augments large language models(LLMs) by grounding the generation process in external knowledge retrieved from a corpus (Ram et al., 2023; Cheng et al., 2025). Conventional RAG systems (Gao et al., 2023; Ma et al., 2023) typically employ vector-space similarity search over passages that are segmented arbitrarily, without preserving semantic continuity. This coarse-grained segmentation fails to capture essential relationships and contextual dependencies between chunks. Consequently, fragmented knowledge integration limits the LLMs ability to perform cross-source synthesis and produce globally coherent responses (Edge et al., 2024; Guo et al., 2024; Chen et al., 2025). To address this limitation, recent research has focused on enhancing RAG pipelines, such as FLARE (Jiang et al., 2023), Self-RAG (Asai et al., 2024), and MemoRAG (Qian et al., 2024). While MemoRAG improves long-context handling through corpus compression and global keyvalue caching, its reliance on trainable modules hinders integration with closed-source LLMs. Other efforts leverage structured knowledge bases to enhance information aggregation and semantic-level retrieval (Sun et al., 2023; Li et al., 2024; He et al., 2024; Zhu et al., 2025; Zhang, 2025). Within this paradigm of structured retrieval, our work introduces graph-structured indices that enable contextual reasoning as an inherent capability.

2.2 Graph-based RAG

Graph-based RAG has emerged as a promising direction for enhancing LLMs with structured knowledge (Han et al., 2024; Zhang et al., 2025; Liu, 2025). In contrast to traditional RAG, graph-based RAG approaches extract entities and their relations to construct knowledge graphs that guide retrieval and generation. GraphRAG (Edge et al., 2024) is the first method proposed to automatically construct an entity graph from the corpus and pre-generate community-level summaries based on graph structure, emphasizing the importance of integrating global information. While GraphRAGs reliance on topological community detection may overlook latent semantic groupings, its efficacy is further constrained by lengthy (avg. 400-600 words) community summaries that aggregate heterogeneous contentincluding all entities, relations, and claimsinto dense textual blocks. The fundamental integration approach is to optimize keyword indexing independently from query context or keyword generation, exemplified by LightRAG (Guo et al., 2024) and GraphRAG. LightRAG extends GraphRAG's paradigm by integrating graph structure with vector representations and adopting a dual-level retrieval framework: a low-level retriever targets specific entities, while a high-level retriever captures their relations. Alternative methods employ query refinement leveraging global information (Qian et al., 2024). However, decoupling query refinement from index construction hinders capturing implicit high-level graph relationships, thereby degrading retrieval performance. To address this, our work establishes consistent global information awareness across all pipeline stages to holistically inte-

grate corpus-level context.

3 Methodology

PanoramaRAG incorporates global information throughout its retrieval pipeline, as illustrated in Fig. 1. At the offline indexing stage, we first leverage an LLM to extract corpus topics and construct a hierarchical topic tree through bottom-up clustering (Sec. 3.1). During the online keyword construction stage, we then generate dual-level keywords from user queries and map them onto the topic tree, thereby achieving semantic alignment between keywords and corpus-specific themes (Sec. 3.2). Finally, at the KG retrieval stage, we retrieve top-k relevant nodes based on the keywords, perform multi-hop expansion on the graph with filtering, and finally generate the response via LLMs (Sec. 3.3).

3.1 Offline Hierarchical Topic-Aware Indexing

Inspired by human semantic memory—where knowledge is organized hierarchically from local details into abstract global representations for efficient recall (Binder and Desai, 2011)—we propose a tree-structured architecture built upon a collection of document chunks to concisely model global semantics.

Given a collection of document chunks $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, we first extract fine-grained topics as leaf nodes. For a chunk c_i, we extract latent topics by prompting an LLM $\mathcal{M}$ to generate a set of topic-description pairs:

$$\{(t_j, s_j)\}_{j=1}^{N} = \mathcal{T}(\mathcal{M}, c_i)$$

Here, t_j is a topic node and s_j is its textual description. These leaf nodes form the initial topic set $\mathcal{T}_L$ at the finest layer L. We then perform bottom-up hierarchical clustering on these topics to form higher-level abstractions $\mathcal{T}_{l-1}$, continuing until a stopping criterion δ is met. The set of topic nodes at layer l is denoted as $\mathcal{T}_l = \{t_1, t_2, \ldots, t_m\}$.

Building upon this foundation, we leverage topic information to guide knowledge graph construction. For each text chunk c_i, we utilize its associated topic set T^i as auxiliary context. This augmented input is processed by an LLM to jointly extract three knowledge components:

$$\mathcal{S}_{\text{LLM}_{\text{kg}}}(c_i, \mathcal{T}_i) \rightarrow \begin{pmatrix} \mathcal{E}_i & = & \{e_j\} \\ \mathcal{R}_i & = & \{r_k \mid e_a \xrightarrow{r_k} e_b\} \\ \mathcal{A}_i & = & \{(e_j, t_p, \rho_{jp})\} \end{pmatrix} \tag{1}$$

$\mathcal{E}_i$ and $\mathcal{R}_i$ denote the entities and relations extracted from c_i, while $\mathcal{A}_i$ represents affiliations between entities and topics with ρ_{jp} indicating the strength of association. Topic nodes bridge semantically similar but structurally disconnected entities, enabling the graph to capture latent relationships and enhance its representational capacity.

In summary, this hierarchical topic representation provides a precise and efficient mechanism for capturing global information. The tree architecture organizes knowledge at multiple granularities, from coarse themes to fine details, and serves two key functions: it supports LLM keyword generation by providing abstract themes, and enhances KG semantic linking through fine-grained associations.

3.2 Online Panorama-Aware Keyword Construction

To adapt the RAG system for diverse query types (e.g., *Specific*, *Abstract*, *Summarization*, *Multi-document QA*, etc.), we refine queries by segmenting them into multi-granular keywords to better align with KG structures. This stage leverages the panorama built in Sec. 3.1 to construct corpus-aware keywords for bridging query abstraction levels with KG information granularity.

First, we employ an LLM to generate dual-level keywords from the query per (Sudhi et al., 2024), capturing multi-scale semantics. 1) **Low-level keywords** are detail-oriented and typically refer to concrete entities or salient topic-specific terms (e.g., actress name like *Scarlett*). 2) **High-level keywords** focus on broader topics and abstract themes (e.g., *literary criticism*). The generated dual-level keywords enable simultaneous retrieval of granular details and more global contexts, enhancing both recall via high-level abstraction, and precision via low-level specificity.

Next, We align the dual-level keywords to topic tree to obtain corpus-aware representations. For each keyword k extracted from the query, we identify its optimal semantically similar topic node in the topic tree $t^* \in \mathcal{T}$ measured by the topic nodes' cosine similarity in the embedding space:

$$t^* = \arg\max_{t \in \mathcal{T}} \cos\left(\mathcal{E}(k), \mathcal{E}(t)\right)$$

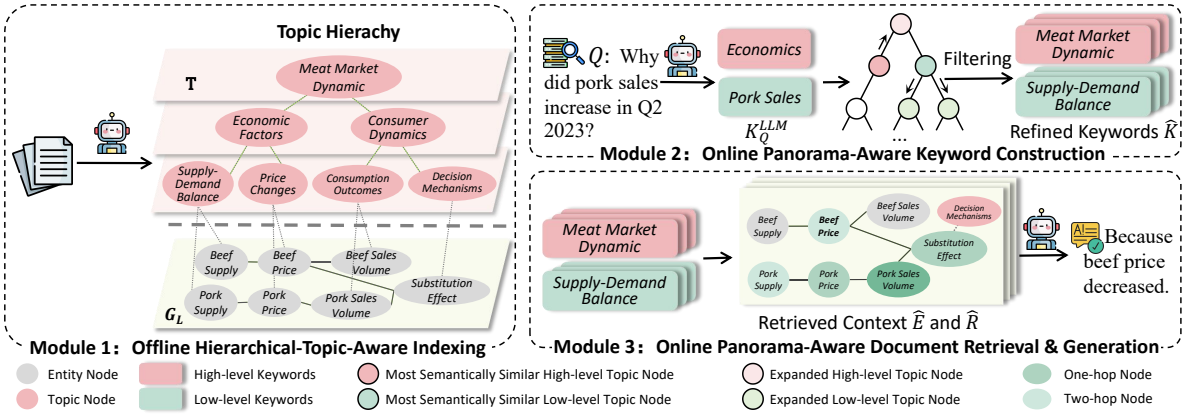

Figure 1: The overall framework of PanoramaRAG. During offline indexing (*Module 1*), we build a topic hierarchy using bottom-up clustering. During online keyword construction stage (*Module 2*), given a user query (e.g., "Why did pork sales increase?"), we construct corpus-aware refined keywords by using generated keywords mapped to the topic tree. Low-level keywords search downward from the initial anchor node for specific details, while high-level keywords search upward to discover broader concepts. At the KG retrieval stage (*Module 3*), we perform panorama-aware retrieval based on the top-k refined keywords: first expanding to directly connected One-hop Nodes (e.g., "Pork Sales Volume ⟶ Substitution Effect", "Pork Sales Volume ⟶ Pork Price"), then traversing to Two-hop Nodes to capture panoramic context (e.g., "Pork Sales Volume ⟶ Substitution Effect ⟶ Beef Price", "Pork Sales Volume ⟶ Pork Price ⟶ Pork Supply"), and finally we generate the response via LLMs.

where $\mathcal{E}$ denotes an embedding function, and $\cos(\cdot)$ computes cosine similarity between two vectors. Then, depending on keyword level, different expansion strategies are employed. For high-level keywords, we traverse **upward** in the topic tree to collect up to two ancestor nodes, capturing broader semantic abstractions. For low-level keywords, we explore **downward** to collect all child nodes of the matched topic node, representing more fine-grained subtopics. In this way, upward traversal for high-level keywords concatenates fragmented semantics into unified conceptual frameworks, resolving information scattering across documents. While downward expansion for low-level keywords ensures exhaustive evidence coverage, preventing critical detail omission. To avoid noisy or off-topic expansions, we score each candidate topic node t_i by its similarity to the original query q, using cosine similarity:

$$s_i = \cos\left(\mathcal{E}(q), \mathcal{E}(t_i)\right)$$

To ensure computational efficiency while maintaining semantic coverage, we preserve the original keyword set and incrementally incorporate additional candidates derived from the panorama-based topic tree. Specifically, candidate keywords are ranked by their semantic similarity to the query and appended to the original set until a maximum cap n is reached for each keyword level:

$$K'_{\text{high}} = K_{\text{high}} \cup \text{Top-}\left(\min(n, |K^{\text{ext}}_{\text{high}}|)\right)\left(K^{\text{ext}}_{\text{high}}\right) \tag{2}$$

$$K'_{\text{low}} = K_{\text{low}} \cup \text{Top-}\left(\min(n, |K^{\text{ext}}_{\text{low}}|)\right)\left(K^{\text{ext}}_{\text{low}}\right) \tag{3}$$

Here, K_{high} and K_{low} denote the original high- and low-level keyword sets generated from the query, while $K^{\text{ext}}_{\text{high}}$ and $K^{\text{ext}}_{\text{low}}$ represent the expansion candidates from the topic tree. The hyper-parameter n denotes the maximum number of additional keywords allowed from the extended set. This panorama-guided expansion and similarity-based filtering ensures that the keyword set is both semantically aligned and query-relevant, effectively bridging the gap between LLM-generated keywords and graph-based content structure.

3.3 Online Panorama-Aware Document Retrieval & Generation

While existing keyword-based methods typically retrieve nearest-neighbor nodes based on structural proximity at the online retrieval stage (Edge et al., 2024; Guo et al., 2024), this paradigm may fall short in complex scenarios that demand deeper contextual understanding or multi-step reasoning. To address this, we incorporate a multi-hop expansion strategy that gathers semantically related neighbors to gather potential structural information from the graph.

To avoid overwhelming the LLM with irrelevant information (Zhu et al., 2024), we integrate a similarity-guided filtering mechanism aligned with the query intent. The process comprises three stages: Dual Similarity Matching, Neighbor Expansion, and Node Filtering.

Step 1: Dual Similarity Matching.

- *Entity Matching*: For low-level keywords, we compute similarity to graph nodes and retain top-k_{graph} entities:

$$E_0 = \text{TopK}_{k_{\text{entity}}}\{v_i \in V \mid \cos(\mathbf{E}_{\text{low}}, \mathbf{E}_{v_i})\}$$

- *Relation Matching*: For high-level keywords, we compute similarity to relations and retain top-k_{rel} relations:

$$R_0 = \text{TopK}_{k_{\text{rel}}}\{r_j \in R \mid \text{RelSim}(\mathbf{E}_{\text{high}}, \mathbf{E}_{r_j})\}$$

where E_0 is the initial entity anchor set, and R_0 is the initial relation anchor set. $\mathbf{E}_{\text{low}}$ and $\mathbf{E}_{\text{high}}$ are embeddings of low-level and high-level keywords. The function RelSim is a relational similarity function (e.g., TransE scoring).

Step 2: Neighbor Expansion. For each node $v \in E_0$, its neighbor set $N(v)$ is retrieved, and a breadth-first traversal over the knowledge graph is performed up to two hops. The expansion at each hop is defined as:

$$E_i = \bigcup_{v \in E_{i-1}} \{t \mid (v, t) \in \text{edges}\}$$

All entities collected across hops are globally deduplicated. By expanding neighbors through topic nodes, we bridge thematically related entities and retrieve potentially critical entities lacking direct KG connections. The final candidate entity set is:

$$E = E_0 \cup E_1 \cup E_2$$

Step 3: Node Filtering. Each node $u \in E$ is scored via a composite ranking function:

$$\phi(u) = \alpha \cdot \text{sim}(\mathbf{q}, \mathbf{v}_u) + (1 - \alpha) \cdot w_u$$

where w_u denotes the degree of node u, and $\text{sim}(\mathbf{q}, \mathbf{v}_i)$ computes the embedding-based similarity (e.g., cosine similarity) between the query embedding $\mathbf{q}$ and node embedding $\mathbf{v}_i$. Nodes are greedily selected into the final set $\hat{E}$ until the combined `description` fields reach the LLM context limit B tokens. Relations are ranked in descending order according to their weights, and the associated descriptions are aggregated to construct the edge-level information set $\hat{\mathcal{R}}$.

The final selected entity set $\hat{E}$ consists of nodes directly matched by the keywords, as well as the most informative two-hop neighbors. Additionally, a high-quality set of relations and relevant text chunks $\hat{R}$ are jointly serialized into a structured format and provided as grounding context for the LLM. This neighborhood exploration enables panorama-aware document retrieval, after which candidate passages undergo joint optimization based on textual similarity and structural importance within the topic hierarchy prior to generation, further generating answers based on precise local matches and broader relational evidence.

4 Experiments

In this section, we conduct experiments to verify PanoramaRAG for tasks including Query-Focused Summarization, Long Context Handling, and Multi-hop QA. In particular, we answer the following research questions (RQ):

RQ1: How effectively does PanoramaRAG handle diverse and complex tasks, including multi-hop reasoning, summarization, and cross-domain queries, compared with existing baselines?

RQ2: How are the design choices and optimization strategies of PanoramaRAG effective, particularly in addressing high-level, abstract questions?

RQ3: Is PanoramaRAG robust to hyperparameter variations additionally introduced in our method?

RQ4: How does PanoramaRAG compare to existing baselines in cost and time efficiency?

4.1 Dataset

To explore **RQ1**, we curate two complementary task categories: (1) Global tasks demanding holistic information synthesis, evaluated on Ultradomain (Qian et al., 2024) (emphasizing Agriculture and Mix tasks within its 20-domain suite) and LongBench (Qian et al., 2024) (covering single/multi-document QA, non-QA tasks, and long-book QA), for which we employ a randomly sampled subset of 95 queries to manage computational demands while preserving benchmark representativeness. (2) Fact-intensive tasks requiring localized evidence verification, assessed via classical multi-hop QA benchmarks, includ-

ing 2WikiMultiHopQA (Ho et al., 2020) and HotpotQA (Yang et al., 2018).

4.2 Baselines

We consider both naive RAG methods and graph-based RAG methods for comparison. **NaïveRAG** (Gao et al., 2023) is a standard baseline that segments raw text into chunks and stores them in a vector database. At query time, it retrieves the most similar chunks using embedding-based similarity. For graph-based RAG methods, we include LightRAG (Guo et al., 2024), MemoRAG (Qian et al., 2024), and PathRAG (Chen et al., 2025). Our method is largely based on **LightRAG**, where a graph-based structure with entity-relation extraction and dual-level keyword retrieval is applied. **MemoRAG** is another graph-based RAG method that incorporates global memory in an additional memory model. When given a query, it first generates draft answers as clues and then guides the retrieval of relevant evidence from the full context. **PathRAG** integrates flow-based pruning and path-guided prompting to retrieve critical relational paths from the index graph, enabling precise capture of complex relationships in structured datasets.

4.3 Evaluation

We compare PanoramaRAG with various baseline methods across multiple tasks, as summarized in Table 1. Given the distinct characteristics of each dataset, we adopt a combination of evaluation metrics, including both LLM-based subjective scoring and objective accuracy measurements, to facilitate a comprehensive performance comparison.

For subjective evaluation, we deploy deepseek-r1-0528 to score responses on a 0-10 scale across four dimensions adapted from GraphRAG (Edge et al., 2024): comprehensiveness, diversity, empowerment, and overall quality. Responses that substantially deviate from reference answers or are completely incorrect receive 0 points. To ensure impartiality, we randomly shuffle the order of all responses before having the LLM evaluate them side-by-side. Model rankings are determined by their average performance across all questions. To mitigate potential biases introduced by LLM-based evaluation, we additionally conducted human evaluations on four datasets and verified their correlation with the model-generated scores. For multi-hop datasets, we further incorporated the F1 metric as an indicator to assess the models retrieval capability. For tasks requiring precise matching, objective accuracy provides a direct measure of reasoning capabilities without needing additional LLM assessment.

[1]HAS (Human-Annotated Scores) show significant correlation with LLM-generated Scores: Pearson's $r = 0.76$ ($p = 0.010$), Spearman's $\rho = 0.76$ ($p = 0.011$).

4.4 Performance Comparison

To address **RQ1**, we evaluate across multiple task categories. As summarized in Table 1, our proposed PanoramaRAG demonstrates strong performance across all tasks.

For global tasks, PanoramaRAG achieves an 15.8% score improvement over the best graph-based baseline and a 4.3% improvement over the best baseline (e.g., 8.44 Score on Agriculture and 7.45 Score on Mix). Ultradomain emphasizes holistic corpus-level understanding and integration, where most tasks involve content organization and query-focused summarization. In this context, PanoramaRAG demonstrates a clear advantage by jointly modeling fine-grained entity information and global topic tree structures. In contrast, all models, including PanoramaRAG, perform relatively poorly on LongBench due to its challenging multiple-choice format, high difficulty, and diverse question types. Nevertheless, PanoramaRAG achieves a 1.7% improvement over the best-performing baseline.

We observe that although PanoramaRAG exhibits a clear advantage on 2WikiMultiHopQA and ranks second on HotpotQA, this performance discrepancy can be attributed to differences in the underlying graph data scale-the data volume in the graph structure of 2WikiMultiHopQA is approximately 52.96% that of HotpotQA. This reduced graph complexity likely enhances retrieval effectiveness for more complex reasoning tasks by minimizing noise and improving information alignment. It is worth noting that our method demonstrates strong competitiveness on the metric F1, achieving an 8.4% improvement and a 12.1% improvement over graph-based methods on HotpotQA. PanoramaRAG consistently outperforms other graph-based methods, underscoring its robustness across varying datasets.

In conclusion, these results affirm that PanoramaRAG consistently outperforms existing baselines across diverse and complex tasks. The promising experimental results demonstrate that our panorama-graph-based RAG paradigm signifi-

Method	Agriculture			Mix			LongBench	2Wikimultihopqa				HotpotQA			
	Score	Rank	HAS[1]	Score	Rank	HAS[1]	Accuracy(%)	Score	Rank	HAS[1]	F1(%)	Score	Rank	HAS[1]	F1(%)
NaiveRAG	8.24	2.67	6.8	6.01	3.13	5.3	32.57	5.47	3.24	6.8	42.8	5.33	3.60	6.3	49.2
LightRAG	8.06	2.71	7.1	5.60	3.39	5.2	33.68	5.96	3.13	6.4	33.4	4.80	3.65	5.6	44.9
MemoRAG	7.74	3.50	7.0	7.02	2.66	6.9	34.08	6.42	2.74	7.6	**51.7**	**6.23**	**2.04**	**7.8**	59.3
PathRAG	7.21	3.72	5.4	5.87	3.45	5.7	31.58	5.69	3.19	6.3	33.8	4.82	3.90	5.6	45.6
PanoramaRAG	**8.44**	**2.38**	**7.6**	**7.45**	**2.33**	**7.2**	**35.78**	**7.07**	**2.39**	**8.5**	45.4	5.64	2.48	7.1	**67.7**

Table 1: Performance comparison of retrieval architectures. Best and runner-up in **bold** and underlined. Score and Rank are both LLM-evaluated metrics. Score denotes average LLM-evaluated quality (higher is better), while Rank represents average position (lower is better). HAS denotes Human-Annotated Scores.

Method	Score	Rank
LightRAG	5.81	3.74
PanoramaRAG	**7.58**	**2.62**
w/o topic tree	7.38	2.88
w/o topic nodes	7.35	2.89
w/o two-hop	7.39	2.88

Table 2: Performance of ablated versions on Mix dataset. Best results are in **bold**.

cantly enhances performance across the board. It achieves a harmonious unification of multi-level information, ensuring robust and competitive results on global, fact-intensive, and other challenging tasks alike.

4.5 Ablation Study

To address **RQ2**, we conduct ablation studies on core components of topic tree architecture, as summarized in Table 2. Key findings are as follows.

1) Topic Tree Design and Hierarchy Strategy. In conventional graph-based RAG systems, entity-centric designs often underutilize relational semantics, particularly for abstract queries (e.g., "What is the central theme of Gone with the Wind?"). The Topic Tree architecture addresses this by hierarchically organizing knowledge: high-level nodes (e.g., "central theme") capture query intent and guide retrieval toward semantically relevant sub-graphs. Empirically, on Ultradomain, Topic Tree achieves a 2.7% higher answer score than entity-focused methods by activating contextually appropriate abstraction layers.

2) Topic Node Design. The integration of topic nodes as semantic anchors fundamentally enables cross-community reasoning through bridging structurally disconnected but semantically aligned subgraphs (e.g., linking "Gender Pay Gap" and "Racial Wealth Divide" via Social Justice). Concurrently, it optimizes retrieval efficiency by 3.1%, confirming its necessity for efficient knowledge retrieval and structural organization.

3) 2-Hop Expansion Mechanism. We compare the effect of selecting only the nearest neighbors versus including second-hop neighbors as well. Disabling results in a 2.6% average score reduction, proving its importance for capturing multi-hop relationships and contextual evidence chains.

4.6 Sensitivity Analysis

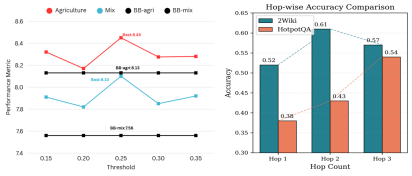

Figure 2: Sensitivity Analysis: Threshold δ (Left) vs. Hop Count (Right). "BB" refers to "Best Baseline".

To address **RQ3**, we evaluate PanoramaRAG's parameter robustness. As shown in Fig. 2, threshold variations (0.15/0.20/0.25/0.30/0.35) yield negligible performance differences, partially attributed to subsequent similarity filtering that mitigates impacts from marginally relevant nodes, and also to embedding both the node names and descriptions simultaneously for similarity computation and filtering. Moreover, under different threshold settings, the performance consistently surpasses the best baseline. For hop-count parameters, considering the complexity of model outputs and the possibility of semantic-equivalent mismatches, we adopt an LLM-based evaluation strategy in which the model determines whether an answer is correct (i.e., outputs true or false).

4.7 Cost Analysis

To address **RQ4**, we benchmark computational efficiency metrics, including time cost and API token consumption. We compare our **PanoramaRAG** with **LightRAG**, which has previously demonstrated significant advantages in efficiency. When using the same vector database for em-

bedding, the retrieval overhead of dense vectors remains within an acceptable range. Although PanoramaRAG incurs a slight increase in end-to-end time due to the latency of the embedding model (less than 3%), it demonstrates better overall performance in terms of the balance between time and retrieval effectiveness in Table 3.

Stage	E2E Time(s)	Avg Entities(/s)
NaiveRAG	19.835	3.227
LightRAG	21.427	5.207
PathRAG	27.136	4.112
PanoramaRAG	22.873	7.652

Table 3: Under the same conditions on Ultradomain, the average time each method takes to answer queries and the average number of entities retrieved per second.

Additionally, we quantify the LLM-based extraction cost in Table 4. For baseline GraphRAG, the dominant cost comes from KG extraction and community-report generation. Let N_{E+R} denote the extracted entities and relations, and $N_{community}$ the detected communities. GraphRAG requires regenerating all community summaries whenever new entities are added, and each community report typically consumes around 5,000 tokens. In contrast, our method introduces an additional topic-tree construction step with size N_T. Empirically, on 2WikiMultihopQA and HotpotQA, we observe $N_T \approx \frac{1}{30} N_{E+R}$, meaning that the added processing is significantly smaller than the baseline KG extraction (about 5%). As the corpus size grows, this ratio becomes even smaller, causing the relative overhead to diminish further.

Baseline	GraphRAG	LightRAG
Tokens	$2 \times 5000 \times N_{community} + O(N_{E+R})$	$O(N_{E+R})$
API Calls	$2 \times N_{community} + O(N_c)$	$O(N_{chunk})$
Method	KG Construction	Topic-tree Construction
Tokens	$O(N_{E+R})$	$O(N_T)$
API Calls	$O(N_{chunk})$	$O(N_{chunk}) + O(N_{cluster})$

Table 4: LLM-based topic extraction cost analysis

4.8 Robustness Evaluation

To assess the robustness of LLM-based scoring, we conducted a multi-round evaluation on Ultradomain, where each question-answer set was scored independently. We used paired *t*-tests to measure the statistical significance of performance differences between our method and various baselines. PanoramaRAG achieves the highest average score and the best top-2 ranking ratio, indicating both superior quality and consistent performance across evaluation rounds. The consistently strong performance of our method across independent evaluation rounds supports the reliability of LLM-based scoring. These results in Table 5 confirm that our method maintains both high accuracy and robustness under repeated judgment conditions.

Method	Score	Agriculture Top2-Ratio	t-stat	p-value
NaiveRAG	8.24	0.47	2.34	1.9×10^{-2}
LightRAG	8.06	0.46	4.95	1.0×10^{-6}
MemoRAG	7.74	0.26	8.65	7.4×10^{-17}
PathRAG	7.21	0.24	12.22	3.4×10^{-30}
PanoramaRAG	**8.44**	**0.56**	-	-
Method	**Score**	**Mix Top2-Ratio**	**t-stat**	**p-value**
NaiveRAG	6.01	0.35	10.40	2.1×10^{-23}
LightRAG	5.60	0.24	12.28	4.0×10^{-31}
MemoRAG	7.02	0.54	3.25	1.2×10^{-3}
PathRAG	5.87	0.25	11.33	4.0×10^{-27}
PanoramaRAG	**7.45**	**0.61**	-	-

Table 5: Performance comparison on Agriculture and Mix datasets by the LLM over five rounds($p < 0.01$).

To verify that our topic extraction and hierarchical clustering modules are model-agnostic, we conducted comparative experiments using two distinct LLM backbones in Table 6. Evaluations on the Mix dataset across six independent runs yielded a negligible performance variance of 1.68% while consistently outperforming all baselines, thereby underscoring the framework's robustness to the choice of LLM backbone.

Model	Average Score	Std. Dev.
DeepSeek R1 0528	7.73	0.038
Qwen2.5-14B-Instruct	7.86	0.027
MemoRAG	7.08	-
PathRAG	5.92	-

Table 6: Performance using different backbones

5 Conclusions

Graph-based RAG techniques face challenges in integrating distributed information for summarization tasks. To provide a global view of the corpus during the indexing, retrieval, and generation stages, we propose a new paradigm, PanoramaRAG. This method introduces a topic-driven hierarchical structure to incorporate higher-level semantic information during panoramic graph construction and designs a topic-tree-based retrieval expansion mechanism. By combining similarity-based scoring with structural filtering in the graph, PanoramaRAG selects informative nodes as inputs to the LLM for answer generation. This approach

facilitates query disambiguation, contextual alignment, and improves the precision, comprehensiveness and diversity of retrieval.

Limitations

In this section, we discuss some limitations of our proposed model and point out potential directions for improvement.First, our model may struggle with fine-grained, detail-oriented multi-hop questions. This issue arises because the detailed descriptions introduced by the panorama graph can sometimes interfere with effective matching. A promising direction to address this limitation is to dynamically adjust the weights of retrieved evidence or to reduce the influence of such detailed descriptions for tasks that require precise reasoning.Second, the model shows limited capability in handling external distracting or ambiguous information. To mitigate this, one possible approach is to leverage a pre-trained LLM to perform preliminary cleaning of newly introduced information before integrating it into the panorama graph, and to align it with similar nodes within the graph for more robust reasoning.

Acknowledgements

Xiting Wang and Meng Li are the corresponding authors. This work was supported by the National Natural Science Foundation of China (NSFC) (NO. 62476279, NO. 92470205NO. U2436209), Major Innovation Planning Interdisciplinary Platform for the Double-First Class Initiative, Renmin University of China, the Huawei-Renmin University joint program on Information Retrieval, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China No. 24XNKJ18. Supported by fund for building world-class universities (disciplines) of Renmin University of China and Public Computing Cloud, Renmin University of China.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jeffrey R Binder and Rutvik H Desai. 2011. The neurobiology of semantic memory. *Trends in cognitive sciences*, 15(11):527–536.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2025. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. *arXiv preprint arXiv:2502.14902*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Mingyue Cheng, Yucong Luo, Jie Ouyang, Qi Liu, Huijie Liu, Li Li, Shuo Yu, Bohou Zhang, Jiawei Cao, Jie Ma, and 1 others. 2025. A survey on knowledge-oriented retrieval-augmented generation. *arXiv preprint arXiv:2503.10677*.

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

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

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

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

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

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and

Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

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

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*.

Xiting Wang Xu Chen Liu, Ruihua Song. 2025. Select, read, and write: A multi-agent framework of full-text-based related work generation.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*, 1.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Karthik Soman, Peter W Rose, John H Morris, Rabia E Akbas, Brett Smith, Braian Peetoom, Catalina Villouta-Reyes, Gabriel Cerono, Yongmei Shi, Angela Rizk-Jackson, and 1 others. 2024. Biomedical knowledge graph-optimized prompt generation for large language models. *Bioinformatics*, 40(9):btae560.

Viju Sudhi, Sinchana Ramakanth Bhat, Max Rudat, and Roman Teucher. 2024. Rag-ex: A generic framework for explaining retrieval augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2776–2780.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

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*.

Qinhan Yu, Zhiyou Xiao, Binghui Li, Zhengren Wang, Chong Chen, and Wentao Zhang. 2025. Mramg-bench: A comprehensive benchmark for advancing multimodal retrieval-augmented multimodal generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3616–3626.

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*.

Wan Zhang and Jing Zhang. 2025. Hallucination mitigation for retrieval-augmented large language models: a review. *Mathematics*, 13(5):856.

Weijieying Ren Lu Jiang Dongjie Wang Kunpeng Liu Zhang, Xiting Wang. 2025. Ratt: A thought structure for coherent and correct llm reasoning.

Kun Zhu, Xiaocheng Feng, Xiyuan Du, Yuxuan Gu, Weijiang Yu, Haotian Wang, Qianglong Chen, Zheng Chu, Jingchang Chen, and Bing Qin. 2024. An information bottleneck perspective for effective noise filtering on retrieval-augmented generation. *arXiv preprint arXiv:2406.01549*.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025. Knowledge graph-guided retrieval augmented generation. *arXiv preprint arXiv:2502.06864*.

A Experimental Details

To enhance the reproducibility of our work, we provide detailed experimental settings and procedural steps below.

A.1 Configurations

In our experiments, we utilize **text-embedding-3-small** as the default embedding model across all experiments. For topic tree construction, answer generation, and automated evaluation, we employ **DeepSeek R1 0528**. To facilitate efficient similarity-based retrieval and accelerate the matching and ranking process, we use the **NanoVectorDB** vector database to store and index embeddings of entities, relations, and topics. Additionally, we set the number of clusters to **50** for upper topics and **200** for lower topics and the maximum keyword set size to **10**.

A.2 Indexing

The system supports flexible input formats, including both unstructured text and structured JSON data. An LLM is employed to extract all entities and relations from the corpus. Notably, "relations" in our framework encompass not only traditional knowledge graph edges but also higher-level semantic associations. Furthermore, the LLM generates a concise description for each extracted element; these descriptions are embedded jointly with the nodes to improve the accuracy of semantic matching. Building upon this foundational extraction, the system further organizes the corpus into a hierarchical topic tree. Specifically, the LLM identifies core topic-description pairs from partitioned text chunks, which serve as the initial leaf nodes. These localized topics are then synthesized through a bottom-up abstraction process to form a multi-layered hierarchy. This structure enables the framework to capture the global thematic landscape of the dataset at varying levels of granularity.

A.3 Query and Retrieval

For each query, the LLM initially generates a set of keywords, which are matched against the topic tree using cosine similarity. These keywords then undergo vertical expansion within the treetraversing both upward and downwardto capture semantically relevant parent and child nodes. After filtering, we obtain a refined set of multiscale keywords. The final keyword sets are used to retrieve dense embeddings of entities and relations from the knowledge graph. Leveraging the graph structure, we collect both first-hop and second-hop neighbors. We kemudian filter redundant or irrelevant nodes and truncate the results to fit within the LLM's context window. This curated context is finally passed to the LLM for response generation.

A.4 Evaluation

To optimize evaluation efficiency for large-scale datasets, we implement parallel LLM calls for scoring. To mitigate position bias (where LLMs may favor answers presented earlier in the prompt), we randomly shuffle the order of candidate answers during each evaluation run. A model index list is maintained to ensure that scores are accurately mapped back to their original sources for consistent aggregation. LLM evaluation is based on two key aspects:

- Alignment with the gold standard answer: This is the primary criterion for datasets like 2WikiMultihopQA and HotpotQA, where answers are concrete and concise.

- Clarity, Conciseness, Comprehensiveness, and empowerment: For more open-ended datasets like UltraDomain, additional criteria are used to assess answer quality beyond factual correctness.

A.5 Statistical Information

Table 7 presents the statistical profiles of the panoramic graphs generated by PanoramaRAG across two multi-hop, detail-oriented datasets (2WikiMultihopQA and HotpotQA) and the more challenging LongBench dataset. These statistics underscore PanoramaRAGs capacity to manage large-scale data; while HotpotQA involves approximately 5,0000 entities and relations, the LongBench graphs utilized in our study scale to over 300,000representing a medium-to-large-scale setting. Our framework maintains robust performance under these demanding conditions, further validating its scalability.

Dataset	Entity Count	Relation Count
2WikimultihopQA	17945	10438
HotpotQA	32711	23793
LongBench	149,863	162049

Table 7: Statistical information of panoramic graphs

B Graph-based RAG Analysis

Current prominent Graph-based RAG approaches, frequently employed as strong baselines, primarily fall into two categories: triplet-based retrieval, exemplified by HippoRAG, and summary-based structures, represented by GraphRAG. While HippoRAG excels on multi-hop datasets such as 2WikiMultihopQA and HotpotQA, its performance often stagnates in query-focused summarization tasks or scenarios requiring integrated global information. Furthermore, fundamental distinctions exist between GraphRAG and our PanoramaRAG: GraphRAGs community summaries and PanoramaRAGs topic tree differ significantly in both construction principles and semantic granularity:

(1) Construction Principle: GraphRAG relies on topological community detection, which may overlook latent semantic groupings. In contrast, PanoramaRAG utilizes semantic clustering to uncover meaningful associations that transcend mere lexical or structural proximity.

(2) Semantic Granularity: GraphRAG generates verbose, heterogeneous community summaries (avg. 400-600 words) that indiscriminately aggregate disparate entities, relations, and claims. Conversely, our Topic Tree is optimized for efficient vector retrieval: leaf nodes are concise (10-20 words), and high-level nodes are capped at 200 words. Crucially, each topic node and its description are jointly embedded into a semantic vector for direct use during keyword expansion, providing explicit and fine-grained semantic guidance.

C Case Study

C.1 Success Case Analysis

To demonstrate our framework's superior performance in query-focused summarization, we select a representative query from the Mix subset of the Ultradomain dataset for a detailed case study, which is *"What were some of the significant achievements of the horse Charisma?"* (Correct Answer: Charisma, ridden by New Zealander Mark Todd, won many competitions in the sport of eventing and is considered one of the greatest event horses ever. Some of his significant achievements include winning individual gold at the 1984 and 1988 Olympics, winning the Luhmuhlen, Germany Three-Day Event in 1986, and being voted the best NZ Sport Horse in the world in 1983 and 1985.)

PathRAG (Failure): Charisma was a notable Thoroughbred racehorse ..... securing victories in several prestigious races. Notably, this includes the Kentucky Derby ..... successful career in breeding stallions .....

The system correctly detects the entity "Charisma" but fails to navigate the appropriate subgraph. Because the corpus contains multiple horses with similar names, it retrieves passages about an unrelated "Thoroughbred racehorse" and produces answers involving events such as the Kentucky Derby. This failure stems from the absence of global semantic grouping: without topic-level context, the model cannot distinguish between semantically unrelated but lexically similar entities, leading to retrieval drift.

PanoramaRAG (Success): Charisma ... ridden by New Zealand equestrian Mark Todd, renowned for its remarkable achievements in the world of eventing ..... winning two Olympic gold medals at the 1984 and 1988 Summer Olympics ..... celebrated as one of the greatest eventing horses in history .....

Our method expands key phrases such as significant achievements and Charisma and identifies high-level topic nodes including Charisma (horse) biography and competitive career, supported by nodes like Champions and their careers and Races won. These global semantic cues guide retrieval toward Charismas eventing career, producing the correct answer: Charisma won two Olympic gold medals in 1984 and 1988.

C.2 Failure Analysis

We provide a representative example that illustrates both the strength of our approach and the circumstances under which it may underperform. Consider the query: *"Describe the transformation of the Witch of Atlas' mother."* (Correct Answer: vapor cloud meteor star).

PanoramaRAG consistently excels at capturing high-level semantic structures. In this case, it correctly identifies the entity "Witch of Atlas" and successfully maps it to high-level topic nodes such as "Symbolic Connection of Atlantides". Consequently, the generated answer emphasizes a broader thematic interpretation: "The transformation... can be perceived metaphorically, reflecting the enduring legacy of beauty and the intricate dynamics of motherhood." PanoramaRAG's strong ability to leverage global semantic cuesan ability that simple retrieval methods do not possess.

However, because the query requires an exact sequence of physical transformations, NaiveRAG by directly retrieving the literal text spanproduces the fully correct factual sequence. PanoramaRAG, in contrast, provides a partially correct but thematically enriched interpretation. This phenomenon stems from a granularity mismatch, rather than information loss. PanoramaRAG preserves all factual details by using corpus-derived descriptions for entities and relations. This issue arises because the abundance of high-quality global semantic cues can sometimes overshadow these low-level details. The LLM, seeing rich thematic context, may overemphasize semantical information rather than a specific factual trace, leading to answer drift.

Notably, this is not a hallucination issue or a misleading-evidence problem. We sampled 30 queries for which the KG contained no relevant answers. Although the retrieval system returned partially matched entities, the LLM was not misled into fabricating false answers in any of these cases. This indicates that PanoramaRAG is robust to noise and that the observed drift stems from preference for global semanticsnot from being misled by irrelevant content.

To resolve the granularity mismatch in detail-heavy tasks, we suggest a clear path for improvement. Chain-of-Thought (CoT) prompting may help enforce step-by-step grounding and encourage the LLM to focus on fine-grained entity evidence when the task demands specificity. Query Decomposition could guide the model to process detailed sub-questions first, reducing the tendency to default to high-level thematic interpretations. These strategies explicitly strengthen the causal relationship between the query constraints and the output, guiding the LLM to prioritize entity evidence over high-level topics when specific details are requested.