Query-Aware Graph Attention for Precise Subgraph Retrieval in Knowledge-Augmented Reasoning

Anonymous ACL submission

Abstract

Large language models (LLMs) increasingly rely on external knowledge to mitigate hallucinations, yet retrieving precise multi-hop evidence for knowledge-augmented reasoning remains difficult. Existing Knowledge Graph (KG)-based Retrieval-Augmented Generation (RAG) systems insufficiently model the interaction between query semantics and relation types, resulting in imprecise subgraph retrieval and unstable reasoning. We propose Query-aware Subgraph Retrieval Augmented Generation (QSRAG), a retrieval framework built upon a Query-Relational Graph Attention Network (QR-GAT) that integrates query semantics and relation embeddings directly into the attention mechanism, enabling fine-grained triple scoring and scalable subgraph construction. This query–relation conditioning improves relevance estimation and suppresses noisy edges, producing faithful reasoning subgraphs. Experiments on WebQSP and CWQ establish new state-of-the-art results in both Triple Recall and Answer Recall, and significantly enhance LLMs reasoning accuracy without finetuning. These findings underscore the effectiveness of modeling query–relation interactions for reliable knowledge-augmented reasoning.

1 Introduction

Large language models (LLMs) have driven rapid advances in natural language processing (Brown et al., 2020; Huang and Chang, 2022; Wei et al., 2022), yet their internal knowledge remains static and incomplete (Kasai et al., 2023; Ji et al., 2023). Retrieval-Augmented Generation (RAG) mitigates this limitation by supplementing LLMs with external information (Borgeaud et al., 2022; Gao et al., 2023). Knowledge graphs (KGs), with their structured and multi-hop relational information, provide an especially promising source of factual evidence for complex reasoning tasks such as Knowledge Graph Question Answering (KGQA) (Guo et al., 2024; Pan et al., 2024; Peng et al., 2024).

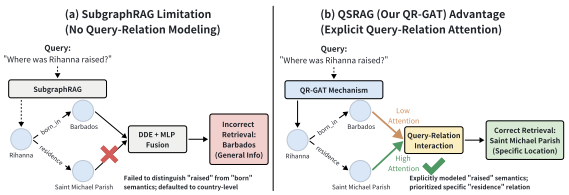

Figure 1: Comparison of Query-Aware Retrieval Mechanisms using a Concrete Example ("Where was Rihanna raised?").

However, effectively integrating KGs into RAG requires accurately retrieving a compact subgraph that reflects the semantics of the natural-language query. Despite recent progress, existing KG-based RAG approaches still face several fundamental obstacles.

Challenge 1: Identifying correct multi-hop reasoning evidence remains difficult. Real-world KGQA queries often rely on subtle relational distinctions. Although recent retrieval methods (Jiang et al., 2023a; Luo et al., 2024; Sun et al., 2024) can capture plausible paths, they remain susceptible to redundant neighbors and spurious relations that obscure the actual reasoning chain.

Challenge 2: Insufficient modeling of interactions between query semantics and relation types. Most graph retrieval and GNN-based approaches encode graph structure independently of the query (Mavromatis and Karypis, 2025; Yasunaga et al., 2021; Li et al., 2025), preventing attention from adapting to the reasoning intent. This limitation becomes especially evident in questions requiring fine-grained relational distinctions.

Figure 1 provides an illustrative example that can well explain this problem. For the question *"Where was Rihanna raised?"*, the ground truth is *Saint Michael Parish*. SubgraphRAG (Li et al., 2025) retrieves *Barbados* instead, because its DDE+MLP scoring lacks explicit query–relation interaction. Without understanding the semantic nuance be-

tween "born" and "raised," it overweights general geographic relations (e.g., nationality, birthplace) and ignores more specific residence-related edges. This example highlights the need for retrieval mechanisms that are sensitive to both query semantics and relation types across multi-hop structures.

Challenge 3: High latency from repeated LLM calls during retrieval or reasoning. Several KG-based RAG systems invoke the LLM multiple times during planning, path expansion, or iterative reasoning (Jin et al., 2024; Gao et al., 2024; Kim et al., 2023; Ma et al., 2024; Xiong et al., 2024; Sun et al., 2024). While effective, these pipelines incur substantial latency and limit scalability in practical settings. A more efficient paradigm would perform structured retrieval once and rely on a single LLM inference step for final reasoning.

Our Solution. To address these challenges, we propose **QSRAG**, a two-stage KG-based RAG framework built around a novel **Query-Relational Graph Attention Network (QR-GAT)**. QR-GAT injects global query semantics and explicit relation embeddings directly into the attention computation of graph message passing, enabling fine-grained, query-aware, and relation-guided triple scoring across large KGs. This allows the model to prioritize relations aligned with the query (e.g., "raised" $\rightarrow$ residence-like relations), thereby correcting the failure modes observed in systems like SubgraphRAG.

Using the resulting triple-level scores, QSRAG constructs a concise, high-quality subgraph that minimizes redundancy and reduces noise within the limited LLM context window. The final reasoning stage requires only a single LLM call and does not rely on any LLM fine-tuning, maintaining generality and efficiency.

Our contributions are summarized as follows:

- We propose QSRAG, a framework for structured evidence retrieval in KGQA, centered around a novel QR-GAT for precise triple scoring.
- Experiments on WebQSP and CWQ demonstrate that QR-GAT achieves state-of-the-art performance on standard retrieval metrics.
- Integrating QR-GAT subgraphs significantly improves LLM reasoning accuracy, yielding state-of-the-art or competitive results without LLM fine-tuning.
- Ablation studies confirm the importance of query–relation attention and show that confidence scores facilitate more effective LLM evidence utilization.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces preliminaries. Section 4 describes QR-GAT and the QSRAG framework. Section 5 presents experiments, followed by additional analysis in Section 6. Section 7 concludes the paper. Additional QA examples, ablation results and auxiliary experiments are provided in the appendix.

2 Related Work

KG-based RAG and Graph Retrieval. Retrieval-Augmented Generation has emerged as an effective paradigm for enhancing the factual reliability of large language models, and incorporating knowledge graphs as structured external sources has shown particular promise for multi-hop reasoning. A central challenge in KG-based RAG is to retrieve compact, query-relevant subgraphs from large KGs while preserving the multi-hop structures required for reasoning. Recent efforts tackle this problem through diverse mechanisms: planning-based path retrieval in RoG (Luo et al., 2024), trainable subgraph scoring models such as SubgraphRAG (Li et al., 2025), and heuristic or combinatorial search methods (Sun et al., 2024; He et al., 2024; Hu et al., 2024). Other lines of work convert KGs into serialized text for LLM consumption (Wu et al., 2023), introduce biologically inspired retrieval mechanisms (Gutierrez et al., 2024), or build unified retriever–reasoner architectures for KGQA (Jiang et al., 2023b). Separately, earlier semantic parsing approaches for complex KGQA, such as the hierarchical query graph generation method HGNet (Chen et al., 2023)—also emphasize structural decomposition (e.g., decoupling topology prediction from instance selection) to handle complex queries. However, these methods are primarily designed for generating executable SPARQL queries and do not provide triple-level relevance scoring conditioned jointly on query semantics and relation types. Consequently, even structurally informed retrieval in KG-based RAG or SP-based parsing in KGQA—may still deliver coarse or spurious relational evidence, especially for multi-hop queries. Our work directly addresses this gap by introducing a query and relation-aware attention

mechanism that assigns fine-grained, triple-level scores and enables globally precise subgraph retrieval.

Graph Neural Networks for KGs. Graph neural networks have been widely applied to knowledge graphs for representation learning, link prediction, and reasoning. GNNs propagate information across graph neighborhoods, and Graph Attention Networks (GATs) (Brody et al., 2022) further enhance expressiveness through attention-based aggregation. Several KGQA and KG-based RAG systems leverage GNNs to encode question-specific subgraphs. GRAFT-Net (Sun et al., 2018) integrates KG and textual evidence using heterogeneous attention; more recent work applies GNNs to dense retrieved subgraphs or shortest-path structures (Mavromatis and Karypis, 2025). Variants of GATs also explore temporal or structural dependencies in retrieved contexts (Gao et al., 2024).

However, existing GNN-based methods typically operate on pre-selected candidate subgraphs and are not tailored to fine-grained, query-conditioned triple scoring over large KGs. Standard GATs treat neighbors uniformly with respect to the query and cannot dynamically adjust attention based on relation semantics. Our work builds on these foundations but departs by introducing a graph attention mechanism that explicitly encodes query semantics and relation types into the attention computation, allowing direct triple-level relevance estimation for precise and efficient subgraph retrieval.

3 Preliminaries

In this section, we introduce key concepts relevant to our work and formally define the problem studied in this paper.

Knowledge Graphs. A knowledge graph G is a structured representation of factual knowledge in the form of a graph. It typically consists of a set of entities E, a set of relation types R, and a set of factual triples T. Each triple $(h, r, t) \in T$ denotes a fact, where $h \in E$ is the head entity, $r \in R$ is the relation, and $t \in E$ is the tail entity.

Knowledge Graph Question Answering. KGQA refers to the task of answering a natural language question q by identifying the correct answer a from a knowledge graph G. The answer a may be a single entity or a set of entities in the graph. Solving KGQA typically requires understanding the semantic intent of the question, mapping it to the structure of the KG, and performing reasoning or structured querying to locate the correct answer.

Retrieval-Augmented Generation. RAG is a powerful paradigm that combines information retrieval with the generative capabilities of LLMs. In a typical RAG framework, given a user query, a retrieval module first retrieves relevant knowledge pieces or evidence from an external knowledge source. These retrieved items are then concatenated with the original query and fed into a generation module—typically an LLM—to produce the final response or answer. RAG enables LLMs to incorporate external, real-time, or domain-specific knowledge, thereby reducing hallucinations and improving the factual accuracy and reliability of generation.

Problem Definition. We study complex multi-hop KGQA within the RAG framework, where the external knowledge source is a knowledge graph. Formally, given a natural language question q and a knowledge graph $G = (E, R, T)$, the goal is to train a two-stage model to output the correct answer a from G. The first stage retrieves a subgraph $S \subseteq G$ containing triples highly relevant to q that support reasoning. In the second stage, the retrieved subgraph S is provided as context to an LLM, which performs multi-hop reasoning via in-context learning (ICL)(Brown et al., 2020) to generate the final answer. These two stages work jointly to achieve accurate, interpretable KG-based RAG.

Our main focus is to design an efficient and accurate retrieval module, especially to leverage the query-relational graph attention mechanism to accurately extract multi-hop evidence from large-scale knowledge graphs.

4 Methodology

We propose **QSRAG**, a retrieval-augmented generation framework enhanced by knowledge graphs, which accurately retrieves structured evidence relevant to a given question and guides LLMs in reasoning. QSRAG consists of two main stages: structured evidence retrieval and evidence-based contextual reasoning, as illustrated in Figure 2.

Before building the graph representation, we use the `Qwen3-Embedding-0.6B` encoder(Zhang et al., 2025) to obtain semantic embeddings of entities, relations, and the input question. This yields rich

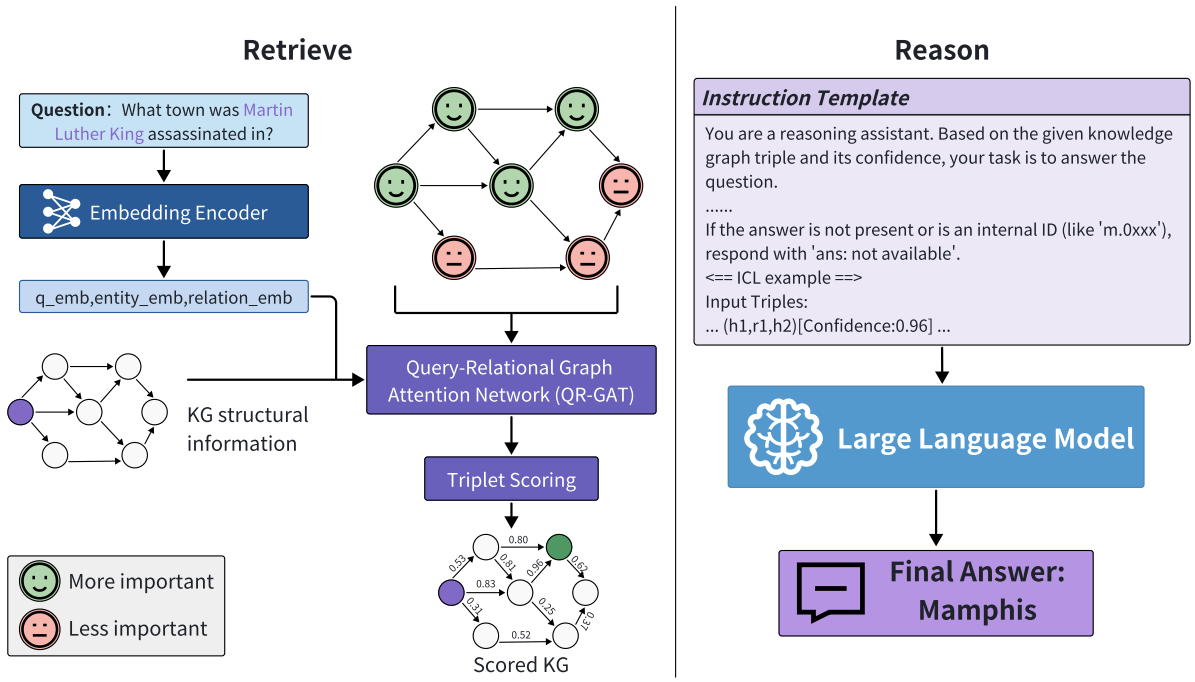

Figure 2: Overview of our QSRAG framework, consisting of (1) a QR-GAT for structured evidence retrieval from the KG, and (2) a contextual reasoning module using an LLM with in-context learning.

representations: $\mathbf{e}_i$ for entity v_i, $\mathbf{r}_{ij}$ for relation r_{ij}, and $\mathbf{q}$ for the question.

4.1 Structured Evidence Retrieval

The goal of this stage is to extract a subgraph $S \subseteq G$ from the KG that contains triples most relevant to the input question. Rather than using traditional text matching or neighborhood expansion, we model the fine-grained semantic interactions among the query, entities, and relations across multi-hop paths using a novel attention mechanism—**QR-GAT**.

QR-GAT is designed to guide attention dynamically toward entities and relations that are critical for answering a specific question. It incorporates both query semantics and relation embeddings into the attention computation process. Standard GATs typically treat neighbors uniformly or based only on structure, ignoring query-specific signals. QR-GAT overcomes this limitation by introducing query- and relation-aware attention, which enables more focused evidence retrieval that is tailored to the question at hand.

Each entity node v_i is initialized with:

$$\mathbf{h}_i^{(0)} = \text{Dropout}([\mathbf{e}_i \| \mathbf{q} \| \mathbf{p}_i])$$

where $\mathbf{e}_i \in \mathbb{R}^{d_e}$ is the entity embedding, $\mathbf{q} \in \mathbb{R}^{d_q}$ is the query embedding, and $\mathbf{p}_i$ is a one-hot vector encoding used to label whether the entity is the topic entity. This initialization injects both semantic information from the query and structural knowledge from the entity into the graph representation at the outset, facilitating a more query-aware attention mechanism.

At each layer l, we perform linear projections:

$$\mathbf{z}_i^{(l)} = W_s^{(l)} \cdot \mathbf{h}_i^{(l-1)}, \quad \mathbf{z}_j^{(l)} = W_t^{(l)} \cdot \mathbf{h}_j^{(l-1)}$$

where $W_s^{(l)}$ and $W_t^{(l)}$ are learnable weights for source and target roles respectively. These weights help refine the message passing between neighboring nodes, capturing the relationships more effectively.

The attention score $\alpha_{ij}^{(l)}$ is computed by combining structural and query-guided terms:

$$\alpha_{ij,\text{base}}^{(l)} = \mathbf{a}^{(l)\top} \cdot \text{LeakyReLU}(\mathbf{z}_i^{(l)} + \mathbf{z}_j^{(l)} + W_e^{(l)} \cdot \mathbf{r}_{ij})$$

$$\alpha_{ij,\text{plus}}^{(l)} = (W_q^{(l)} \cdot \mathbf{q})^\top \cdot (W_r^{(l)} \cdot \mathbf{r}_{ij})$$

$$\alpha_{ij}^{(l)} = \text{softmax}_j(\alpha_{ij,\text{base}}^{(l)} + \alpha_{ij,\text{plus}}^{(l)})$$

Here, $\mathbf{a}^{(l)}$ is a learnable attention vector that produces a scalar compatibility score. The transformation $W_e^{(l)}$ injects relation semantics into the structural attention term, while $W_q^{(l)}$ and $W_r^{(l)}$ project the query and relation embeddings into a shared

space to model their semantic alignment. This design enables query-conditioned attention over relations and mitigates attention dilution by emphasizing query-relevant relational signals.

Node representations are updated via multi-head attention:

$$\mathbf{h}_i^{(l)} = \text{LayerNorm}([\mathbf{h}_{i,1}^{(l)} \| \ldots \| \mathbf{h}_{i,H}^{(l)}])$$

where $\mathbf{h}_{i,k}^{(l)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l,k)} \cdot \mathbf{z}_j^{(l)}$ for attention head k. This multi-head attention allows the model to capture diverse patterns of interaction between entities and relations.

We use a bidirectional QR-GAT (BiQR-GAT) to encode both forward and reverse edges. Final entity representation is:

$$\mathbf{h}_i = [\mathbf{h}_i^{\rightarrow} \| \mathbf{h}_i^{\leftarrow}]$$

Key Engineering Insights: Beneath its conceptual simplicity, QR-GAT is founded on two pivotal engineering insights that enable its performance: 1. **Query-aware initialization strategy**: By injecting semantic information from the query into the initialization of node embeddings, we enhance the model's ability to capture query-specific attention patterns from the outset. This strategy ensures that the graph is "aware" of the query semantics early in the process, improving evidence retrieval. 2. **Optimal integration of query and relation embeddings**: The use of learnable projection matrices $W_q^{(l)}$ and $W_r^{(l)}$ allows for a fine-grained integration of query and relation information. This ensures that the attention mechanism is sensitive to both the structural relations in the graph and the specific semantics of the query, preventing the attention scores from being diluted by irrelevant relational information.

By combining these engineering insights, QR-GAT is able to dynamically adjust its attention based on both the query and the relations, leading to more accurate and efficient subgraph retrieval that supports complex multi-hop reasoning.

Triplet Scoring. Using final node representations $\mathbf{h}_h$ and $\mathbf{h}_t$ for head and tail entities, the score of a triple (h, r, t) is computed via a two-layer MLP:

$$s(h, r, t) = W_2 \cdot \text{ReLU}(W_1 \cdot [\mathbf{q} \| \mathbf{h}_h \| \mathbf{r} \| \mathbf{h}_t])$$

Training and Inference. The retriever is trained as a binary classifier, using a binary cross-entropy loss function to distinguish between positive triplets (triplets on the shortest path between the topic entity and the answer entity) and negative triplets. At inference time, the scores of all candidate triplets are calculated, and the top-k triplets are selected to form a structured evidence subgraph.

4.2 Evidence-based Contextual Reasoning

The retrieved structured evidence is serialized into textual form and provided to the LLM for final answer generation. Each triple is expressed as a concise factual statement such as `(head, relation, tail) (Confidence: 0.96)`, and the concatenation of all selected triples forms the evidence context appended to the question. The LLM then performs reasoning over this evidence-conditioned prompt.

Beyond using the top-k triples ranked by their relevance scores, we propose an adaptive evidence filtering mechanism inspired by nucleus sampling (top-p) in language generation. Instead of fixing the number of retrieved triples, we dynamically select a minimal subset of high-quality evidence whose cumulative normalized probability exceeds a threshold p (e.g., $p = 0.9$). This procedure includes a pre-filtering stage that removes extremely low-scoring triples via a sigmoid gate, followed by a softmax normalization over the remaining logits. The algorithm adaptively determines the smallest prefix of triples whose cumulative mass surpasses the threshold, subject to constraints on the minimum and maximum number of triples allowed. This filtering strategy prevents the LLM from being overwhelmed by irrelevant evidence while preserving enough structural information for multi-hop reasoning. The specific algorithm implementation can be found in the appendix B.1.

The final set of evidence is formatted together with their confidence scores and fed to the LLM. We find that confidence-conditioned prompting significantly improves the model's ability to prioritize reliable facts and reduces hallucinations, especially in long reasoning chains. Full prompt templates and qualitative examples—including both successful and failure cases—are provided in Appendix C, illustrating how the LLM leverages evidence of varying confidence levels during multi-hop reasoning.

5 Experiments

To thoroughly evaluate the effectiveness of our proposed QSRAG framework, we conduct extensive

	WebQSP		CWQ	
	Triple Recall	Answer Recall	Triple Recall	Answer Recall
SR+NSM w/ E2E	0.487	0.707	–	–
Retrieve-Rewrite-Answer	0.058	0.740	–	–
RoG	0.713	0.807	0.623	0.841
G-Retriever	0.294	0.545	0.183	0.375
GNN-RAG	0.522	0.818	0.500	0.841
SubgraphRAG	0.883	0.944	0.811	0.914
QSRAG	**0.906**	**0.951**	**0.914**	**0.974**

Table 1: Retrieval evaluation results on WebQSP and CWQ datasets. Best results are in **bold**.

experiments on two standard Knowledge Graph Question Answering datasets. This section details the experimental setup, evaluation results, and in-depth analysis.

5.1 Experimental Setup

Datasets. We use two widely adopted KGQA benchmarks: WebQuestionsSP (WebQSP)(Yih et al., 2016) and Complex WebQuestions (CWQ)(Talmor and Berant, 2018), both constructed over Freebase (Bollacker et al., 2008). WebQSP comprises 4,737 questions requiring up to two-hop reasoning, reflecting relatively simple to moderately complex queries. CWQ includes 34,699 questions with higher compositionality and multi-hop requirements. We followed the preprocessing and data splits of prior work such as RoG.

Evaluation Metrics. We evaluate performance at two levels: retrieval and reasoning. For retrieval, Triple Recall@k measures the proportion of retrieved top-k triplets that lie on the shortest path from the topic entity to the answer. Answer Recall@k measures the proportion of answer entities covered by the subgraph formed from the retrieved top-k triplets. For KGQA, we adopt standard evaluation metrics in the KGQA field: Micro F1 measures the overall F1 performance across all question-answer pairs, better reflecting the effectiveness on common questions and questions with many answers; Macro F1 averages the F1 for each individual question-answer pair, better reflecting the average performance of the method across different question types; Hit evaluates whether at least one of the answers generated by the model is correct; Hit@1 evaluates whether the model's most frequently predicted answer is correct according to any of the ground truth answers.

Baselines. We compare QSRAG against a broad set of KGQA approaches, ranging from simple retrieval strategies—Random Triplet Selection and Ground Truth Triplet Selection—to state-of-the-art graph-based and LLM-based methods. These include SR+NSM w/ E2E (Zhang et al., 2022), Retrieve-Rewrite-Answer (Wu et al., 2023), RoG (Luo et al., 2024), ToG (Sun et al., 2024), G-Retriever (He et al., 2024), GNN-RAG (Mavromatis and Karypis, 2025), and SubgraphRAG (Li et al., 2025), which employ path planning, rewriting, optimization, GNN-based reasoning, or directional structural encoding for retrieval. We also reference HGNet (Chen et al., 2023), a hierarchical query graph generation method, though its results are not included due to unavailable code and parameters. Together, these baselines span random, perfect, and advanced retrieval paradigms for comprehensive comparison.

Implementation Details. We conduct all model training and inference on a K100-AI cluster. For the retrieval stage, we select the top-k triplets with k values ranging from 50 to 500 and analyze the impact of different k-values in subsequent sections. In the reasoning stage, we employ several Large Language Models, including Llama-3.1-8B-Instruct, GPT-4o-mini-2024-07-18, and GLM-4-Flash, with the temperature parameter set to 0 during inference. While we primarily use the top 100 triplets for relevant operations, we also investigate representative k-values like 50 and 200, and present the results in the report. Additionally, we also adopt the top-p sampling strategy described earlier, where we set $p = 0.9$ and constrain the number of retained triples within a range of 50 to 200.

5.2 Evaluation Results

Retrieval Performance. Table 1 presents the retrieval results on WebQSP and CWQ. QSRAG

	WebQSP				CWQ			
	Micro-F1	Macro-F1	Hit	Hit@1	Micro-F1	Macro-F1	Hit	Hit@1
Random + Llama	16.57	31.97	50.12	45.33	22.69	22.09	28.43	25.83
Ground Truth + Llama	60.62	83.73	88.82	88.39	67.25	60.10	63.24	62.62
SR+NSM w/ E2E	–	64.10	–	–	–	46.30	–	–
ToG	–	–	82.60	–	–	–	**67.60**	–
Retrieve-Rewrite-Answer	–	–	79.36	–	–	–	–	–
G-Retriever	–	53.41	73.46	–	–	–	–	–
RoG	52.60	70.45	85.38	79.36	46.12	54.44	60.97	56.10
GNN-RAG	10.89	71.28	85.69	80.59	28.80	**59.08**	66.69	**61.34**
SubgraphRAG + Llama (200)	43.64	70.13	81.88	77.36	42.90	46.96	51.32	48.23
SubgraphRAG + GPT-4o-mini (200)	49.78	69.76	85.54	78.89	44.82	43.00	53.33	48.12
QSRAG + Llama	48.63	70.64	85.61	80.60	43.35	45.72	56.96	51.64
QSRAG + Llama (200)	49.05	70.14	85.56	79.57	40.80	44.59	57.16	51.06
QSRAG + Llama (Adaptive k)	49.97	70.12	85.26	78.69	47.74	49.56	60.32	54.74
QSRAG + GPT-4o-mini	55.26	69.56	83.11	76.41	51.42	46.75	54.40	49.56
QSRAG + GPT-4o-mini (200)	**55.52**	**71.83**	85.01	77.76	50.96	46.95	55.20	50.35
QSRAG + GPT-4o-mini (Adaptive k)	54.92	69.91	83.11	76.47	**52.35**	46.93	54.43	49.73
QSRAG + GLM-4-Flash	46.12	66.80	90.28	80.00	33.97	45.80	66.03	55.43
QSRAG + GLM-4-Flash (200)	43.52	66.48	**92.12**	**81.45**	29.03	45.71	67.48	55.84
QSRAG + GLM-4-Flash (Adaptive k)	49.15	68.39	90.38	80.27	38.53	47.76	66.56	56.03

Table 2: KGQA results on WebQSP and CWQ. **Bold** indicates the overall best results, while underline marks our best performance. Parentheses (200) denote the number of retrieved triples (default is 100). "Adaptive k" refers to dynamic triple selection.

consistently achieves the best performance across both Triple Recall and Answer Recall. On WebQSP, it attains a Triple Recall of 0.906, outperforming SubgraphRAG (0.883) and substantially surpassing RoG (0.713). Its Answer Recall reaches 0.951, slightly higher than SubgraphRAG's 0.944. The gains are even more pronounced on CWQ, where QSRAG improves Triple Recall to 0.914—well above SubgraphRAG (0.811) and RoG (0.623)—and achieves a new state-of-the-art Answer Recall of 0.974. These results demonstrate that the QR-GAT enables more precise multi-hop evidence retrieval and provides LLMs with significantly cleaner and more informative structured context.

Reasoning Performance. Table 2 summarizes the KGQA results on WebQSP and CWQ. QSRAG delivers clear and consistent improvements across LLMs and retrieval budgets. On WebQSP, the strongest configuration—QSRAG + GPT-4o-mini (200)—achieves a Micro-F1 of 55.52 and a Macro-F1 of 71.83. This corresponds to a +5.55% Micro-F1 and +1.96% Macro-F1 improvement over RoG (52.60 / 70.45), and an even larger +11.53% / +2.97% gain over SubgraphRAG + GPT-4o-mini (49.78 / 69.76). In terms of retrieval-sensitive indicators, QSRAG paired with GLM-4-Flash (200) attains the highest Hit (92.12) and Hit@1 (81.45), outperforming all prior systems and showing that our retriever supplies substantially more answer-relevant evidence.

On the more challenging CWQ dataset, QSRAG again provides notable improvements. The adaptive-k version with GPT-4o-mini achieves the highest Micro-F1 (52.35), outperforming RoG (46.12) by +13.51% and SubgraphRAG + GPT-4o-mini (44.82) by +16.80%. QSRAG also obtains competitive Macro-F1 scores and strong Hit/Hit@1 performance: QSRAG + GLM-4-Flash variants consistently reach Hit above 66.03, and Hit@1 above 55.43, rivaling or surpassing specialized graph models such as GNN-RAG and ToG. These consistent improvements across datasets, LLMs, and retrieval budgets highlight the strength of QSRAG's query- and relation-aware retrieval, enabling LLMs to reason more accurately over multi-hop KG evidence.

6 Further Analysis

To gain a deeper understanding of the contributions and robustness of each component in QSRAG, we conduct several analysis experiments; additional analyses are provided in the appendix.

TopK	WebQSP		CWQ	
	Triple Recall	Answer Recall	Triple Recall	Answer Recall
50	0.825	0.882	0.868	0.916
100	0.888	0.927	0.920	0.947
200	0.939	0.958	0.957	0.965
300	0.961	0.967	0.974	0.973
400	0.974	0.972	0.982	0.979
500	0.979	0.973	0.987	0.980

Table 3: Impact of Retrieved Top-k on Retrieval Performance (Recall@k).

TopK	WebQSP		CWQ	
	Macro-F1	Hit	Macro-F1	Hit
50	69.46	83.85	45.78	56.13
100	70.61	85.63	45.70	56.95
200	70.13	85.55	44.55	57.18
300	68.80	84.89	42.70	54.46
400	67.61	83.91	41.01	52.23
500	67.58	84.83	41.64	53.19

Table 4: Impact of Retrieved Top-k on Reasoning Performance with Llama-3.1-8B-Instruct.

Impact of Retrieved Top-k (Retrieval). Table 3 reports how varying the number of retrieved triples affects retrieval quality. On both WebQSP and CWQ, increasing k consistently improves Triple Recall and Answer Recall, confirming that a larger evidence pool captures more relevant multi-hop facts. For WebQSP, Triple Recall rises from 0.825 at $k = 50$ to 0.979 at $k = 500$, while Answer Recall increases from 0.882 to 0.973. CWQ shows a similar trend, with Triple Recall improving from 0.868 to 0.987 and Answer Recall from 0.916 to 0.980 as k grows from 50 to 500. These results highlight that broader retrieval ranges provide richer structural cues for downstream reasoning.

Impact of Retrieved Top-k (Reasoning). Table 4 shows how varying k affects QSRAG's reasoning performance (Macro-F1 and Hit) using `gte-large-en-v1.5` (Li et al., 2023). Unlike monotonic retrieval recall gains (Table 3), reasoning exhibits a rise-then-fall pattern: initial increases provide more useful evidence, but beyond a point, excessive triplets inject noise that overwhelms the LLM. WebQSP peaks at k=100 (70.61/85.63); CWQ peaks at k=50 for Macro-F1 (45.78) and k=200 for Hit (57.18). This divergence shows that while larger k improves recall, it can introduce distracting information, ultimately degrading QA performance. These observations informed our main-experiment k-value selections.

Model	WebQSP		CWQ	
	Triple Recall	Answer Recall	Triple Recall	Answer Recall
Base	0.857	0.906	0.880	0.945
Plus	0.906	0.951	0.914	0.974

Table 5: Ablation study of the QR-GAT module. `Plus` (our full model) incorporates query-relation interaction, showing consistent gains in both triple and answer recall over the `Base` attention mechanism.

Retriever Ablations. To evaluate the contribution of our query-relational attention module, we ablate the enhanced QR-GAT by removing its query–relation interaction term and retaining only the base attention component. The comparison between the full model (`Plus`) and the simplified variant (`Base`) is shown in Table 5. Across both datasets, the `Plus` model consistently outperforms the `Base` model in both Triple Recall and Answer Recall. On WebQSP, Triple Recall increases from 0.857 to 0.906 ($+5.71\%$), while Answer Recall improves from 0.906 to 0.951 ($+4.97\%$). On CWQ, the gains are similarly pronounced: Triple Recall rises from 0.880 to 0.914 ($+3.86\%$), and Answer Recall from 0.945 to 0.974 ($+3.07\%$). These consistent improvements highlight the effectiveness of injecting query-aware and relation-guided signals into the attention mechanism. The enhanced QR-GAT not only better identifies question-relevant triples but also retrieves more complete and answer-supportive subgraphs, contributing directly to downstream reasoning improvements.

7 Conclusion

In this work, we address the challenge of accurately retrieving structured evidence for knowledge graph-augmented generation by proposing the QSRAG framework, centered around the Query-Relational Graph Attention Network. QR-GAT enables precise triple scoring and subgraph retrieval through a query-aware and relation-guided attention mechanism. Experimental results demonstrate the strong retrieval performance of our method and its state-of-the-art or competitive KGQA results on both the WebQSP and CWQ datasets. This study demonstrates that faithfully modeling the query's semantics within graph attention is key to building reliable evidence pathways. Precise structured retrieval is not only beneficial but often essential for enabling large language models to reason accurately over knowledge graphs.

Limitations

While QSRAG achieves strong retrieval performance and effectively identifies question-relevant subgraphs, our work does not exhaust the design space of how retrieved triples should be organized and utilized during downstream reasoning. In this study, we mainly explore two simple strategies for constructing the reasoning input—top-k selection and probability–mass–based top-p filtering—and perform only a single retrieval pass followed by a single LLM generation step. We do not investigate more elaborate post-retrieval procedures such as re-ranking, denoising, evidence consolidation, or adaptive triple filtering, all of which may further reduce noise and mitigate the impact of irrelevant or spurious triples that occasionally persist in the retrieved set. As a result, part of the performance upper bound may be limited by the direct use of raw retrieved evidence. Enhancing QSRAG with advanced reasoning-stage evidence processing presents a promising avenue for future research.

Ethical Considerations

We confirm that we have fully complied with the ACL Ethics Policy in this study. Our research utilizes two publicly available datasets: WebQSP and CWQ. WebQSP comprises questions requiring up to two-hop reasoning over Freebase, while CWQ includes more complex, multi-hop questions also based on Freebase. All datasets used in this study have been extensively employed in knowledge graph question answering research and do not contain private, sensitive, or personally identifiable information. We carefully select these datasets to ensure ethical compliance and to mitigate potential biases. Our study does not involve the collection or modification of user-generated content, nor does it introduce synthetic data that could lead to unintended misinformation.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yongrui Chen, Huiying Li, Guilin Qi, Tianxing Wu, and Tenggou Wang. 2023. Outlining and filling: Hierarchical query graph generation for answering complex questions over knowledge graphs. *IEEE Trans. Knowl. Data Eng.*, 35(8):8343–8357.

Yifu Gao, Linbo Qiao, Zhigang Kan, Zhihua Wen, Yongquan He, and Dongsheng Li. 2024. Two-stage generative question answering on temporal knowledge graph using large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6719–6734. Association for Computational Linguistics.

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.

Tiezheng Guo, Qingwen Yang, Chen Wang, Yanyi Liu, Pan Li, Jiawei Tang, Dapeng Li, and Yingyou Wen. 2024. Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph. *Complex & Intelligent Systems*, 10(5):7063–7076.

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*.

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.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. GRAG: graph retrieval-augmented generation. *CoRR*, abs/2405.16506.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023a. 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.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. 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.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, and 1 others. 2023. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9410–9421. Association for Computational Linguistics.

Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *The Thirteenth International Conference on Learning Representations*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *CoRR*, abs/2308.03281.

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.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, and Jian Guo. 2024. Think-on-graph 2.0: Deep and interpretable large language model reasoning with knowledge graph-guided retrieval. *CoRR*, abs/2407.10805.

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.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242. Association for Computational Linguistics.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 641–651. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *CoRR*, abs/2309.11206.

Guanming Xiong, Junwei Bao, and Wen Zhao. 2024. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10561–10582. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 535–546. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5773–5784. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176.

A Additional Experiments

A.1 Multi-hop Reasoning Performance

We analyze QSRAG's behavior across different reasoning complexities by grouping test questions according to hop count and comparing its performance with SubgraphRAG. We report both Triple Recall and Answer Recall on WebQSP and CWQ.

Key Findings. Three clear patterns emerge from the results:

First, QSRAG achieves its largest gains on **2-hop questions**. On WebQSP, it improves 2-hop Triple Recall by +3.1% (0.771 vs. 0.748). On CWQ, where multi-hop structure dominates, the advantage is significantly amplified: +12.3% for Triple Recall (0.921 vs. 0.820) and +3.3% for Answer Recall (0.946 vs. 0.916). These improvements highlight the effectiveness of query- and relation-aware attention in capturing intermediate relational dependencies.

Second, QSRAG provides **consistent improvements across all hop levels** on the more complex CWQ dataset. Compared with SubgraphRAG, it improves Triple Recall by +12.5% on 1-hop, +10.1% on 2-hop, and +11.8% on 3+ hop questions. Answer Recall shows similar boosts: +3.1%, +3.0%, and +6.6% for 1-hop, 2-hop, and 3+ hops respectively. These gains indicate that QR-GAT's fine-grained triple scoring mitigates the retrieval degradation observed in long reasoning chains.

Third, although hop count captures structural complexity, it does not fully reflect the semantic diversity of multi-hop reasoning. Our current analysis does not distinguish compositional, comparative, or temporal questions. Conducting a more detailed semantic breakdown is an important next step for understanding QSRAG's behavior across reasoning categories.

Overall, these results show that QSRAG is particularly advantageous in multi-hop scenarios where structured relational interactions are essential, and its benefits become increasingly pronounced as reasoning complexity grows.

A.2 Inference Efficiency Comparison

We compare the inference efficiency of QSRAG with representative methods. All baseline latency numbers are taken directly from Table 1 of the SubgraphRAG paper, ensuring a fair and consistent comparison.

End-to-End Latency. According to the reported measurements in SubgraphRAG, iterative methods such as RoG (948 s per query) and GNN-RAG (68 s) incur substantial computational overhead due to multi-round retrieval and repeated LLM calls. SubgraphRAG, which adopts a single-pass retrieval followed by a single LLM invocation, achieves the best end-to-end latency among the baselines at only 6 seconds per query. Since QSRAG follows the same single-retrieval, single-inference paradigm, SubgraphRAG is the most appropriate efficiency baseline for comparison.

Retrieval Efficiency. Table 8 reports retrieval latency for QSRAG and SubgraphRAG. Although QSRAG's retrieval module is approximately 8–9× slower due to its query-aware graph-attention computation, both systems operate within the same practical time scale (0.01 s vs. 0.1 s). This indicates that the additional overhead introduced by QSRAG is small enough to be negligible in most real-world settings.

Key Observations. Since both QSRAG and SubgraphRAG execute exactly one retrieval step and one LLM inference step, the retrieval stage is where meaningful efficiency differences may arise. The results show that: (i) both systems remain within sub-second retrieval latency, keeping computational overhead minimal; (ii) QSRAG's additional 0.08–0.09 s cost is insignificant relative to the total query-processing pipeline; (iii) this small overhead yields substantial accuracy improvements over SubgraphRAG, including +2.6% Triple Recall on WebQSP and +12.7% on CWQ. Overall, QSRAG provides a strong accuracy–efficiency balance and remains a practical, low-latency KGQA solution despite adopting a heavier retrieval mechanism.

A.3 Impact of Different Encoders on Retrieval Performance

To better understand how the choice of encoder affects retrieval quality, we evaluate QSRAG using two widely adopted sentence encoders: **GTE-large-en-v1.5** and the **Qwen3-Embedding-0.6B encoder**. These encoders are used to obtain semantic embeddings of entities, relations, and the input question, which are then fed into QR-GAT. The rest of the retriever and graph structure remain unchanged. We examine Triple Recall@k and Answer Recall@k on both WebQSP and CWQ, using several representative values of k.

	Triple Recall		Answer Recall	
	1-hop (65.8%)	2-hop (34.2%)	1-hop (65.8%)	2-hop (34.2%)
SubgraphRAG	**0.953**	0.748	**0.977**	**0.881**
QSRAG	0.949	**0.771**	0.963	0.858

Table 6: Reasoning performance on WebQSP across different hop counts (with data distribution in parentheses).

	Triple Recall			Answer Recall		
	1-hop (28%)	2-hop (65.9%)	≥3-hop (6.1%)	1-hop (28%)	2-hop (65.9%)	≥3-hop (6.1%)
SubgraphRAG	0.831	0.820	0.626	0.946	0.916	0.741
QSRAG	**0.956**	**0.921**	**0.744**	**0.977**	**0.946**	**0.807**

Table 7: Reasoning performance on CWQ across different hop counts (with data distribution in parentheses).

Method	WebQSP	CWQ
QSRAG	0.1006 s	0.0920 s
SubgraphRAG	0.0115 s	0.0127 s

Table 8: Retrieval latency comparison.

TopK	WebQSP		CWQ	
	Triple Recall	Answer Recall	Triple Recall	Answer Recall
GTE-large-en-v1.5 Encoder				
50	0.845	0.879	0.827	0.880
100	0.900	0.919	0.886	0.911
200	0.946	0.948	0.934	0.942
300	0.963	0.959	0.954	0.954
400	0.974	0.967	0.964	0.961
500	0.980	0.971	0.971	0.966
Qwen-0.6B-Embedding Encoder				
50	0.825	0.882	0.868	0.916
100	0.888	0.927	0.920	0.947
200	0.939	0.958	0.957	0.965
300	0.961	0.967	0.974	0.973
400	0.974	0.972	0.982	0.979
500	0.979	0.973	0.987	0.980

Table 9: Retrieval performance of different query encoders on WebQSP and CWQ.

Analysis. From the results in Table 9, we observe that the choice of encoder has a substantial effect on retrieval performance, especially in terms of Triple Recall and Answer Recall:

GTE-large-en-v1.5 Encoder : This encoder performs notably better in Triple Recall on WebQSP, where it achieves 0.946 at $k = 200$, compared to Qwen3-Embedding-0.6B's 0.939. GTE-large-en-v1.5 excels in maintaining high Triple Recall across most k-values, making it particularly effective for retrieving precise triplet evidence for queries in WebQSP.

Qwen3-Embedding-0.6B Encoder : Despite GTE-large-en-v1.5's superiority in Triple Recall for WebQSP, the Qwen3-Embedding-0.6B encoder consistently outperforms in Answer Recall across both datasets, especially at $k = 50$, where it achieves a score of 0.916 on CWQ, and at $k = 100$ with 0.927 on WebQSP. This encoder is particularly effective in capturing the full context of a query, making it more robust for Answer Recall, which is crucial for high-quality question answering in complex datasets like CWQ.

Impact of Increasing k : As k increases, both encoders show improvements in both Triple Recall and Answer Recall, but the performance gap narrows. At higher k-values (e.g., 300, 400, 500), the retrieval performance of both encoders becomes more comparable. However, at smaller k-values (such as 50 or 100), Qwen3-Embedding-0.6B consistently demonstrates superior performance in Answer Recall, making it the preferred encoder for tasks that prioritize answer quality over triplet precision.

Given the results, the Qwen3-Embedding-0.6B encoder emerges as the optimal choice for our task. While GTE-large-en-v1.5 provides better Triple Recall in simpler query scenarios (e.g., WebQSP), Qwen3-Embedding-0.6B outperforms it in Answer Recall across both datasets, particularly on more complex tasks like CWQ. This encoder's ability to handle long and complex queries, while maintaining high accuracy in answer retrieval, makes it the ideal choice for our knowledge graph question answering system. Thus, we choose Qwen3-Embedding-0.6B for its robust performance in real-world, complex KGQA tasks, balancing high Answer Recall with competitive retrieval efficiency.

A.4 Reasoning Input Ablations.

In addition to simply removing confidence scores, we also explored various strategies to utilize these

scores, particularly focusing on filtering the retrieved Top-k triplets based on confidence thresholds. Table 10 presents a comparative analysis of the impact of different confidence filtering thresholds on reasoning performance, measured by Macro-F1 and Hit metrics. The methods compared include using no confidence scores with Llama, using confidence scores with Llama, using confidence scores with GLM-4-Air, and GLM-4-Air at different confidence threshold settings.

The filtering strategy `Confidence > Threshold` retains only triplets with scores above the specified threshold. This approach allows us to assess the effectiveness of confidence-based filtering in enhancing reasoning performance. Table 10 shows the results we obtained using the GTE-large-en-v1.5 encoder.

Analysis. From the results in Table 10, it is clear that the method using confidence scores with Llama consistently achieves the best performance on both datasets. In particular, both Macro-F1 and Hit metrics are significantly higher when confidence scores are incorporated, compared to when they are not used. This demonstrates that confidence scores provide valuable signals to LLMs, assisting in fine-grained integration of evidence.

The positive impact of confidence scores on performance can be attributed to several key factors: 1. Refining Relevant Triplet Selection: Confidence scores are generated by the QR-GAT encoder, which indicates how strongly each triplet is connected to the query context. This information allows the model to focus on the most relevant triplets, reducing the noise and enhancing the focus on meaningful facts. In turn, LLMs can utilize these high-confidence triplets to improve answer quality, as seen in the increased Hit and Macro-F1 scores with Llama.

2. Guiding LLM Attention: LLMs, like Llama and GLM-4-Air, process a limited context window and can struggle with noisy or irrelevant input. By incorporating confidence scores, the model can prioritize triplets with higher relevance, enabling more accurate reasoning and mitigating the risk of irrelevant paths contributing to hallucinations. This is evident in the improved performance when confidence scores are added, compared to the "No Confidence" baseline.

3. Improved Path Selection: Higher confidence triplets are more likely to belong to valid reasoning paths, which align better with the underlying knowledge graph structure. This reduces the potential for selecting incorrect or weakly correlated paths, enhancing both Triple Recall and Answer Recall, as the model focuses on high-quality evidence.

In contrast, when applying threshold filtering with GLM-4-Air, we observe a general degradation in performance as the threshold increases. For example, on WebQSP, increasing the threshold from 0.0001 to 0.1 results in a decrease in Macro-F1 from 64.09 to 60.11 and in Hit from 78.77 to 71.13. A similar trend is observed on CWQ. This suggests that overly aggressive filtering, which removes too many low-confidence triplets, inadvertently discards valuable information. While filtering is meant to reduce noise, it can also lead to the loss of relevant evidence, impacting the quality of downstream reasoning.

These findings reinforce the notion that confidence scores from QR-GAT provide critical signals for LLM reasoning, enabling the model to selectively focus on the most relevant evidence. The results also highlight that feeding all Top-k triplets along with their confidence scores yields the best performance, as it avoids the risk of filtering out useful, yet lower-confidence, triplets.

Incorporating confidence scores significantly improves reasoning performance by guiding the LLM to focus on more relevant triplets. Although threshold filtering can reduce noise, it may also discard useful information, leading to suboptimal performance.

A.5 Adaptive k

Table 11 shows the impact of different adaptive k-values on QSRAG's reasoning performance with Llama-3.1-8B-Instruct, evaluating both WebQSP and CWQ datasets. The adaptive k strategy adjusts the number of retrieved triplets based on the specific query context, aiming to optimize the balance between retrieval coverage and computational efficiency. Additionally, we employ topp sampling with $p = 0.9$ during the reasoning stage to further refine the triplet selection process. The values of k in our experiments correspond to the range of retrieved triplets, where triplets below or above these ranges are selected or truncated accordingly.

Analysis. From the Table 11, we observe that adaptive k values significantly improve performance compared to static retrieval strategies. For instance, the Optimal Retrieval Range (50-200)

Filter Strategy	WebQSP		CWQ	
	Macro-F1	Hit	Macro-F1	Hit
No Confidence + Llama	69.08	83.48	44.40	56.84
Confidence + Llama	**70.61**	85.63	**45.70**	56.95
QSRAG + GLM-4-Air	68.25	**88.85**	44.16	**63.15**
Confidence > 0.0001 + GLM-4-Air	64.09	78.77	33.94	42.98
Confidence > 0.001 + GLM-4-Air	64.18	78.43	35.63	45.02
Confidence > 0.01 + GLM-4-Air	63.37	76.17	36.77	46.58
Confidence > 0.1 + GLM-4-Air	60.11	71.13	34.40	41.55

Table 10: Reasoning Input Ablation: Impact of Confidence Filtering Thresholds on Reasoning Performance (Macro-F1 and Hit).

	WebQSP				CWQ			
	Micro-F1	Macro-F1	Hit	Hit@1	Micro-F1	Macro-F1	Hit	Hit@1
20–200	48.07	69.34	81.70	77.40	48.38	48.85	57.97	52.79
50–200	49.97	70.12	85.26	78.69	47.74	49.56	60.32	54.74
50–300	49.82	70.20	85.81	79.36	46.00	48.52	59.27	54.15

Table 11: Adaptive k with Llama-3.1-8B-Instruct. The adaptive strategy adjusts the retrieved triplet range based on query context to optimize retrieval performance. $p = 0.9$ is used for top-p sampling during reasoning.

achieves the highest Micro-F1 score (49.97) and Macro-F1 score (70.12) on WebQSP, outpacing other configurations. In terms of Answer Recall (Hit and Hit@1), the Optimal Retrieval Range also demonstrates superior performance, with a notable improvement in Hit (85.26) and Hit@1 (78.69) compared to the Low-Moderate Retrieval Range (20-200) configurations.

On the CWQ dataset, the Optimal Retrieval Range (50-200) continues to outperform other values in terms of Hit@1 (54.74) and Hit (60.32), while still maintaining competitive Micro-F1 and Macro-F1 scores. Notably, while the High Retrieval Range (50-300) configuration has a slight advantage in some metrics, the Optimal Retrieval Range strikes a good balance between recall and computational efficiency, especially on WebQSP.

The results suggest that adjusting the retrieval range based on query characteristics allows QSRAG to focus on the most relevant triplets, improving retrieval quality and reasoning accuracy without excessive computational overhead. This adaptive approach is particularly beneficial for datasets with varying query complexity, such as CWQ, where fine-tuning the number of retrieved triplets enhances performance.

In summary, the adaptive k strategy improves both the precision of triplet selection and reasoning efficiency, demonstrating that a dynamic approach to retrieval is effective in balancing performance and resource usage.

B Implement Details

B.1 Top-p Adaptive Retrieval Strategy

To dynamically determine the number of evidence triplets retrieved for each query, we adopt a Top-p (nucleus sampling)–based adaptive selection method during the retrieval stage, as shown in Algorithm 1. This strategy replaces the traditional fixed Top-k approach with a probability–mass–driven mechanism that adapts the retrieval scope according to the model's confidence distribution.

Given the logits predicted by QR-GAT for all candidate triplets, we first apply a sigmoid-based pre-filter to remove extremely low-confidence candidates. We then perform a softmax over the remaining logits only, compute the cumulative probability mass, and identify the smallest prefix that exceeds the Top-p threshold $p = 0.9$. This nucleus size is further constrained within $[K_{\min}, K_{\max}] = [50, 300]$, ensuring both stability and robustness. The final selected triplets and their associated sigmoid scores are passed to the downstream reasoner.

This Top-p–based adaptive strategy allows the retriever to adjust the evidence size according to the uncertainty of each query. When the model is highly certain, the nucleus is small; when the query

Algorithm 1 Adaptive Triple Retrieval with K-limit

Require: Q, $\mathcal{G}$, thresholds τ_P, τ_A, and limits $K_{\min}, K_{\max}$
Ensure: Retrieved triples set $\mathcal{T}_{ret}$

```
l = Model(Q, G) {Compute logits}
s = σ(l) {Sigmoid scores for filtering}
I_pass = {i | s_i > τ_P} {Pre-filtering}
if I_pass is empty then
    return ∅
end if
N = |I_pass|
p = softmax({l_i | i ∈ I_pass}) {Relative probs}
Sort p descending: (p_(0), ..., p_(N−1)) with indices I_sort
Compute cumulative sum: c_k = Σ_{j=0}^{k} p_(j)
Find k_A = min{k | c_k > τ_A} (if none, k_A = N − 1)
k_final = min(max(k_A + 1, K_min), K_max, N)
I_sel = {I_pass[I_sort[j]] | 0 ≤ j < k_final}
return T_ret = {(h_i, r_i, t_i, s_i) | i ∈ I_sel}
```

requires broader reasoning, the retrieved evidence naturally expands. This eliminates the need for task-specific tuning of k and improves retrieval quality under varied query complexities.

This adaptive retrieval algorithm ensures that the retriever captures sufficient evidence for multi-hop reasoning while preventing context overflow or noise accumulation in the LLM input. The softmax is applied only on the filtered subset, making nucleus detection more concentrated on semantically relevant triplets. The $[K_{\min}, K_{\max}]$ constraint further guarantees stable performance across different datasets, making this strategy robust and effective for knowledge graph question answering.

B.2 Knowledge Graph Visualization

In this section, we present knowledge graph visualizations that demonstrate the effectiveness of our method in handling both single-hop and multi-hop questions. The focus is on target triples, with confidence scores annotated on the edges to show how our model utilizes relevant knowledge for reasoning. The following figures provide examples from two datasets: WebQSP and CWQ.

The first set of visualizations, shown in Figure 3, illustrates examples from the CWQ dataset. The first example asks, "What is the type of government practices in the country where the Israeli Lira is used?" In this case, the question node Israeli Lira links to the intermediate node Israel with a confidence score of 1, and Israel then links to the answer node Parliamentary system with a score of 0.51. This example demonstrates the ability of our model to handle multi-hop reasoning, where intermediate entities provide valuable context for answering complex queries.

The second example, "Which movie with a character called Ajila was directed by Angelina Jolie?" shows how the question node Ajila links to the intermediate node m.0gw7h9w with a confidence score of 0.98. The intermediate node then points to the answer node In the Land of Blood and Honey with a score of 0.84. This demonstrates the model's ability to perform multi-hop reasoning by correctly linking the character Ajila to the movie through an intermediate node.

The second set of visualizations, shown in Figure 4, demonstrates examples from the WebQSP dataset, which primarily focuses on single-hop reasoning. The first example asks, "What do Jamaican people speak?" Here, the question node Jamaica links to two answer nodes: Jamaican English (score: 0.87) and Jamaican Creole English Language (score: 0.97). These answer nodes link back to the question node with scores of 0.99 and 0.55, respectively. This illustrates the effectiveness of our model in identifying and linking relevant entities in a single-hop query.

The second example, "What else did Benjamin Franklin invent?" demonstrates how the question node Benjamin Franklin links to multiple answer nodes with confidence scores higher than 0.7, which indicates strong evidence and supports the accuracy of the single-hop reasoning process.

These visualizations illustrate how our method effectively utilizes the knowledge graph to support both single-hop and multi-hop reasoning. The confidence scores on the edges allow us to track how the model selects and uses relevant information for reasoning, improving the overall quality of the answers.

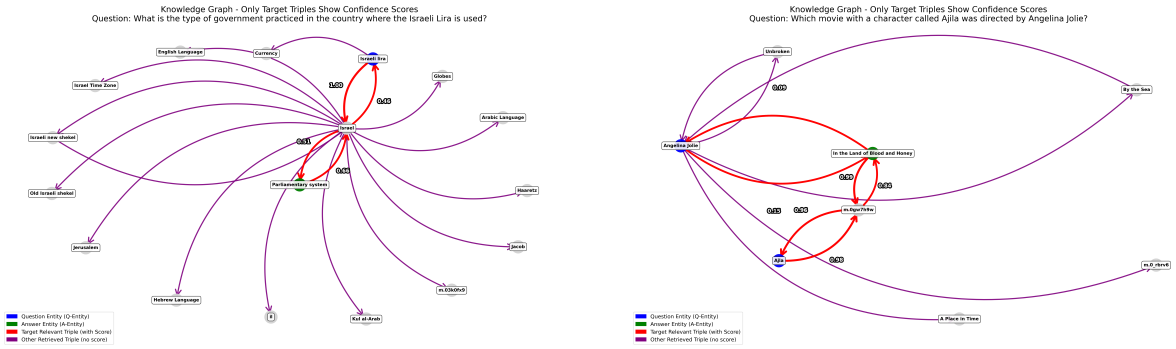

Figure 3: Knowledge graph visualizations for CWQ examples.

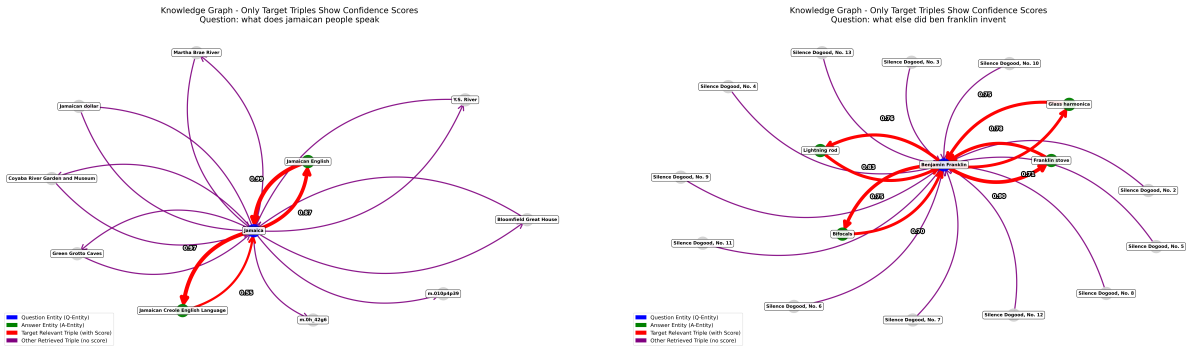

Figure 4: Knowledge graph visualizations for WebQSP examples.

C Question-answer Examples.

In this section, we provide representative question–answer examples to illustrate how our retrieval and reasoning pipeline behaves in practice. Figure 5 first shows the prompt template used by our system, including the structure for presenting retrieved triplets and confidence scores to the LLM.

Following the prompt template, we present four correct QA examples that highlight the strengths of our approach in both multi-hop and single-hop scenarios. These examples demonstrate how confidence-aware triplet retrieval guides the model toward accurate reasoning paths.

Finally, we include two incorrect QA examples. These cases reveal several typical sources of failure in our system: incomplete or missing answer entities in the underlying knowledge graph, the presence of non-textual or low-quality entities that disrupt semantic matching, and occasional retrieval gaps that prevent the model from assembling a complete reasoning chain. Together, these examples outline the practical limitations imposed by the KG itself as well as the retriever, offering a clearer view of the boundary conditions of our approach.

System:
You are a reasoning assistant. Based on the given knowledge graph triple and its confidence, your task is to answer the question.
You must use only the entities found in the triplets that are **meaningful names** (e.g., 'Aviva Stadium', not 'm.0wz2kl3').
Each answer must be a full entity name as it appears in the triplets.
Return each answer in a new line, prefixed with 'ans:'.
If the answer is not present or is an internal ID (like 'm.0xxx'), respond with 'ans: not available'.

User:
Input Triplets:
(Lou Seal,sports.mascot.team,San Francisco Giants)[Confidence: 0.9999743700027466]
(San Francisco Giants,sports.sports_team.team_mascot,Lou Seal)[Confidence: 0.9999568462371826]
(2014 World Series,sports.sports_championship_event.champion,San Francisco Giants)[Confidence: 0.8941453695297241]
(2012 World Series,sports.sports_championship_event.champion,San Francisco Giants)[Confidence: 0.8416239619255066]
(2010 World Series,sports.sports_championship_event.champion,San Francisco Giants)[Confidence: 0.8150305151939392]
(San Francisco Giants,sports.sports_team.championships,2014 World Series)[Confidence: 0.7321670055389404]
(San Francisco Giants,sports.sports_team.championships,2012 World Series)[Confidence: 0.6976805329322815]
(San Francisco Giants,sports.sports_team.championships,2010 World Series)[Confidence: 0.6664191484451294]
(San Francisco Giants,time.participant.even2t,2012 World Series)[Confidence: 0.07613193988800049]
(San Francisco Giants,time.participant.event,2010 World Series)[Confidence: 0.07032240927219391]
…
Question: What year did the team with mascot named Lou Seal win the World Series?

Assistant:
Based on the given triplets, we can determine that Lou Seal is the mascot of the San Francisco Giants, as indicated by the high-confidence relation (Lou Seal, sports.mascot.team, San Francisco Giants) [Confidence: 0.9999743700027466]. Furthermore, the San Francisco Giants are recorded as champions of the 2014 World Series [Confidence: 0.8941453695297241], the 2012 World Series [Confidence: 0.8416239619255066], and the 2010 World Series [Confidence: 0.8150305151939392]. Therefore, the team with mascot named Lou Seal won the World Series in the years corresponding to those championships.
ans: 2014 World Series
ans: 2012 World Series
ans: 2010 World Series

User:
Input Triplets:
(h1,r1,h2)[Confidence:0.96]
(h3,r2,h2)[Confidence:0.83]
…
Question: Which … ?

Please provide your answer in the same format as the example. The answer should be a meaningful entity or information, not a random string like 'm02h98gq' or 'm.0xxx'.Please list all reasonable answers from the above triplets. If the answer is not present in the triplets, respond with 'ans: not available'.

Figure 5: Input prompts for KGQA.

CWQ-WebQTest-576_01e2da60a2779c4ae4b5d1547499a4f8

Question:
Which nation has the Alta Verapaz Department and is in Central America?

Retrieved Triples:
Input Triplets:
(Guatemala, location.country.administrative_divisions, Alta Verapaz Department)[Confidence:0.43120062351226807]
(Guatemala, location.location.containedby, Central America)[Confidence: 0.4033266603946686]
(Guatemala, base.aareas.schema.administrative_area.administrative_children, Alta Verapaz Department)[Confidence:0.393011152744]
(Guatemala, common.topic.notable_types, Country)[Confidence: 0.30294546484947205]
(Alta Verapaz Department, location.administrative_division.country, Guatemala)[Confidence: 0.16839228570461273]
(Central America, location.location.contains, Costa Rica)[Confidence: 0.14474257826805115]
(Central America, location.location.contains, Guatemala)[Confidence: 0.13684657216072083]
…
Question: Which nation has the Alta Verapaz Department and is in Central America?

LLM Response:
Based on the given triplets, we can determine the following:
1. Guatemala is an administrative division that contains Alta Verapaz Department.
2. Guatemala is also located in Central America.
Therefore, the nation that has the Alta Verapaz Department and is in Central America is Guatemala.The formatted answer is:

ans: Guatemala

Ground-truth:
Guatemala

Figure 6: Correct question-answer example 1.

CWQ-WebQTest-100_524908899a8aa334a18a0ac00f8f2fe6

Question:
People from the country that contains Nord-Est Department speak what languages today?

Retrieved Triples:
Input Triplets:
(French, language.human_language.countries_spoken_in, Haiti)[Confidence: 0.8332462310791016]
(Haiti, base.aareas.schema.administrative_area.administrative_children, Nord-Est Department)[Confidence: 0.8295478224754333]
(Nord-Est Department, location.location.containedby, Haiti)[Confidence: 0.7992613315582275]
(Haiti, location.country.first_level_divisions, Nord-Est Department)[Confidence: 0.7959402203559875]
(Haitian Creole, language.human_language.countries_spoken_in, Haiti)[Confidence: 0.7403706908226013]
(Haiti, location.country.languages_spoken, Haitian Creole)[Confidence: 0.7223041653633118]
(Nord-Est Department, location.administrative_division.country, Haiti)[Confidence: 0.7090997695922852]
…
Question: People from the country that contains Nord-Est Department speak what languages today?

LLM Response:
Based on the given triplets, the country that contains Nord-Est Department is Haiti. The languages spoken in Haiti today are Haitian Creole and French.
Therefore, the formatted answers are:

ans: Haitian Creole
ans: French

Ground-truth:
Haitian Creole
French

Figure 7: Correct question-answer example 2.

WebQSP-WebQTest-1656

Question:
What are some of leonardo da vinci most famous inventions?

Retrieved Triples:
Input Triplets:
(Aerial Screw, law.invention.inventor, Leonardo da Vinci)[Confidence: 0.9940758943557739]
(Anemometer, law.invention.inventor, Leonardo da Vinci)[Confidence: 0.9930522441864014]
(Diving suit, law.invention.inventor, Leonardo da Vinci)[Confidence: 0.9929106831550598]
(Triple Barrel Canon, law.invention.inventor, Leonardo da Vinci)[Confidence: 0.9921991229057312]
(33-Barreled Organ, law.invention.inventor, Leonardo da Vinci)[Confidence: 0.9916179776191711]
(Double hull, law.invention.inventor, Leonardo da Vinci)[Confidence: 0.9903085827827454]
(Leonardo da Vinci, law.inventor.inventions, Aerial Screw)[Confidence: 0.9774912595748901]
(Leonardo da Vinci, law.inventor.inventions, Anemometer)[Confidence: 0.9757876396179199]
…
Question: What are some of leonardo da vinci most famous inventions?

LLM Response:
Based on the given triplets, Leonardo da Vinci is credited with inventing several items. Here are some of his most famous inventions listed in the triplets:

ans: Aerial Screw
ans: Anemometer
ans: Diving suit
ans: Triple Barrel Canon
ans: 33-Barreled Organ
ans: Double hull
ans: Armored car
ans: Parachute
ans: Ornithopter
ans: Viola organista

Ground-truth:
Parachute
Diving suit
Aerial Screw
Viola organista
Double hull
33-Barreled Organ
Triple Barrel Canon
Armored car
Ornithopter
Anemometer

Figure 8: Correct question-answer example 3.

WebQSP-WebQTest-801

Question:
What kind of government is egypt?

Retrieved Triples:
Input Triplets:
(Egypt, location.country.form_of_government, Provisional government)[Confidence: 0.9987523555755615]
(Egypt, location.country.form_of_government, Semi-presidential system)[Confidence: 0.9983388185501099]
(Provisional government, government.form_of_government.countries, Egypt)[Confidence: 0.9960112571716309]
(Semi-presidential system, government.form_of_government.countries, Egypt)[Confidence: 0.9928788542747498]
(Egypt, base.aareas.schema.administrative_area.administrative_parent, Earth)[Confidence: 0.00018889721832238138]
(Egypt, location.country.currency_used, Egyptian pound)[Confidence: 3.1110161216929555e-05]
…
Question: What kind of government is egypt?

LLM Response:
Based on the given triplets, there are two forms of government associated with Egypt:
1. Provisional government
2. Semi-presidential system
Therefore, the answers are:

ans: Provisional government
ans: Semi-presidential system

Ground-truth:
Provisional government
Semi-presidential system

Figure 9: Correct question-answer example 4.

WebQSP-WebQTest-402

Question:
What does joey jordison play in slipknot?

Retrieved Triples:
Input Triplets:
(Slipknot, music.musical_group.member, m.010kt51y)[Confidence: 0.6338161826133728]
(Slipknot, music.musical_group.member, m.0h2_x53)[Confidence: 0.28905683755874634]
(Slipknot, music.musical_group.member, m.01ttvxs)[Confidence: 0.12631060183048248]
(Slipknot, music.musical_group.member, m.05nn4sx)[Confidence: 0.12411950528621674]
(Slipknot, music.musical_group.member, m.0h2_x57)[Confidence: 0.10983577370643616]
(m.010kt51y, music.group_membership.group, Slipknot)[Confidence: 0.10371658951044083]
(Slipknot, music.musical_group.member, m.05nn4tj)[Confidence: 0.09712643176317215]
…
Question: What does joey jordison play in slipknot?

LLM Response:
Joey Jordison is a member of Slipknot and has played various roles in the band. According to the triplets provided, Joey Jordison has played the following roles in Slipknot:Lead guitar,Backing Vocals,Vocals,Bass guitar.
Therefore, the answers are:

ans: Lead guitar
ans: Backing Vocals
ans: Vocals
ans: Bass guitar

Ground-truth:
Drums

Figure 10: Incorrect question-answer example 1.

CWQ-WebQTrn-1259_1997cb4922db71983be26e6a509950f4

Question:
Where did the "Country Nation World Tour" concert artist go to college?

Retrieved Triples:
Input Triplets:
(Brad Paisley, people.person.place_of_birth, Glen Dale)[Confidence: 0.9620670676231384]
(Glen Dale, location.location.people_born_here, Brad Paisley)[Confidence: 0.9496924877166748]
(Brad Paisley, music.artist.origin, Glen Dale)[Confidence: 0.6797878742218018]
(Country Nation World Tour, music.concert_tour.artist, Brad Paisley)[Confidence: 0.33858561515808105]
(Colorado State University, common.topic.notable_types, College/University)[Confidence: 0.3122248351573944]
(University of California, Berkeley, common.topic.notable_types, College/University)[Confidence:0.1940333992242813]
…
Question: Where did the "Country Nation World Tour" concert artist go to college?

LLM Response:
Based on the given triplets, we need to find the college Brad Paisley attended. The triplets provide information about Brad Paisley's education, including the institution he attended.
From the triplets:
- Brad Paisley, people.person.education, m.0h3d7qj
- m.0h3d7qj, education.education.degree, Bachelor's degree
- m.0h3d7qj, education.education.student, Brad Paisley

The triplet indicates that Brad Paisley received a Bachelor's degree, but it does not specify the institution he attended. The identifier 'm.0h3d7qj' does not correspond to a meaningful name of a college or university in the triplets.
Therefore, the answer is:

ans: not available

Ground-truth:
Belmont University

Figure 11: Incorrect question-answer example 2.