KHL-RAG: Enhancing Retrieval-Augmented Generation via Dual-Graph Mechanism based on Rhetorical Structure Theory

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation has become a dominant paradigm for improving the reasoning reliability of large language models by grounding generation in external knowledge. However, existing approaches remain limited by unstable vector-based retrieval and the restricted coverage of entity-centric knowledge graphs. These limitations arise from the absence of paragraph-level discourse modeling. We propose KHL-RAG, a dual-graph RAG framework that integrates a knowledge graph with a heterogeneous logical paragraph graph grounded in Rhetorical Structure Theory. By explicitly modeling implicit discourse relations and adopting a Dual-Semantic Vector Filtering and query-aware aggregation strategy, KHL-RAG improves retrieval robustness and contextual completeness. Extensive experiments demonstrate that KHL-RAG consistently reduces hallucinations in complex reasoning tasks, achieving up to a 3.82% improvement in answer correctness over state-of-the-art RAG baselines.

1 Introduction

Large Language Models continue to face challenges (Ren et al., 2024; Guo et al., 2024a) related to knowledge staleness, content hallucination, and multi-hop reasoning. Retrieval-Augmented Generation has emerged as a pivotal technology (Lewis et al., 2020; Gao et al., 2023), addressing these challenges by leveraging flexible external knowledge bases. By enabling easily updatable knowledge, verifiable content, and domain-specific information, RAG has become a key research direction.

Early RAG systems predominantly treat external knowledge as flat collections of unstructured text chunks and retrieve context via cosine similarity in embedding space. While simple and scalable, this paradigm is inherently brittle: it assumes that semantic relevance can be reliably approximated by surface-level vector similarity. To address this limitation, more recent studies (Edge et al., 2024; Luo et al., 2024; Jiang et al., 2023; Guo et al., 2024b) restructure knowledge into entity-centric knowledge graphs, enabling structured retrieval and reasoning over explicit entity relations.

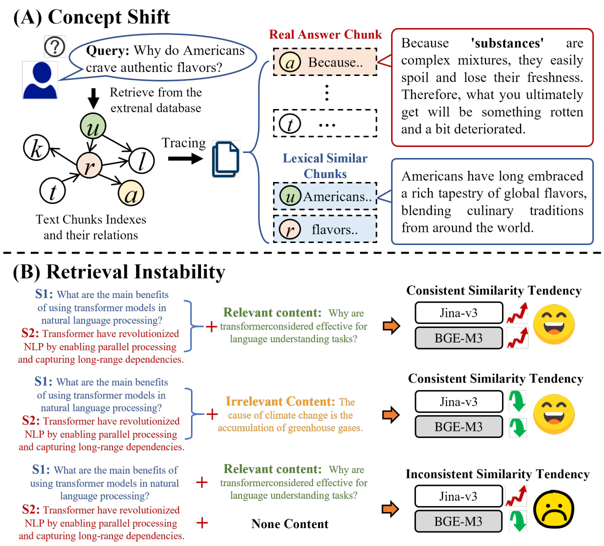


Figure 1: **(A)** For the culinary query "Why do Americans crave authentic flavors?", conventional retrieval methods primarily rely on entity-level keyword matching (e.g., "Americans, authentic, flavors"). As a result, they fail to retrieve a semantically critical preceding paragraph that discusses "curry powder," which provides the actual causal explanation despite being lexically unrelated to the query. **(B)** The similarity scores assigned to the same sentence pair can differ substantially across text encoder types, leading to unstable retrieval rankings and inconsistent evidence selection.

However, existing RAG approaches largely fall into two disjoint design philosophies, each with a distinct but critical limitation, as illustrated in Fig. 1. (1) **Vector-centric RAG** methods rely on dense similarity computations in embedding space, making retrieval results highly dependent on the specific embedding model. Due to substantial differences in how encoders model semantic alignment, the same pair of texts may yield markedly different similarity scores across modelsfor example, low similarity under BGE but high similarity under Jina. (2) **Entity-centric KG-based RAG** methods emphasize retrieval around explicit entities and their relations. While this strengthens structured semantic constraints, it also limits coverage. In particular, paragraphs that are highly relevant in terms of semantic reasoning or discourse logic but lack explicit entity keywords are often missed, leading to information loss.

Crucially, these limitations stem not from insufficient modeling capacity, but from a shared structural blind spot: both paradigms overlook paragraph-level discourse logic. As a result, implicit relationshipssuch as causal explanations, illustrative elaborations, or contrastive reasoningremain largely inaccessible to current retrieval mechanisms. This makes existing RAG systems particularly fragile in complex reasoning scenarios where relevant knowledge is distributed across semantically connected but lexically or entity-wise distant passages.

We argue that addressing retrieval instability and concept shift requires a structural rethinking of how contextual knowledge is represented and retrieved. Specifically, paragraph-level rhetorical and logical relations must be made explicit, computable, and retrievable, rather than being implicitly encoded in dense vectors or discarded during entity extraction. This insight motivates the integration of implicit discourse logic with explicit entity semantics, rather than treating them as competing alternatives.

To address these issues, we propose KHL-RAG, an architecture that integrates a knowledge graph with a heterogeneous logical paragraph graph. Our key contribution lies in introducing Rhetorical Structure Theory(Mann and Taboada, 2006) into the text attribute graph, enabling the construction of a heterogeneous logical paragraph graph that captures implicit discourse-level relations between paragraphs. This graph compensates for the coverage limitations of entity-based retrieval and complements the knowledge graph.

At the retrieval stage, knowledge graph retrieval follows existing paradigms. For the heterogeneous logical paragraph graph, we propose a Dual-Semantic Vector Filtering mechanism: it first retrieves target paragraphs based on semantic similarity, and then identifies contextually complementary paragraphs via first-order neighbor semantic relations. Compared with pure similarity-based retrieval, this approach explicitly incorporates paragraph-level discourse logic.

At the aggregation stage, since both graph structures contribute contextual evidence, KHL-RAG dynamically coordinates the proportion of retrieved contexts according to query type. For simple factoid questions, a higher proportion of knowledge-graph context is emphasized, whereas for complex logical reasoning queries, greater weight is assigned to the heterogeneous logical paragraph graph.

Our contributions are summarized as follows:

- We propose KHL-RAG, a dual-graph Retrieval-Augmented Generation framework. In particular, we construct a heterogeneous logical paragraph graph grounded in Rhetorical Structure Theory, which introduces paragraph-level semantic logic beyond similarity-based retrieval.
- At the retrieval and aggregation stages, we propose a Dual-Semantic Vector Filtering mechanism and a query-aware strategy that dynamically adjusts the proportion of retrieved context according to different query types.
- Extensive experiments on four question-answering benchmarks demonstrate that KHL-RAG consistently outperforms state-of-the-art RAG variants, especially in complex reasoning scenarios.

2 Preliminaries

We define the following key concepts related to this work: knowledge graph context, heterogeneous logical paragraph graph context, and the corresponding RAG Task Objective.

Knowledge Graph Context. Given contextual document information C_{text}, we extract a graph structure $G_{\text{KG}} = (N_{\text{KG}}, E_{\text{KG}})$, where N_{KG} denotes a set of entities as graph nodes, and E_{KG} represents the relations among these entities as edges. For a given query q, we identify the entities it contains and match them to entities in the knowledge graph G. The Knowledge Graph Context C_{KG} is defined as the set of relevant entity-relation triples.

Heterogeneous Logical Paragraph Graph. Using the same context C_{text}, we apply a chunking strategy to obtain a set of paragraph chunks $\{C^i_{\text{para}}\}_{i=1}^{d}$. From these, we construct a graph $G_{\text{HLPG}} = (N_{\text{HLPG}}, E_{\text{HLPG}})$, where N_{HLPG} consists of text paragraphs enriched with keywords, main

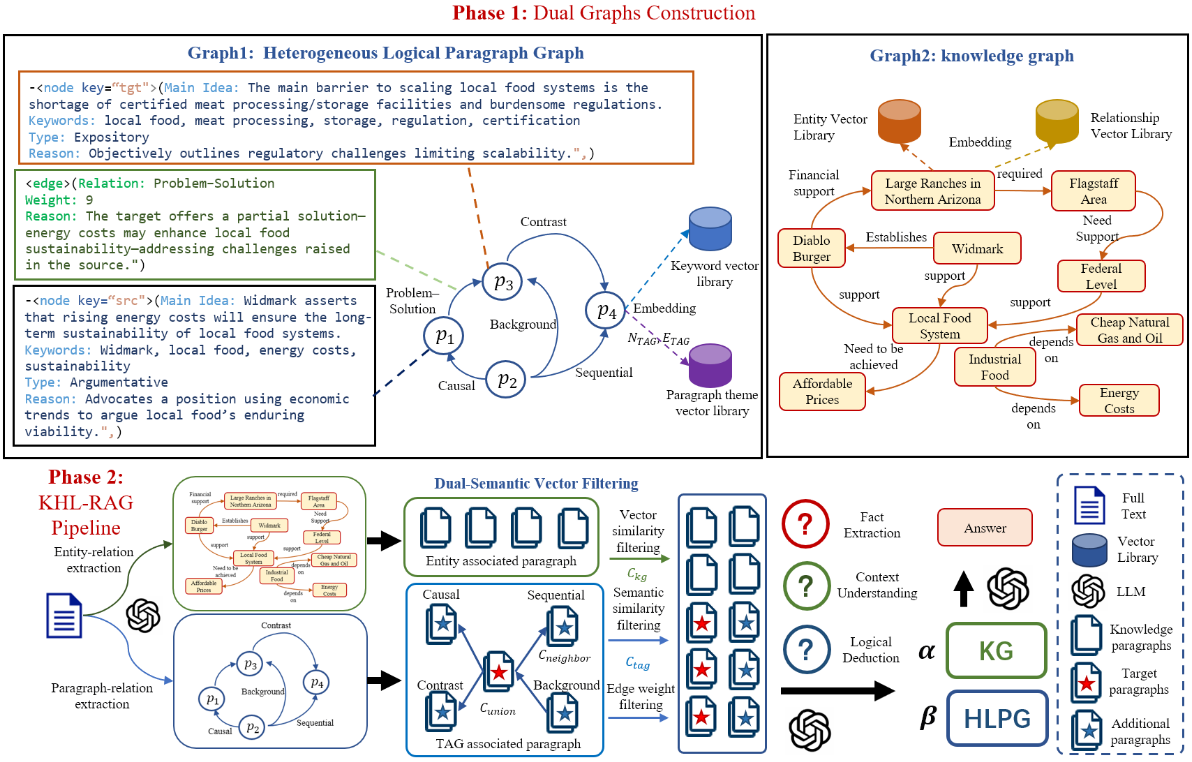


Figure 2: The overall architecture of KHL-RAG. Primarily comprising the stages of knowledge base construction and retrieval.

ideas, etc., and E_{HLPG} encodes logical relations between them. For a given query q, we identify related paragraphs, and the Heterogeneous Logical Paragraph Graph context C_{HLPG} comprises the associated paragraphlogical relation pairs.

RAG Task Objective. Given a query q and context C_{text}, standard RAG retrieves key textual evidence C_m from C_{text}. A large language model then generates an answer based on q and C_{main}, formally represented as:

$$P(y \mid q) = \sum_{C_m \in C_{\text{text}}} P(y \mid q, C_m) P(C_m \mid q, C_{\text{text}}) \tag{1}$$

3 Methodology

KHL-RAG integrates a heterogeneous logical paragraph graph with a dual-semantic vector filtering mechanism. The paragraph-level logical graph overcomes the limitations of entity-centric retrieval by capturing implicit contextual associations, while the dual-semantic filtering removes question-irrelevant content, thereby steering the LLM toward more precise and focused responses.

3.1 KHL-RAG Architecture Overview

The overall framework of KHL-RAG is illustrated in Figure 2. The process consists of two main stages:

Phase 1: Dual Graphs Construction. This stage aims to build relevant knowledge bases from user-provided context. First, the complete body of knowledge text is segmented into paragraph chunks. Then, guided by Rhetorical Structure Theory, each paragraph is annotated with its type, main idea, and extracted keywords. Logical relationships between paragraph pairs are computed based on these attributes, resulting in a complete heterogeneous logical paragraph graph. This graph is stored as a directed graph, while keyword and main idea representations are embedded and stored in a vector database.

Phase 2: KHL-RAG Pipeline. In this stage, relevant context is retrieved and processed to enable accurate response generation by a large language model. The query is first semantically parsed to extract its main idea, keywords, entities, and relations. The main idea and keywords are then used to retrieve the Top-K most similar paragraphs from the vector database, after which high-weight first-order neighbors are selected and merged to form the heterogeneous logical paragraph graph context. Simultaneously, entity and relation embeddings query the knowledge graphs vector database to obtain the Top-K associated paragraphs. The query is subsequently classified as Fact-extraction, Context-understanding, or Logical-deduction to dynamically determine the contribution of each context source. Finally, the query and curated context are input to the language model to generate a precise response.

3.2 Heterogeneous Logical Paragraph Graph

To capture paragraph-level latent logical relations, we propose the heterogeneous logical paragraph graph. *Extraction of Inter-Paragraph Relations* To construct the heterogeneous logical paragraph graph G_{HLPG}, the input context C_{text} is first segmented into paragraph-level chunks. For a balance between efficiency and representational fidelity, a fixed-size chunking strategy is employed, producing the paragraph set $\{C^i_{\text{para}}\}_{i=1}^d$. Subsequently, the graph structure is instantiated, and key elements are extracted using a large language model. Paragraph types and semantic roles are analyzed to establish logical links, with inter-paragraph relations guided by RST (Mann and Taboada, 2006). Paragraph categories and relation types are defined to facilitate reasoning over complex discourse structures. The resulting graph G_{HLPG} comprises node information N_{HLPG}, representing paragraph-level semantic embeddings, and edge information E_{HLPG}, capturing inter-paragraph relation types.

Node Definition. Given the paragraph set $\{C^i_{\text{para}}\}_{i=1}^d$, we extract node features for each paragraph as $N^i_{\text{HLPG}} = \{n^i_{\text{main}}, n^i_{\text{keyword}}, n^i_{\text{type}}, n^i_{\text{reason}}\}$, representing the main idea, keywords, paragraph type, and rationale for the type assignment. Paragraph types are classified into seven categories: narrative, descriptive, argumentative, expository, lyrical, narrative-argumentative, and descriptive-narrative.

Edge Definition. For any two paragraphs $C^{\text{src}}_{\text{para}}$ and $C^{\text{tgt}}_{\text{para}}$, we extract their relational features as $E^k_{\text{HLPG}} = \{e^k_{\text{type}}, e^k_{\text{reason}}, e^k_{\text{weight}}\}$, denoting the relation type, its rationale, and a confidence-weight score. Drawing on core-satellite and multinuclear structures from RST (Mann and Taboada, 2006), we define 16 relation types applicable at the paragraph level: None, Evidence, Elaboration, Background, Purpose, Result, Cause, Comparison, Concession, Sequence, Summary, Evaluation, Problem-Solution, Conclusion, Topic, and Example. The confidence score $e^k_{\text{weight}} \in (0, 10]$ reflects the strength of inferred relation.

Information Extraction with Large Language Models. Based on the above graph structure, we design prompt templates p_n for node extraction and p_e for edge extraction (Mei et al., 2025). These are used with a large language model F_{LLM} to extract d node entries and m edge entries:

$$N_{\text{HLPG}} = F_{\text{LLM}}\left(N^i_{\text{HLPG}} \middle| p_n, C^i_{\text{para}}\right)_{i=1}^d, \quad (2)$$

$$E_{\text{HLPG}} = F_{\text{LLM}}\left(E^k_{\text{HLPG}} \middle| p_e, C^i_{\text{para}}, C^j_{\text{para}}\right)_{i\neq j}^d. \quad (3)$$

For every pair of distinct paragraphs, the model determines potential edge relations. Multiple edges may exist between a pair; only those with confidence scores above a threshold are retained.

3.3 Dual-Semantic Vector Filtering

To enable retrieval over the heterogeneous logical paragraph graph, we design a dual-semantic vector filtering mechanism. Specifically, we design prompts p_{keyword} and p_{main} to extract the keyword set and main idea from the user query q via a large language model:

$$Q_{\text{keywords}} = \{u_1, u_2, ..., u_d\}\ LLM\left(q, p_{\text{keyword}}\right), \quad (4)$$

$$Q_{\text{main}} = LLM(Q, p_{\text{main}}). \quad (5)$$

We then use the same encoder $\mathbb{F}_{\text{emb}}$ as in the graph construction to embed the extracted query information into vector representations E_k and E_m, and retrieve the Top-K most similar paragraph IDs from the vector databases:

$$C_k = Top_k\left(sim\left(\mathbb{F}_{\text{emb}}\left(q_{\text{keywords}}\right), h_{\text{keyword}}\right)\right), \quad (6)$$

$$C_v = Top_k\left(sim\left(\mathbb{F}_{\text{emb}}\left(q_{\text{main}}\right), h_{\text{main}}\right)\right). \quad (7)$$

Here, sim denotes the cosine similarity function. The overall retrieved context set is the union of the two sets: $C_{\text{union}} = C_k \cup C_v$. To enrich the context with relevant supplemental knowledge, we identify first-order neighbors of the target paragraphs and retain only those edges whose weights exceed a threshold $\tau \in (0, 10]$,

$$C_{\text{HLPG}} = \mathbb{F}_{\text{filter}}(C_{\text{union}} \cup C_{\text{neighbor}} | e^k_{\text{weight}} < \tau). \quad (8)$$

Here, $\mathbb{F}_{\text{filter}}$ denotes the filtering function, and e^{k}_{weight} represents the edge weight between the paragraph C_{union} and its k-th neighbor in C_{neighbor}.

3.4 Dual-Graph Context Aggregation

For two complementary graph data structures, we propose a dual-graph context aggregation method to adaptively integrate information from the two complementary graph structures. Given the diversity of reasoning demands, we classify user queries into three categories based on complexity: fact QA, context-understanding, and logical deduction, as shown in Table 1.

Entity-level retrieval via the knowledge graph is more effective for fact QA-type queries, while the heterogeneous logical paragraph graph excels at capturing discourse logic for complex reasoning tasks. Leveraging these complementary strengths, we design a prompt p_{type} to classify the user query q:

$$T_q = LLM\left(q, p_{\text{type}}\right), \tag{9}$$

where T_q denotes the predicted query type. Based on this classification, we adaptively combine contexts from both sources:

$$C_{\text{KHL}} = C_{\text{KG}}(m) + C_{\text{HLPG}}(n), \tag{10}$$

where m and n refer to the top-m retrieved segments from the knowledge graph and the top-n from the heterogeneous logical paragraph graph, respectively. Finally, the combined context C_{KHL} is integrated into a retrieval-augmented prompt p_{query}, and the large language model generates the final response using successful exemplars p_{exa}:

$$y^* \sim LLM(Q, C_{\text{KHL}}, p_{\text{query}}, p_{\text{exa}}), \tag{11}$$

where y^* is LLM generated response.

3.5 Graph Extension Mechanism

To support dynamic knowledge updates and incremental learning, we propose a heterogeneous logical paragraph graph extension method. Specifically, when p_{new} arrives, it is added as a new node N_{new}. We then compute the logical relevance between N_{new} and each existing paragraph node $N_{\text{i}} \in N_{\text{HLPG}}$, using predefined similarity metrics and structured dependency rules (e.g., semantic or rhetorical relations). Based on the estimated relevance strength, heterogeneous logical edges $E_{\text{new}}, N_{\text{i}})$ are constructed and appended to E_{HLPG}.

This extension process enables the paragraph graph to continuously preserve global context and long-range logical coherence between new and existing knowledge, providing an up-to-date structured foundation for subsequent retrieval and reasoning.

4 Experiment

We compare our approach with current mainstream methods and address the following research questions: **RQ1**: Is the KHL-RAG method more effective than previous approaches? **RQ2**: Does the proposed heterogeneous logical paragraph graph prove to be effective? **RQ3**: How do the parameters within KHL-RAG influence the performance of different large language generation models?

4.1 Experimental Setting

4.1.1 Dataset

Existing QA benchmarks focus on shallow semantic matching, often solvable via surface-level similarity, and thus inadequately assess complex reasoning. To address this, we adopt a reasoning-depthaware dataset design, separating questions by inference complexity. KHL-RAG is evaluated on four domain-specific datasetsAgriculture, Computer Science, Legal, and Cookingeach with Original, Easy, Medium, and Hard variants, providing a fine-grained view of model performance across reasoning depths (Table 1). Dataset prompts are provided in the appendix 7.

4.1.2 Baseline Methods

We compare KHL-RAG with representative baselines, including StandardRAG (Gao et al., 2023), GraphRAG (Edge et al., 2024), and LightRAG (Guo et al., 2024b). In addition, we include two recent structure-aware variants: CG-RAG (Hu et al., 2025), which enhances retrieval via coarse-grained graph modeling, and HyperGraphRAG (Luo et al., 2025), which captures high-order semantic dependencies through hypergraph-based reasoning.

4.1.3 Evaluation Metrics

Following evaluation methodologies from RAGAS(ES et al., 2024) and OG-RAG(Sharma et al., 2024), we employ the following metrics: Context Recall (C-Rec), Answer Similarity (A-Sim), Answer Correctness (A-Corr), and Answer Relevance (A-Rel). Details are in the appendix A.1.

Table 1: Question Type Matrix Information.

Dimension	Easy	Medium	Hard
Citation Length	1-2 sentences	2-3 sentences	3-5 sentences
Question Type	Fact-extraction	Context-understanding	Logical-deduction
Answer Source	Direct extraction	Local reassembly	Global inference
Inference Steps	0 steps	1 step	≥ 2 steps
Logic Chain Length	None	Single chain	Multiple, intersecting chains
Typical Prompt Words	What/When/Who	Why/How	If/Suppose
Answer Length Limit	5-15 words	15-30 words	30-50 words

Table 2: Performance of the KHL-RAG Model on the Hard-Type Dataset.

Model	Agriculture				CS			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
StandardRAG	16.52 ±1.01	21.42 ±0.92	35.73 ±1.55	13.12 ±0.67	21.79 ±1.24	26.11 ±1.17	37.89 ±0.81	22.42 ±1.05
GraphRAG	33.34 ±0.55	36.73 ±1.09	54.50 ±1.28	42.12 ±1.59	38.86 ±0.96	45.76 ±1.33	61.59 ±0.77	54.48 ±1.45
LightRAG	34.20 ±1.22	45.80 ±0.91	74.30 ±1.15	41.90 ±1.08	42.10 ±0.72	51.70 ±1.39	78.40 ±1.18	50.20 ±0.88
CG-RAG	41.80 ±0.92	49.60 ±1.17	77.40 ±1.09	43.80 ±0.87	49.30 ±1.10	54.20 ±1.31	81.20 ±0.91	54.10 ±1.04
HyperGraph	42.50 ±0.78	50.10 ±1.22	77.80 ±0.93	44.00 ±0.81	50.80 ±1.05	54.50 ±1.38	81.50 ±0.85	54.70 ±1.12
KHL-RAG	**48.73** ±0.66	**52.93** ±1.49	**79.67** ±1.03	**44.89** ±0.80	**55.01** ±1.51	**55.61** ±1.36	**83.41** ±0.71	**56.58** ±1.26

Model	Legal				Cooking			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
StandardRAG	18.77 ±1.47	24.07 ±0.91	40.98 ±1.58	16.62 ±0.74	28.53 ±1.08	32.13 ±1.37	49.89 ±0.99	28.32 ±1.21
GraphRAG	36.50 ±0.62	39.40 ±1.11	60.60 ±0.87	45.80 ±1.46	47.20 ±1.14	52.10 ±1.35	71.20 ±0.69	57.20 ±0.93
LightRAG	38.40 ±1.18	48.20 ±0.85	78.90 ±1.02	45.10 ±1.40	49.80 ±0.83	57.30 ±1.21	85.10 ±1.32	56.80 ±0.91
CG-RAG	45.70 ±1.05	51.90 ±0.94	80.70 ±1.03	46.50 ±1.06	53.80 ±0.84	59.60 ±1.29	87.90 ±1.05	58.90 ±0.89
HyperGraph	46.30 ±0.91	52.40 ±1.01	80.90 ±0.95	46.80 ±0.92	54.10 ±0.76	60.00 ±1.35	88.20 ±0.83	59.40 ±0.97
KHL-RAG	**50.93** ±1.56	**54.83** ±0.90	**81.97** ±1.20	**47.69** ±1.00	**57.22** ±0.64	**60.98** ±1.54	**89.14** ±1.02	**61.27** ±1.10

4.2 Experimental Results

4.2.1 Main Experimental Results (RQ1)

To validate KHL-RAG on LLM-based question answering (RQ1), we evaluated its performance on the Original dataset across three difficulty levels: fact-extraction (Easy), context-understanding (Medium), and logical-deduction (Hard). Results are averaged over multiple runs. As shown in Table 2, KHL-RAG consistently outperforms all baselines, including GraphRAG, LightRAG, CG-RAG, and HyperGraphRAG. Compared with LightRAG, it achieves improvements of up to 10.11% in C-Rec, 4.47% in A-Sim, 2.09% in A-Rel, and 3.82% in A-Corr. Notably, KHL-RAG surpasses HyperGraphRAG and CG-RAG, demonstrating that integrating paragraph text-attribute graphs with knowledge graphs enables more effective retrieval and multi-hop reasoning, particularly for medium- and hard-level tasks.

KHL-RAG extends retrieval along logical edges, enabling hierarchical, logic-driven expansion that enhances contextual coverage and alignment with reference answers. While traditional KG-based retrieval relies on entityrelation triplets and is limited to explicitly encoded associations, KHL-RAG captures richer semantic relationships, improving coherence and filtering noise. This approach achieves notable gains across multiple metrics, especially in complex reasoning over extended texts.

4.2.2 Ablation Study Results (RQ2)

To evaluate the contributions of KHL-RAGs core components, as shown in Table 3, we conducted ablation experiments on the heterogeneous logical paragraph graph (HLPG) and the dual-semantic vector filtering (DPVF) mechanism.

Starting from the baseline StandardRAG, adding a knowledge graph (+KG) significantly improves performance on fact-centric questions by leveraging entity-level semantics. Replacing or complementing it with HLPG (+HLPG) further boosts performance, particularly on Medium- and Hard-type queries, by capturing multi-paragraph logical and contextual relations beyond explicit entities. Incorporating DPVF (+HLPG+DPVF) mitigates noise from excessive candidate paragraphs, enhancing context relevance and overall metrics. The full KHL-RAG model, combining HLPG and DPVF, consistently achieves the highest scores across all

Table 3: Results of the ablation study for the HLPG component on the Agriculture dataset, shown by Hard-Type level.

Model	Original				Easy			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
StandardRAG	12.23 ±0.95	16.13 ±1.12	24.91 ±0.83	10.87 ±1.47	18.82 ±1.01	24.82 ±1.35	38.33 ±0.92	16.72 ±1.28
+KG	26.05 ±1.06	34.02 ±1.19	51.92 ±0.88	31.11 ±1.44	40.07 ±0.97	52.26 ±1.31	79.88 ±0.80	47.88 ±1.04
+HLPG	41.86 ±1.41	46.12 ±0.88	72.91 ±1.06	37.45 ±1.19	39.89 ±0.97	46.74 ±1.44	75.43 ±1.31	43.21 ±0.80
+HLPG+DPVF	48.73 ±1.25	52.93 ±0.85	79.67 ±1.04	44.89 ±1.49	43.33 ±1.08	54.33 ±0.94	81.17 ±1.22	48.49 ±1.17
KHL-RAG	**50.68** ±1.33	**54.75** ±0.90	**81.42** ±1.10	**46.53** ±0.82	**45.98** ±1.46	**56.12** ±1.05	**83.84** ±1.29	**50.25** ±0.88

Model	Medium				Hard			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
StandardRAG	21.12 ±1.33	26.02 ±0.90	40.33 ±1.10	17.72 ±0.82	16.52 ±1.01	21.42 ±0.92	35.73 ±1.55	13.12 ±0.67
+KG	42.17 ±1.39	53.26 ±0.86	82.18 ±1.52	48.78 ±1.00	37.57 ±1.30	48.66 ±0.84	77.58 ±1.07	44.18 ±1.19
+HLPG	42.57 ±1.42	49.88 ±1.16	78.95 ±1.03	39.84 ±0.99	42.15 ±1.24	44.23 ±0.91	74.82 ±1.37	35.47 ±1.14
+HLPG+DPVF	46.33 ±1.39	57.53 ±0.86	84.27 ±1.52	49.49 ±1.00	46.21 ±1.11	50.19 ±0.81	78.15 ±1.48	43.37 ±1.07
KHL-RAG	**48.17** ±1.21	**59.30** ±1.04	**86.02** ±0.84	**51.33** ±1.39	**48.73** ±0.66	**52.93** ±1.49	**79.67** ±1.03	**44.89** ±0.80

difficulty levels, demonstrating that integrating entity-level precision with paragraph-level structural reasoning substantially enhances retrieval-augmented question answering.

4.2.3 Parameter Analysis Experiments (RQ3)

In this section, we analyze the key parameters of KHL-RAG (RQ3), examining how different parameter settings for each component affect the performance of the large models responses.

Analysis of Neighbor-Relation Weights τ. In the heterogeneous logical paragraph graph, the supplementary information provides effective knowledge beyond the target paragraphs. To investigate how KHL-RAG can efficiently filter these additional passages, we conducted experiments on the weight threshold for selecting logically related neighbors. As illustrated in Fig. 3 (a,b,c,d), we observe that with increasing question difficulty, a threshold set too low admits excessive noiseweakly related paragraphs that adversely affect the LLMs outputwhereas a threshold set too high yields no valid supplementary passages, thus failing to leverage the graphs advantage. Empirically, we find that an optimal threshold lies around a value of 6.

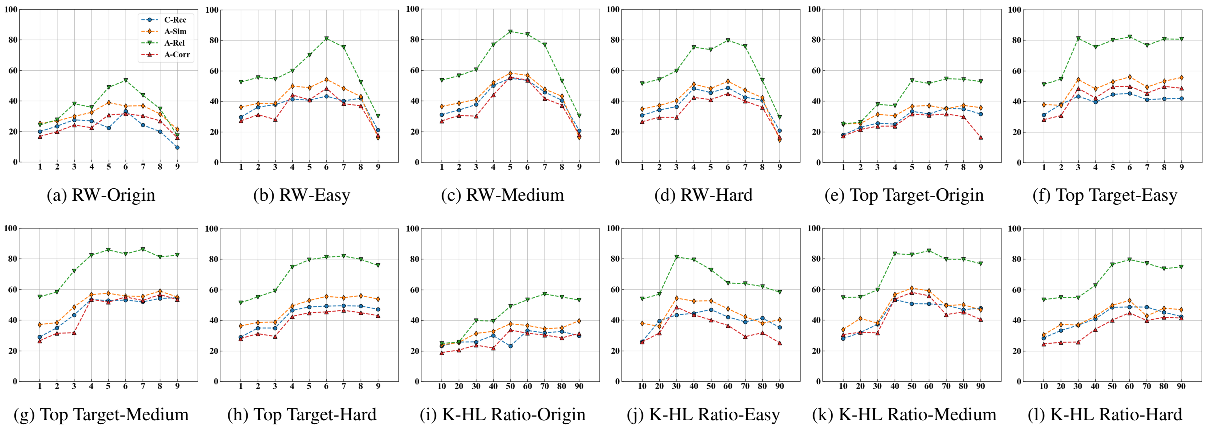
Topk Analysis of Target-Paragraph Weights. We investigate the trade-off between knowledge association strength and noise control by varying the neighborhood expansion depth in attribute graph retrieval (TopK parameter). As illustrated in Fig. 3 (e,f,g,h), increasing TopK yields a larger pool of candidate paragraphs; however, performance across all four task types consistently peaks at TopK=5. Beyond this point, additional paragraphs offer diminishing returns and may even degrade performance due to noise, while also incurring higher computational overhead. These results suggest that retrieving five target paragraphs is sufficient for the paragraph-level attribute graph.

Analysis of the Context Selection Ratio between Knowledge Graph and heterogeneous logical paragraph graph. In the analysis of context-selection ratios between the knowledge graph and the heterogeneous logical paragraph graph (Fig. 3 (i,j,k,l)), the x-axis denotes the proportion contributed by the attribute graph. For fact-extraction tasks, performance peaks when the attribute-graph share is 30–50%, since simpler responses benefit from more precise entity matching. In context-understanding tasks, the optimal share stabilizes at 50–70%, requiring a balance between KG context and attribute-graph–driven joint reasoning. Logical-deduction tasks exhibit a clear shift: maximal gains occur when the attribute-graph proportion rises to 60–75%, owing to its ability to capture cross-paragraph causal chains. Experimental results show that dynamically adjusting the ratio as a function of task type yields an overall performance improvement of 8%, with the most pronounced gain on Hard-Type reasoning tasks (+14%), thereby validating the adaptability advantage of the hybrid retrieval strategy.

5 Related Work

5.1 Graph-based RAG Techniques

More recently, knowledge-graphcentric RAG has gained traction. GraphRAG (Edge et al., 2024) adopts KGs as the retrieval backbone, while LightRAG (Guo et al., 2024b) emphasizes efficient graph-assisted inference. These trends highlight a



(a) RW-Origin (b) RW-Easy (c) RW-Medium (d) RW-Hard (e) Top Target-Origin (f) Top Target-Easy

(g) Top Target-Medium (h) Top Target-Hard (i) K-HL Ratio-Origin (j) K-HL Ratio-Easy (k) K-HL Ratio-Medium (l) K-HL Ratio-Hard

Figure 3: Parameter tuning experiments for four task types: **RW** denotes paragraph relation weight, **Top Target** refers to the number of selected HLPG target paragraphs, and **K-HL Ratio** indicates the ratio between knowledge graph and heterogeneous logical paragraph graph contexts.

shift toward structured, high-fidelity, and efficient retrieval-augmented generation. Memory-based approaches focus on long-term knowledge retention and querymemory decoupling, such as MemoRAG, MemQ, and HippoRAG (Qian et al., 2025; Xu et al., 2025; Gutierrez et al., 2024).

Graph-based methods model structural dependencies among retrieved contexts, including CG-RAG (Hu et al., 2025) and HyperGraphRAG (Luo et al., 2025), which capture coarse-grained and high-order semantic relations, respectively. Other works explore GNN-based retrieval, multi-agent reasoning, causal and event-centric modeling, and multi-hop linking (Mavromatis and Karypis, 2025; Gao et al., 2025; Yang et al., 2025).

5.2 Text-Attributed Graphs

Recent studies explore LLM reasoning over text-attributed graphs, though its integration with graph-based RAG remains underexplored (Jin et al., 2024a). Representative works include GAugLLM (Fang et al., 2024), which enhances textual node semantics, and Graph-CoT (Jin et al., 2024b), which enables joint reasoning over graph structure and text. Extensions to multi-modality and task-specific graph learning have also been investigated (Fang et al., 2025; Zhang et al., 2025; Fu et al., 2025; Zhang et al., 2024).

6 Time Complexity and Efficiency

As shown in Tab 4, KHL-RAG maintains comparable time complexity to LightRAG in graph construction stage, yet achieves significantly lower latency than GraphRAG during answer generation. Detailed analysis are as follow: From a time-complexity perspective, existing methods differ markedly in their cost sources and scalability. LightRAG incurs $O(m \cdot k)$ for local extraction, followed by a centralized post-processing cost of $O(e{\cdot}k{+}r \log r)$, which becomes a bottleneck as the scale grows. HyperGraphRAG supports $O(\log N)$ entity retrieval and $O(\log M)$ relation retrieval, but graph traversal introduces density-sensitive overheads ranging from $O(d^2)$ to $O(r \cdot d)$. CG-RAG is dominated by chunk encoding $O(|\mathcal{C}| \cdot T_{\text{enc}})$ and GNN inference $O(K \cdot |\tilde{\mathcal{E}}| \cdot D^2)$, which limits scalability on large corpora. By contrast, KHL-RAG adopts batched paragraph-relation construction with $O(N^2/K)$ complexity and linear node extraction $O(N)$, yielding more stable and efficient scaling in large-scale and dynamic document settings. Symbol definitions are provided in the Appendix 9.

7 Conclusion

This work presents KHL-RAG, a dual-graph retrieval framework that addresses retrieval instability and context omission in existing RAG systems by explicitly modeling paragraph-level discourse logic. By integrating a heterogeneous logical paragraph graph with an entity-based knowledge graph and employing query-aware retrieval and aggregation, KHL-RAG achieves superior performance, especially in complex reasoning scenarios. These results underscore the importance of discourse-aware structural modeling for robust and effective retrieval-augmented generation.

Limitations

HLPG construction relies on offline RST parsing, which incurs non-trivial upfront cost and limits suitability for real-time or on-the-fly knowledge integration. Consequently, KHL-RAG is better suited to scenarios where the knowledge base is pre-built and reused. Although explicit logical paths in HLPG reduce irrelevant context during inference, construction overhead remains a practical constraint; future work may explore lightweight or approximate discourse parsers to mitigate this cost.

Moreover, the current design mainly supports intra-document reasoning. Extending HLPG to cross-document settings introduces scalability challenges due to expensive paragraph-level edge construction. Future research may investigate cost-aware strategiessuch as heuristic pruning or approximate graph constructionto enable efficient and scalable cross-document HLPGs.

Ethics Statement

This paper addresses retrieval instability and concept shift in Retrieval-Augmented Generation by proposing KHL-RAG, a dual-graph framework composed of a knowledge graph and a heterogeneous logical paragraph graph, which enhances reasoning reliability by mitigating hallucination and modeling fine-grained semantic relations. Experiments across agriculture, computer science, and law demonstrate its effectiveness. The proposed architecture provides a general and extensible paradigm for reliable knowledge-enhanced reasoning across diverse domains. All datasets are constructed from publicly available standard benchmarks and contain no sensitive or personally identifiable information, raising no ethical concerns.

References

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

Shahul ES, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - System Demonstrations, St. Julians, Malta, March 17-22, 2024*, pages 150–158. Association for Computational Linguistics.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. 2024. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 747–758. ACM.

Yi Fang, Bowen Jin, Jiacheng Shen, Sirui Ding, Qiaoyu Tan, and Jiawei Han. 2025. GRAPHGPT-O: synergistic multimodal comprehension and generation on graphs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 19467–19476. Computer Vision Foundation / IEEE.

Yiwei Fu, Yuxing Zhang, Chunchun Chen, JianwenMa JianwenMa, Quan Yuan, Rong-Cheng Tu, Xinli Huang, Wei Ye, Xiao Luo, and Minghua Deng. 2025. MARK: multi-agent collaboration with ranking guidance for text-attributed graph clustering. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 6057–6072. Association for Computational Linguistics.

Junqi Gao, Xiang Zou, Ying Ai, Dong Li, Yichen Niu, Biqing Qi, and Jianxing Liu. 2025. Graph counselor: Adaptive graph exploration via multi-agent synergy to enhance LLM reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 24650–24668. Association for Computational Linguistics.

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

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024a. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 8048–8057. ijcai.org.

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

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yuntong Hu, Zhihan Lei, Zhongjie Dai, Allen Zhang, Abhinav Angirekula, Zheng Zhang, and Liang Zhao.

2025. CG-RAG: research question answering by citation graph retrieval-augmented llms. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 678–687. ACM.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9237–9251. Association for Computational Linguistics.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2024a. Large language models on graphs: A comprehensive survey. *IEEE Trans. Knowl. Data Eng.*, 36(12):8622–8642.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. 2024b. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 163–184. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Ze-min Kuang, Meina Song, Yifan Zhu, and Luu Anh Tuan. 2025. Hypergraphrag: Retrieval-augmented generation with hypergraph-structured knowledge representation. *CoRR*, abs/2503.21322.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

William C. Mann and Maite Taboada. 2006. Rhetorical structure theory: Looking back and moving ahead. *Discourse Studies*, 8(3):423–459.

Costas Mavromatis and George Karypis. 2025. GNN-RAG: graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 16682–16699. Association for Computational Linguistics.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, and 1 others. 2025. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025-2 May 2025*, pages 2366–2377. ACM.

Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh V. Chawla, and Chao Huang. 2024. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 6616–6626. ACM.

Kartik Sharma, Peeyush Kumar, and Yunqing Li. 2024. OG-RAG: ontology-grounded retrieval-augmented generation for large language models. *CoRR*, abs/2412.15235.

Mufan Xu, Gewen Liang, Kehai Chen, Wei Wang, Xun Zhou, Muyun Yang, Tiejun Zhao, and Min Zhang. 2025. Memory-augmented query reconstruction for llm-based knowledge graph reasoning. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 24068–24084. Association for Computational Linguistics.

Zairun Yang, Yilin Wang, Zhengyan Shi, Yuan Yao, Lei Liang, Keyan Ding, Emine Yilmaz, Huajun Chen, and Qiang Zhang. 2025. Eventrag: Enhancing LLM generation with event knowledge graphs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 16967–16979. Association for Computational Linguistics.

Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W. Lauw. 2024. Text-attributed graph representation learning: Methods, applications, and challenges. In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 1298–1301. ACM.

Taiyan Zhang, Renchi Yang, Yurui Lai, Mingyu Yan, Xiaochun Ye, and Dongrui Fan. 2025. Leveraging large language models for effective label-free node classification in text-attributed graphs. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 698–708. ACM.

A Appendix

A.1 Evaluation Metrics Details

In this chapters evaluation methodology, we draw upon the assessment frameworks of RAGAS(ES et al., 2024) and OG-RAG(Sharma et al., 2024), employing the following metrics: Context Recall (C-Rec), Answer Similarity (A-Sim), Answer Correctness (A-Corr), and Answer Relevance (A-Rel).

Context Recall (C-Rec) measures the degree of lexical overlap between the predicted answer and the reference answer. Formally, it is defined as:

$$C - Rec = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathcal{G}_i \cap \mathcal{P}_i|}{|\mathcal{G}_i|}, \tag{12}$$

where, N denotes the total number of test samples. $\mathcal{G}_i$ and $\mathcal{P}_i$ are the sets of tokens in the reference answer and the predicted answer for the i-th sample, respectively. $|\cdot|$ is the number in the set. If $|\mathcal{G}_i \cap \mathcal{P}_i| = 0$, then the contribution of that predicted sample toward the reference is zero.

Answer Similarity (A-Sim) measures the semantic consistency between answers by computing the cosine similarity of their semantic vector representations. Formally, it is expressed as:

$$A - Sim = \frac{1}{N}\sum_{i=1}^{N}\frac{e_g^{(i)} \cdot e_p^{(i)}}{\|e_g^{(i)}\| \cdot \|e_p^{(i)}\|} \tag{13}$$

where $e_g^{(i)} \in \mathbb{R}^d$ denotes the embedding vector of the reference answer text, $e_g^{(i)} \in \mathbb{R}^d$ denotes the embedding vector of the predicted answer text, $\cdot$ represents the dot product between two vectors, and $\|\cdot\|$ denotes the L2 (Euclidean) norm of a vector.

Answer Correctness (A-Corr) is a hybrid metric that combines semantic similarity with lexical matching. Formally, it is expressed as:

$$A - Corr = \frac{1}{N}\sum_{i=1}^{N}[\alpha \cdot A - Sim_i + (1-\alpha) \cdot F1_i], \tag{14}$$

where

$$F1_i = \frac{2 \cdot Recall_i \cdot Precision}{Recall_i + Precision_i}, \tag{15}$$

$$Recall_i = \frac{|\mathcal{G}_i \cap \mathcal{P}_i|}{|\mathcal{G}_i|}, Precision_i = \frac{|\mathcal{G}_i \cap \mathcal{P}_i|}{|\mathcal{P}_i|}, \tag{16}$$

where the corresponding term will be defined to be 0 when the denominator in the fraction equals zero. $\alpha \in [0, 1]$ is a hyperparameter, which defaults to 0.5.

Answer Relevance (A-Rel) is a metric for evaluating the relevance between the predicted answer and the original question. It is formally defined as:

$$A - Rel = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathcal{Q}_i \cap \mathcal{P}_i|}{|\mathcal{Q}_i|}, \tag{17}$$

where $\mathcal{Q}_i$ denotes the set of tokens in the original question for the i-th sample. when $|\mathcal{Q}_i \cap \mathcal{P}_i| = 0$, the relevance score for that predicted sample with respect to its original question is defined to be 0.

A.2 Experiment Implement Details

In our experiments, we employed OpenAIs GPT-4o-mini model for both information extraction and generation tasks, and utilized the text-embedding-3-small model to produce vector representations. For document segmentation, we adopted a fixed length chunking strategy with each chunk containing 1,200 tokens.

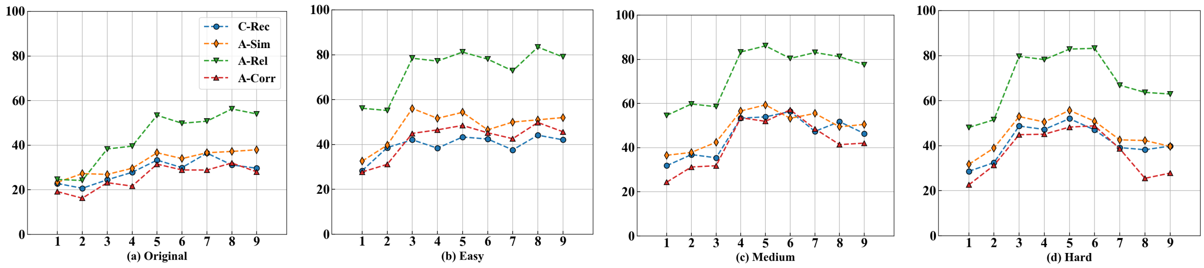
In the context retrieval stage, we set $k = 60$ for knowledge graph lookup, thereby selecting the 60 entities most relevant to the query to link with related passages. In the paragraph text attribute graph retrieval, we filter neighbor relations by weight $\tau = 5$, any first order neighbor of the target paragraph whose confidence score exceeds 5 is incorporated as additional paragraph information.

In the context orchestration phase, we distinguish three task types knowledgebased question answering, reading comprehension, and logical reasoningand tailor our evidence selection accordingly$(l, k) = (7, 3), (5, 5), (3, 7)$. For knowledgebased QA where answers can be retrieved directly, we prioritize passages linked via the knowledge graph; for reading comprehension tasks, we balance both graph linked and textbased passages; and for logical reasoning tasks that require complex inference, we rely more heavily on passages connected through the paragraphtext attribute graph.

All experiments were conducted on a server equipped with an 80-core CPU, 512 GB of RAM, and a single NVIDIA A40 GPU.

A.3 Parameter Analysis Experiments

TopK Analysis of Knowledge-Graph Weights. As shown in Fig. 4, in the TopK analysis for knowledge-graph entity retrieval, there is an interaction effect between the number of recalled



(a) (b) (c) (d)

Figure 4: Experimental Results for TopK Knowledge-Graph Weights.

Table 4: Time consumption analysis (Unit: seconds).

Model	Knowledge-Base Construction			QuestionAnswer Generation		
	10k	**100k**	**500k**	**10k**	**100k**	**500k**
NaiveGeneration	0	0	0	5.30	13.56	123.96
StandardRAG	0	0	0	6.58	19.16	134.88
GraphRAG	378.13	1325	8635	17.15	75.03	185.44
LightRAG	336.93	1863	11571	10.13	43.16	118.26
KHL-RAG (ours)	**423.18**	**2356**	**13215**	**12.24**	**55.69**	**137.41**

Table 5: Analysis of Large Language Model API Access Costs (Unit: CNY).

Model	Knowledge-Base Construction			QuestionAnswer Generation		
	10k	**100k**	**500k**	**10k**	**100k**	**500k**
NaiveGeneration	0	0	0	0.01	0.12	0.72
StandardRAG	0	0	0	0.02	0.19	0.93
GraphRAG	1.68	23.15	134.56	0.03	0.56	1.65
LightRAG	0.21	16.78	89.34	0.02	0.32	1.44
KHL-RAG(ours)	**0.37**	**19.88**	**125.29**	**0.03**	**0.45**	**1.57**

paragraphs and task difficulty. For most tasks, setting Top K = 4 is sufficient to distill the relevant and informative passages. As question difficulty increases, retrieving more paragraphs via the KG introduces greater noise, since entity-level associations cover a broader range and thus include more irrelevant content. This degrades the LLMs reasoning performanceparticularly on Hard tasks, where accuracy drops by nearly 8%indicating that overreliance on breadth-oriented retrieval can severely dilute the signal strength of core entities.

A.3.1 Performance of the KHL-RAG Model on the Original-Easy-Medium-Type Dataset.

From the results presented in Tables 6, 7, and 8, it is evident that as question difficulty increases, the performance advantage of KHL-RAGs dual-driven retrieval-augmented generation progressively widens relative to both the traditional RAG and knowledge-graph-based RAG frameworks. For origin type dataset, table 6 shows that KHL-RAG consistently outperforms both GraphRAG and LightRAG, which rely solely on knowledge graph retrieval. Compared to the KG-RAG pipeline, it achieves up to a 3.5% performance improvement. For fact-extraction questions (Table 7), KHL-RAG outperforms the strongest baseline, LightRAG, by 2.69%. Because the answers to these questions are closely tied to the keywords in the queries, knowledgegraph retrieval alone can swiftly locate the relevant context passages, supplying the large language model with highly precise associative information.

For the more challenging context-understanding tasks (see Table 8), KHL-RAG outperforms the strongest baseline, LightRAG, by 5.50%. While the knowledge graph can directly pinpoint related

Table 6: Performance of different models on the Origin-Type dataset.

Model	Agriculture				CS			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
Standard	12.23 ±1.02	16.13 ±0.81	24.91 ±1.49	10.87 ±0.62	15.72 ±1.11	19.25 ±0.99	25.55 ±1.28	16.33 ±0.71
GraphRAG	23.10 ±1.33	26.15 ±0.68	37.64 ±1.05	29.80 ±0.86	26.75 ±1.45	32.08 ±0.74	41.07 ±1.18	37.82 ±0.91
Cg-RAG	24.86 ±1.21	29.34 ±0.74	43.12 ±1.02	30.21 ±0.81	29.08 ±1.26	34.97 ±0.83	46.85 ±1.10	35.94 ±0.88
HyperGraph	26.71 ±0.96	32.58 ±1.05	48.37 ±0.91	31.04 ±0.92	31.56 ±0.88	36.89 ±1.12	51.62 ±0.97	36.28 ±0.84
LightRAG	26.05 ±0.84	34.02 ±1.29	51.92 ±0.96	31.11 ±1.13	31.12 ±0.67	37.45 ±1.37	53.73 ±1.01	36.57 ±0.80
KHL-RAG	**33.31** ±1.48	**36.61** ±0.79	**53.41** ±1.23	**31.52** ±0.90	**37.25** ±1.09	**38.52** ±0.82	**55.72** ±1.31	**39.17** ±1.04

Model	Legal				Cooking			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
Standard	13.70 ±1.26	17.16 ±0.76	27.01 ±1.41	11.91 ±0.89	19.62 ±1.12	22.39 ±0.94	31.29 ±1.20	28.96 ±0.66
GraphRAG	24.53 ±0.98	27.29 ±1.39	39.16 ±0.83	31.05 ±1.03	30.23 ±1.47	35.26 ±0.64	44.57 ±1.14	37.45 ±0.92
Cg-RAG	26.14 ±1.05	30.92 ±0.81	45.73 ±1.12	31.46 ±0.88	33.28 ±1.20	38.94 ±0.79	51.02 ±1.06	38.11 ±0.97
HyperGraph	28.02 ±0.89	33.84 ±1.17	49.86 ±0.94	32.01 ±0.91	36.12 ±1.03	41.56 ±0.86	55.18 ±1.21	39.02 ±1.05
LightRAG	27.51 ±1.22	35.07 ±0.69	53.29 ±1.46	32.18 ±0.78	35.64 ±1.08	43.23 ±0.85	57.13 ±1.33	39.16 ±1.00
KHL-RAG	**34.80** ±0.73	**37.85** ±1.30	**54.90** ±1.19	**32.69** ±0.97	**43.14** ±1.25	**45.09** ±1.02	**60.02** ±0.63	**43.13** ±1.40

Table 7: Performance of different models on the Easy-Type dataset.

Model	Agriculture				CS			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
Standard	18.82 ±1.25	24.82 ±0.99	38.33 ±0.61	16.72 ±1.49	24.19 ±0.81	29.61 ±1.12	39.39 ±0.95	25.12 ±1.30
GraphRAG	38.54 ±0.78	40.23 ±1.33	57.90 ±0.89	45.92 ±1.02	41.16 ±1.47	49.36 ±0.64	63.19 ±1.15	58.18 ±1.08
LightRAG	40.07 ±0.91	52.26 ±1.28	79.88 ±0.75	47.88 ±1.39	47.82 ±0.84	57.61 ±1.00	82.67 ±1.22	56.26 ±0.70
Cg-RAG	39.86 ±0.91	46.17 ±1.02	69.25 ±0.84	46.81 ±0.96	44.38 ±1.10	53.24 ±0.88	73.41 ±1.03	55.02 ±0.79
HyperGraph	41.12 ±0.87	49.32 ±1.11	75.63 ±0.92	47.36 ±1.08	46.71 ±0.93	55.78 ±1.06	79.18 ±0.89	55.74 ±0.85
KHL-RAG	**43.33** ±1.44	**54.33** ±0.68	**81.17** ±1.11	**48.49** ±0.97	**49.31** ±1.29	**59.31** ±0.88	**85.81** ±1.04	**57.18** ±1.40

Model	Legal				Cooking			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
Standard	21.07 ±1.09	26.47 ±0.85	41.48 ±1.27	18.32 ±0.96	29.23 ±1.31	34.37 ±0.77	45.25 ±1.10	29.71 ±1.01
GraphRAG	37.74 ±0.69	41.98 ±1.43	60.25 ±1.03	47.77 ±0.80	46.52 ±1.19	53.26 ±0.90	68.17 ±1.34	62.31 ±0.87
LightRAG	42.32 ±1.23	53.96 ±0.94	81.98 ±1.38	49.43 ±0.63	54.62 ±1.20	63.25 ±1.07	86.19 ±0.74	58.23 ±1.48
Cg-RAG	40.15 ±0.88	47.62 ±1.04	71.84 ±0.97	48.62 ±0.85	50.36 ±1.10	57.41 ±0.83	76.94 ±1.12	60.21 ±0.92
HyperGraph	41.83 ±0.94	50.71 ±1.15	77.96 ±0.89	49.12 ±1.01	52.14 ±0.97	60.32 ±1.06	82.27 ±0.91	58.74 ±0.88
KHL-RAG	**43.53** ±0.83	**54.23** ±1.13	**82.47** ±1.26	**50.29** ±0.79	**52.71** ±1.42	**62.88** ±0.98	**89.13** ±1.16	**60.53** ±0.67

passages, it may lack certain indirect relational knowledge; nonetheless, the retrieved context remains highly relevant. By incorporating the paragraphtext attribute graph, KHL-RAG additionally captures latent logical connections between passages, yielding answers that are not only semantically consistent but also more comprehensive.

Analyzed in terms of coverage (C-Rec), semantic consistency (A-Sim), relevance evaluation (A-Rel), and the composite metric (A-Corr), the paragraph-text attribute graph explicitly models inter-paragraph logical relations (causal, exemplification, sequential) via Rhetorical Structure Theory (Mann and Taboada, 2006), thereby constructing a multi-hop association network.

A.3.2 Performance Analysis and Energy Consumption (RQ4)

To evaluate the performance and cost of the KHL-RAG framework (RQ4), this section presents a comprehensive comparison of the two stages: knowledge-base construction and question-answer generation. The evaluation focuses on the framework's time consumption and the computational cost of invoking the large language model, considering input context lengths of 10k, 100k, and 500k words.

As shown in Table 4, for context inputs on the

Table 8: Performance of different models on the Medium-Type dataset.

Model	Agriculture				CS			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
Standard	21.12 ±1.03	26.02 ±0.81	40.33 ±1.39	17.72 ±0.98	26.39 ±1.19	30.71 ±0.72	41.49 ±1.48	26.02 ±0.85
GraphRAG	37.94 ±0.91	41.33 ±1.27	59.10 ±0.63	46.72 ±1.08	43.46 ±1.35	50.36 ±1.01	65.19 ±0.79	52.08 ±1.23
LightRAG	42.17 ±0.87	53.26 ±1.45	82.18 ±1.05	48.78 ±0.74	50.02 ±1.11	59.41 ±0.96	85.07 ±1.30	57.36 ±0.90
Cg-RAG	41.26 ±0.97	46.74 ±1.08	71.82 ±0.91	48.06 ±0.85	47.91 ±1.14	54.29 ±0.88	74.83 ±1.02	55.43 ±0.96
HyperGraph	44.38 ±0.88	50.12 ±1.19	78.64 ±0.86	49.21 ±0.92	50.37 ±0.99	57.18 ±1.04	81.76 ±0.90	57.69 ±0.89
KHL-RAG	**53.38** ±1.29	**56.61** ±0.65	**83.32** ±1.41	**53.52** ±0.83	**53.61** ±1.38	**60.21** ±0.92	**88.01** ±1.16	**61.18** ±0.67

Model	Legal				Cooking			
	C-Rec	A-Sim	A-Rel	A-Corr	C-Rec	A-Sim	A-Rel	A-Corr
Standard	23.37 ±0.95	28.67 ±1.24	44.58 ±0.61	20.32 ±1.40	34.78 ±0.78	37.81 ±1.13	44.13 ±0.99	37.32 ±1.04
GraphRAG	41.14 ±1.31	44.08 ±0.89	64.45 ±1.44	49.57 ±0.68	48.25 ±1.20	50.22 ±1.09	78.32 ±0.75	54.18 ±1.17
LightRAG	44.42 ±0.93	54.96 ±1.37	84.28 ±0.80	50.33 ±1.25	55.21 ±1.02	62.13 ±0.97	87.27 ±1.49	58.13 ±0.64
Cg-RAG	43.36 ±0.88	49.27 ±1.06	74.61 ±0.93	50.12 ±0.91	52.04 ±1.08	55.91 ±0.84	82.74 ±1.03	56.82 ±0.95
HyperGraph	45.21 ±0.92	52.48 ±1.11	80.39 ±0.86	50.71 ±0.88	54.96 ±1.02	59.18 ±0.97	85.41 ±1.12	57.93 ±0.89
KHL-RAG	**48.53** ±0.77	**57.43** ±1.12	**86.57** ±1.33	**51.29** ±0.84	**61.22** ±1.42	**62.32** ±0.66	**90.19** ±1.10	**62.72** ±1.07

order of 10 K tokens, KHL-RAGs knowledge-base construction takes 423.18 sslightly longer than GraphRAGs 390.13 s and LightRAGs 316.93 s. The increased context size also causes KHL-RAGs QA generation time to be about 20.5 % higher than LightRAGs. For small data volumes, the core overhead of edge-relation construction remains within an acceptable range. As input size growsand owing to the finer granularity of entity-relation tripleswhen context length increases to 500 K words, the cost of knowledge-graph construction rises substantially. In this regime, the construction-time gap between KHL-RAG and LightRAG narrows from 20.9 % down to 12.4 %.

Both the knowledge-base construction and the QA generation stages require API calls to the large language model. As shown in Table 5, during the knowledge-base construction phase, KHL-RAGs dual-drive architecture must build both a knowledge graph and a paragraph-text attribute graph, resulting in a higher number of API invocations. For a 500 K-token knowledge base, this translates to an additional cost of approximately RMB 35.95 compared to constructing only the knowledge graph. In the QA generation stage, because only a single API call is made, cost differences among frameworks are driven primarily by the length of the provided context, and thus inter-framework disparities are minimal. As context size grows, the knowledge-base construction phase remains the most costly. Since GraphRAGs graph construction is already more complex and KHL-RAG adds the paragraph-text attribute graph on top, KHL-RAGs API consumption is roughly 1.7 times that of LightRAG.

A.4 Prompts Used in KHL-RAG

We present the prompts designed for constructing the Heterogeneous Logical Paragraph Graph, covering paragraph extraction, inter-paragraph relation extraction, and dataset QA-pair construction. Specifically, the paragraph extraction prompts identify semantically coherent and task-relevant text units; the relation extraction prompts infer fine-grained logical dependencies (e.g., contrast, and causality) among paragraphs; and the QA-construction prompts generate difficulty-controlled questionanswer pairs grounded in the HLPG. Together, these prompts operationalize structured knowledge induction and ensure consistent, interpretable retrieval and reasoning within KHL-RAG.

Notation Explanation

The following table provides a detailed explanation of the key symbols and variables used in the time-complexity analysis of the different Retrieval-Augmented Generation methods.

Table 9: Key Notations Used in Time Complexity Analysis

Symbol	Description	Associated Models
N	Number of paragraphs in the context.	KHL-RAG
K	Number of paragraphs processed per batch within the LLM token limit (KHL-RAG); also denotes the number of GNN layers (CG-RAG).	KHL-RAG, CG-RAG
m	Number of text chunks after document segmentation.	LightRAG
e	Total number of extracted entities.	LightRAG
r	Number of relations/edges in the constructed graph.	LightRAG, HyperGraphRAG
k	Average number of attributes per entity or chunk.	LightRAG
d	Average node degree of the graph.	HyperGraphRAG
M	Number of relation vectors stored in the vector database.	HyperGraphRAG
$\|\mathcal{C}\|$	Total number of document chunks (nodes) in the context graph ($\|\tilde{\mathcal{V}}\|$).	CG-RAG
T_{enc}	Encoding time for a single chunk.	CG-RAG
$\|\tilde{\mathcal{E}}\|$	Total number of edges in the context graph $\tilde{G}$.	CG-RAG
D	Embedding dimension used in the GNN model.	CG-RAG

```
-Goal-
Given a text document potentially related to this activity and a list of paragraph types, identify
the main idea, key terms, and type of each paragraph.
Use {language} as the output language.

-Steps-
1. Analyze and summarize the main idea of the paragraph in a single sentence.
2. Identify all prominent topics or high-level keywords based on the main idea and the original
text.
3. Determine the paragraph type and provide the rationale for the classification, considering the
main idea, original text, and paragraph type descriptions.
- paragraph_type: One of the following types: [{paragraph_types}]
- paragraph_type_description: Refer to the relevant paragraph type descriptions:
[{paragraph_type_description}]
4. Return the output in {language}, including the main idea, keywords, paragraph type, and
classification rationale.

########################
Example:
########################
{examples}

########################
Real Data:
########################
Text: {input_text}

Output:
```

Figure 5: Prompt for paragraph extraction and classification.

```
-Goal-
Given two text document paragraphs potentially related to this activity and a list of paragraph relation types, analyze the original text, main ideas, keywords, paragraph types, and reasons to identify the type of relationship between the two paragraphs, the confidence weight of the relationship (0-10), and the reasoning behind the analysis.
Use {language} as the output language. Please note that if the relationship type is analyzed as None, no other types should be present; avoid logical inconsistencies.

-Steps-
1. Analyze the original text of the two document paragraphs and extract relevant information. Based on the paragraph relation types and their descriptions, determine the relationship type between the two paragraphs and the reasoning behind the decision.
- paragraph_relation_type: Refer to the following types: [{paragraph_relation_type}]
- paragraph_relation_type_description: Refer to the related paragraph type description information: [{paragraph_relation_type_description}]
- relation_weights_scope: 0-10
2. Return the output in {language} format, including the relationship types that exist between the two paragraphs (there may be multiple types) and the reasons for the paragraph relationship judgment.
3. Please note that if the relationship type is analyzed as None, no other types should be present; avoid logical inconsistencies.

########################
Example:
########################
{examples}

########################
Real Data:
########################
TextA: {input_textA}
TextA_main_idea: {input_textA_main_idea}
TextA_keyword: {input_textA_keyword}
TextA_paragraph_type: {input_textA_paragraph_type}
TextA_type_reason: {input_textA_type_reason}
TextB: {input_textB}
TextB_main_idea: {input_textB_main_idea}
TextB_keyword: {input_textB_keyword}
TextB_paragraph_type: {input_textB_paragraph_type}
TextB_type_reason: {input_textB_type_reason}

Output:
```

Figure 6: Prompt for paragraph relationship extraction and classification.

-Goal-
Given the following description of a dataset: total_description
Focus on generating question #question_number (out of 5) for User: user, Task: task.
Use language as output language. The question should be of difficult level, requiring multi-step reasoning (3 logical operations) and based on complex logical relationships among citations.

-Steps-
1. Select 3 random paragraphs from total_description ensuring complex logical relationships, e.g., Passage 1 → Passage 2, Passage 3 → Passage 2, Passage 1 → Passage 3.
2. Determine the logic types between paragraphs using: Evidence, Elaboration, Background, Purpose, Result, Cause, Contrast, Concession, Sequence, Summary, Evaluation, Problem-Solution, Conclusion, Theme, Example.
3. Derive a challenging question related to task and property evaluation based on the 3 citations.
4. Provide a detailed answer paragraph that demonstrates reasoning, clarity, and logical flow; answer must not be directly found in the original passages.
5. Output must strictly follow this sequence and format:
- Citations (3 random paragraphs):
- Logic of 3 randomised paragraphs:
- Question: ...
- Answer: ...

-Dataset Users and Tasks-
```
agriculture_USERS_TASKS = {
"User 1: Small Farm Owner": [
"Optimizing Crop and Livestock Production",
"Managing Farm Resources Efficiently",
"Adopting Sustainable Farming Practices",
"Controlling Costs and Improving Profitability",
"Responding to Climate and Market Risks"
],
"User 2: Agricultural Cooperative Manager": [
"Coordinating Farmers and Shared Resources",
"Improving Supply Chain Efficiency",
"Negotiating Market Access",
"Providing Technical Support to Members",
"Strengthening Collective Bargaining Power"
],
"User 3: Agricultural Researcher": [
"Studying Crop and Soil Systems",
"Developing Sustainable Technologies",
"Analyzing Agricultural Data",
"Evaluating Environmental Impacts",
"Disseminating Research Outcomes"
],
"User 4: Agricultural Policy Maker": [
"Designing Agricultural Support Policies",
"Promoting Food Security",
"Encouraging Sustainable Agriculture",
"Monitoring Rural Development",
"Assessing Policy Effectiveness"
],
"User 5: Agribusiness Operator": [
"Managing Agricultural Processing Operations",
"Ensuring Product Quality and Safety",
"Optimizing Logistics and Distribution",
"Adapting to Market Demand",
"Integrating Digital Agriculture Solutions"
]
}
```

Difficult Level Question Requirements:
- Select 3 random paragraphs demonstrating complex logical relationships (3 logical operations).
- Generate a question relevant to task and property evaluation.
- Provide a detailed answer paragraph showing reasoning, clarity, and logical flow; answer cannot be directly found in the original passages.
- Output format must strictly follow:

Paragraph 1: ...
Paragraph 2: ...
Paragraph 3: ...
Logic of 3 randomised paragraphs: ...
Question: ...
Answer: ...

Output: Generate ONE difficult level question following the above requirements.

Figure 7: Prompt for generating difficult-level questions from complex multi-paragraph logical relationships.