Enrich-on-Graph: Query-Graph Alignment for Complex Reasoning with LLM Enriching

Anonymous ACL submission

Abstract

Large Language Models (LLMs) exhibit strong reasoning capabilities in complex tasks. However, they still struggle with hallucinations and factual errors in knowledge-intensive scenarios like knowledge graph question answering (KGQA). We attribute this to the semantic gap between structured knowledge graphs (KGs) and unstructured queries, caused by inherent differences in their focuses and structures. Existing methods usually employ resource-intensive, non-scalable workflows reasoning on vanilla KGs, but overlook this gap. To address this challenge, we propose a flexible framework, Enrich-on-Graph (EoG), which leverages LLMs' prior knowledge to enrich KGs, bridge the semantic gap between graphs and queries. EoG enables efficient evidence extraction from KGs for precise and robust reasoning, while ensuring low computational costs, scalability, and adaptability across different methods. Furthermore, we propose three graph quality evaluation metrics to analyze query-graph alignment in KGQA task, supported by theoretical validation of our optimization objectives. Extensive experiments on two KGQA benchmark datasets indicate that EoG can effectively generate high-quality KGs and achieve the state-of-the-art performance.

1 Introduction

Large language models (LLMs) (Achiam et al., 2023; Brown et al., 2020; Chowdhery et al., 2023) excel in complex natural language processing (NLP) tasks (Wei et al., 2022; Khot et al., 2022) due to extensive pre-training on large corpora (Rawte et al., 2023) embedding prior knowledge in their parameters (Khot et al., 2022). Based on the prior knowledge, LLMs can achieve semantic understanding, perform reasoning, and generate reasonable responses in diverse question-answering tasks (Li et al., 2024). However, LLMs still face challenges like hallucinations and factual errors (Ji et al., 2023), particularly in knowledge-intensive scenarios like knowledge graph question answering.

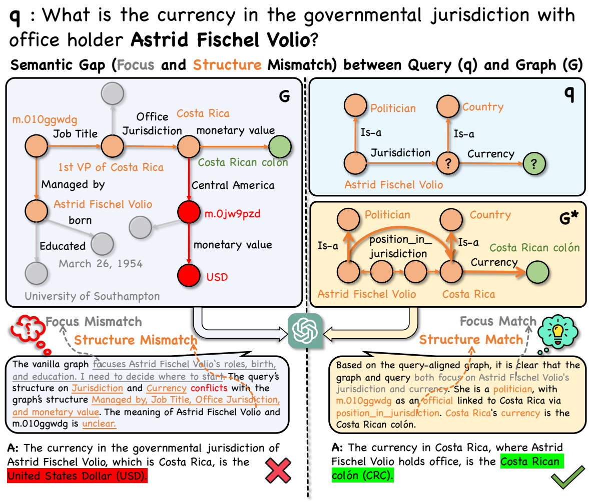

Figure 1: Semantic gap between query and graph: Gray indicates noise, red represents errors, orange denotes reasoning-related information, and green is answer. We use G, Q, G^* to represent vanilla graph, query, query-aligned graph, respectively, and use their logic forms for illustration. **Left:** LLMs misextracts key information due to the semantic gap between Q and G. **Right:** EoG generates G^* for efficient LLM reasoning.

Knowledge graph question answering (KGQA) is the task of answering natural language queries based on structured factual information stored in knowledge graphs (KGs) (Auer et al., 2007; Bollacker et al., 2008a). Existing KGQA methods can be broadly categorized into information retrieval-based and semantic parsing-based. Information retrieval (Sun et al., 2019; Zhang et al., 2022) methods extract subgraphs relevant to a query and reason over them, but retrieval process inevitably introduces noise, thereby reducing accuracy of answer. Semantic parsing methods (Sun et al., 2020; Jiang et al., 2023b) generate logical forms (e.g., SPARQL) for querying KGs. However, both approaches remain limited by the reasoning capabilities of their underlying models, especially for complex queries.

Given LLMs' strong understanding and reasoning capabilities, advanced methods usually employ LLMs for KGQA. *We believe that the core of such methods lies in how to bridge the* ***Semantic Gap Between Queries and Knowledge Graphs***, which stems from the focus and structure mismatch. User query is a precise, goal-driven request with clear semantic focus. In contrast, KGs encompass diverse focuses across many topics, often containing substantial noisy information. This focus mismatches between the query and the KG makes it challenging to accurately retrieve the subgraphs needed for reasoning. Even when relevant subgraphs are retrieved, the rigid structure of KGs often clashes with the linguistic diversity of user queries, complicating reasoning. As shown in Fig. 1 Left, the vanilla graph contains significant noise (gray entities), which is mismatched with the query focus. The query, *"What is the currency in the governmental jurisdiction with office holder Astrid Fischel Volio?"*, requires a two-hop reasoning path (*Jurisdiction* $\rightarrow$ *Currency*), but the vanilla graph uses a 4-hop path (*Managed* $\rightarrow$ *Job* $\rightarrow$ *Office Jurisdiction* $\rightarrow$ *Monetary Value*) and involves ambiguous entity *Astrid Fischel Volio*, creating a structure mismatch. Consequently, reasoning over vanilla KGs is hindered by focus and structure mismatches.

Previous methods struggle to align the semantics between queries and graphs from the perspective of designing complex reasoning pipelines, which can be effective at times but come with high computational or training costs that hinder their efficiency. For example, the DoG (Ma et al., 2025) framework iteratively simplifies queries and focuses on subgraphs through steps like Invoking, Filtering, Answer Trying, and Simplifying, enabling step-by-step reasoning but leading to a rigid and bulky workflow. RoG (Luo et al., 2024) requires fine-tuning LLMs for relationship path planning, which extracts query-relevant subgraph relations for faithful reasoning, but making adaptation to different KGs costly due to retraining. ToG (Sun et al., 2023) leverages LLMs to explore and reason over entities and relations in KG based on the query in an iterative manner. These methods attempt to align the semantics between queries and graphs through reasoning, but they still suffer from the semantic gap when reasoning over vanilla KGs.

To address the challenge, we propose Enrich-on-Graph (EoG), a flexible three-stage framework that leverages LLMs' prior knowledge to enrich graph, aligning the semantics between the vanilla KGs and queries. EoG proceeds in three stages: (1) **Parsing:** parsing the query and graph to enable effective alignment; (2) **Pruning:** proposing focus-aware multi-channel pruning to mitigate focus mismatches; (3) **Enriching:** leveraging LLMs to enrich graph to resolve structure mismatches. The aligned graph then collaborates with LLMs as an external knowledge source for efficient reasoning, shown in Fig. 1 Right. We also introduce three graph quality metrics—**Relevance, Semantic Richness**, and **Redundancy**—to evaluate graphs' structural properties, feature attributes, and LLM-driven aspects, supported by theoretical analyses further linking these metrics to optimization objectives. The major contributions of this work are as follows:

- We highlight the key of KGQA task lies in how to bridge the semantic gap between queries and KGs. We further propose three graph quality metrics—Relevance, Semantic Richness, and Redundancy. Theoretical analyses validate alignment mechanism's effectiveness and the practical utility of these metrics.
- We propose a flexible Enrich-on-Graph (EoG) framework that leverages LLMs as prior to align the semantics between queries and graphs, enabling high-quality KG generation and precise reasoning.
- Extensive experiments on two KGQA benchmark datasets demonstrate that EoG achieves the state-of-the-art performance through generating high-quality KGs, while ensuring flexibility, low computational cost, scalability, and broad compatibility with other methods.

2 Task Formulation and Analysis

2.1 Task Formulation

Given a KG $G = \{(e_s, r, e_o) | e_s, e_o \in E, r \in R\}$, where E and R are the set of entities and relations, it stores a large amount of factual knowledge in the form of triples. For a complex query q, the goal of the KGQA method M_θ with parameters θ is to get the correct answer a^* that $a^* = M_\theta(q, G)$.

2.2 Semantic Gap between Query and Graph

To better illustrate that the semantic gap between the KG and the query arises from **focus mismatch**

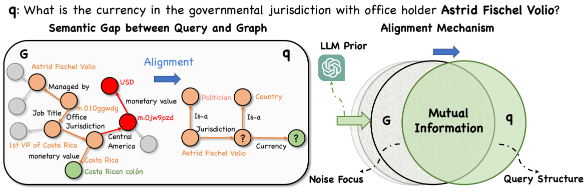

Figure 2: **Left:** focus mismatch and structure mismatch between query and vanilla graph cause semantic gaps. **Right:** Demonstration of EoG's alignment mechanism.

and **structure mismatch**, we refer to previous works (Wang et al., 2023; Zhang et al., 2024) to visualize their logical forms, as shown in Fig. 2 (left).

Focus Mismatch: For the query *"What is the currency in the governmental jurisdiction with office holder Astrid Fischel Volio?"*, it explicitly focuses on the jurisdiction managed by *Astrid Fischel Volio* and its currency, represented as: *Astrid Fischel Volio → Jurisdiction → ? → Currency → ?*. However, the vanilla KG includes irrelevant focuses, such as *Astrid Fischel Volio*'s birthday and education (gray nodes), which we call focus mismatch since they do not help the reasoning process.

Structure Mismatch: As shown in Fig. 2 (left), the vanilla KG contains 4-hop structures (orange nodes): *Astrid Fischel Volio → Managed → m.010ggwdg → Job → 1st VP of Costa Rica → Office Jurisdiction → Costa Rica → Monetary Value → Costa Rican colón*. In contrast, the query has corresponding 2-hop structures: *Jurisdiction → Currency*. The **structure inconsistency in reasoning hops** (e.g., the query's single-hop *Jurisdiction* versus KG's 3-hop *Managed → Job → Office Jurisdiction*) makes multi-hop reasoning harder for LLMs. Additionally, ambiguous entities like *Astrid Fischel Volio* (**lacking a hierarchy structure** such as *Astrid Fischel Volio → is-a → Politician*) further confuse LLMs. These issues are referred to as structure mismatch. In summary, **the semantic gap between the vanilla KG and the query** stems from **focus mismatch** and **structure mismatch**.

3 Method

3.1 Solution of EoG

As shown in Fig. 2 (left), ideally, for complex KGQA tasks, we want to use a graph G^* that is consistent with the logic form of q to help LLMs reason efficiently. However, in reality, we can only use the vanilla KG G, letting LLMs leverage external knowledge from G and its pre-trained knowledge to find information relevant to q for answering. Thus, our goal is to find an optimized graph G^* by maximizing the expected posterior probability:

$$G^* = \underset{G}{argmax}\ \mathbb{E}_{P(q,G)} \left[P\left(M_\theta, q|G\right) \right]$$

By identifying the optimized graph G^*, the LLM M_θ can directly extract key knowledge from G^* to better perform reasoning on q and arrive at a^*.

Theorem 1 *Maximizing the expected posterior probability is equivalent to maximizing the mutual information (MI) between q and G.*

$$P\left(M_\theta, q|G\right) \propto \log\left(\frac{P\left(M_\theta, q|G\right)}{P(q)}\right) = \log\left(\frac{P\left(M_\theta, G|q\right)}{P(G)}\right)$$

$$\propto \log\left(\frac{\int_{M_\theta} P\left(M_\theta, G|q\right) dM_\theta}{P(G)}\right) = \log\left(\frac{P\left(q, G\right)}{P(q)P(G)}\right)$$

Therefore, we can obtain (Details in Appendix C):

$$\mathbb{E}_{P(q,G)} \left[P\left(M_\theta, q|G\right)\right] \propto MI(q, G)$$

As shown in Fig. 2 (right), through maximizing $MI(q, G)$, G eliminates focus mismatch and structure mismatch, resulting in the desired G^*.

3.2 Our Method: Enrich-on-Graph (EoG)

As shown in Fig. 3, given a user query q and a vanilla graph G, our method consists of three stages: **(1) Parsing:** we parse q into new forms Q, while converting G into a query form Q^G (which we refer to as the graph query) and injecting it into triples to construct quadruples, preparing for the better semantic alignment. **(2) Pruning:** Using the Q from the first stage, we prune noisy graph focuses via multiple channels, eliminating focus mismatches. **(3) Enriching:** Leveraging LLMs with parametric knowledge as prior, Q^G related to the Q are identified and corresponding structures in quadruples are semantically aligned using KG's structural and feature properties, thus enhancing reasoning.

3.2.1 Query-Oriented Knowledge Parsing

The complex query q is in natural language, while G is in triple form, making their formats different and hindering subsequent semantic alignment. To address this, we use LLMs to convert q and G into the same query format, Q and Q^G.

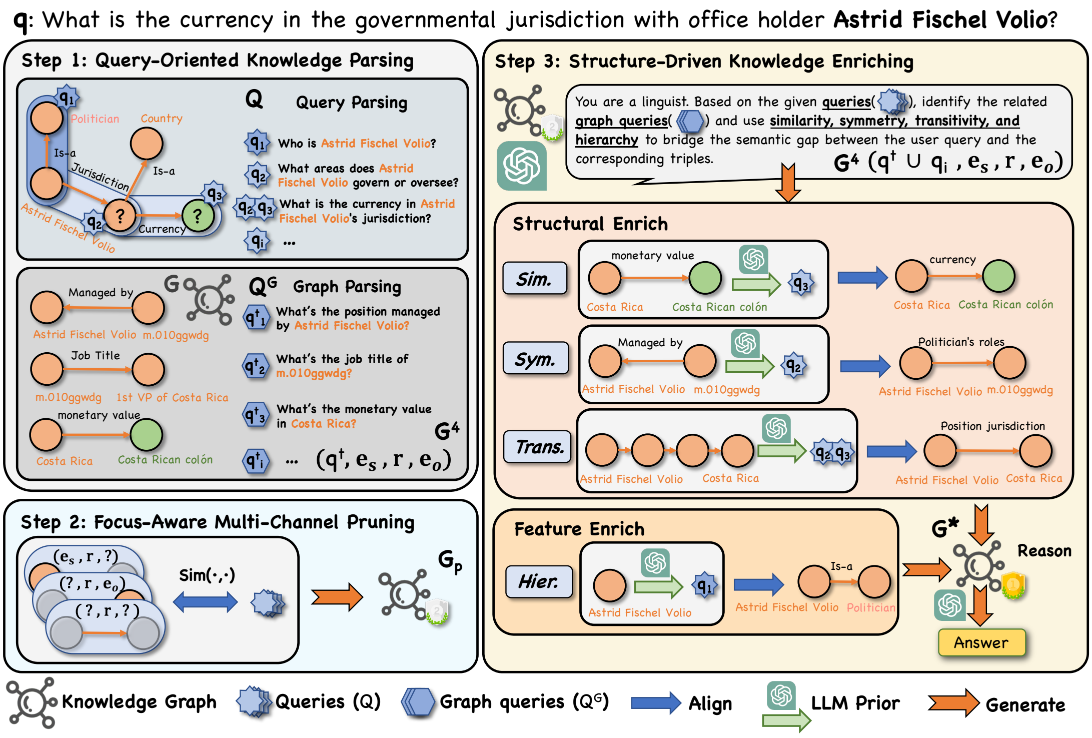

Figure 3: Overview of our Enrich-on-Graph framework.

Query Parsing: LLM breaks down q into sub-queries, which are categorized into two types: compound queries (requiring multi-hop reasoning) and unit queries (single-hop reasoning), collectively forming $Q = \{q, q_1, q_2, ...\}$ (Appendix B.2 for prompts). For example, the q *"What is the currency in the governmental jurisdiction with office holder Astrid Fischel Volio?"* is decomposed into a compound query q_1 *"What is the currency in Astrid Fischel Volio's jurisdiction?"* and unit queries like q_2 *"What areas does Astrid Fischel Volio oversee?"* and q_3 *"Who is Astrid Fischel Volio?"*.

Graph Parsing: LLM transforms triple t into graph query q^t (all triples' graph queries q^t collectively form the set Q^G) to construct quadruples $G^4 = (q^t, e_s, r, e_o)$, preserving the original knowledge graph while introducing graph queries for the subsequent alignment of graphs and query. For instance, the triple *(Costa Rica, monetary value, Costa Rican colón)* is converted into the graph query: *"What's the monetary value in Costa Rica?"* to construct (*"What's the monetary value in Costa Rica?", Costa Rica, monetary value, Costa Rican colón*). q^t and q_i ($q_i \in Q$) are in the same query format to facilitate subsequent alignment.

3.2.2 Focus-Aware Multi-Channel Pruning

To address the focus mismatch between q and G, we adopt pruning techiques to remove noisy focuses in G, inspired by previous approaches (HOMELS (Panda et al., 2024), QA-GNN (Yasunaga et al., 2021), and DynaGraph (Guan et al., 2022)). However, these naive pruning methods only focus q and G at a global level, simply pruning based on query-triple similarity, while ignoring local focuses, which often introduces noise.

To solve this, we propose **Focus-Aware Multi-Channel Pruning** that considers both global and local focuses. The Q from 3.2.1 contains global and local focuses: compound queries (requiring multi-hop reasoning) contain more global information, representing the query's global focuses, while unit queries only require single-hop reasoning, representing the query's local focuses.

For each triple $t = (e_s, r, e_o)$ in G, we mask the head entity ($MASK_1(t) = (?, r, e_o)$), tail entity ($MASK_2(t) = (e_o, r, ?)$), and both ($MASK_3(t) = (?, r, ?)$) to create three recall channels, capturing various local focuses.

$$G_p = Top\ K \left(\sum_j \sum_i sim\left(q_i, MASK_j(t)\right) \right), \tag{288}$$

where $q_i \in Q, t \in G$. To fully focus the semantics between q and G from both global and local perspectives during pruning, we compute semantic similarity between Q and each triple of G in three recall channels. For example, in the $MASK_1$ channel, we calculate the dot-product similarity between all queries in Q and each masked triple $(?, r, e_o)$, summing the scores. The final score of a triple combines scores across all recall channels. We keep the top K triples t_p with the highest final scores as the pruned graph G_p (Appendix D details ablation studies on K).

3.2.3 Structure-Driven Knowledge Enriching

To address the **structure mismatch** caused by **structure inconsistency in reasoning hops** and **lacking hierarchy structure** (see 2.2), we propose **Structure-Driven Knowledge Enriching** to optimize the structures of G_p. First, to efficiently align the structures of q and G_p, we use the LLM to filter the relevant q^{t_p} from G^4 for each q_i in Q (obtained in 3.2.1), and then integrate q_i into the quadruples $G^4 = (q_i \cup q^{t_p}, e_s, r, e_o)$. Since both are parsed into the same query format, LLM can easily handle this integration.

Inspired by previous work (Han et al., 2024; McGuinness et al., 2004), KGs have two key properties: structural attributes (similarity, symmetry, and transitivity of entity relationships) and feature attributes (hierarchical ontologies). Based on these, we improve the reasoning paths in G_p (both single-hop and multi-hop structures) using structural properties and enrich entity hierarchy structures in G_p with feature properties to mitigate ambiguity.

Specifically, we use Q as indicators to enrich graph structures (e_s, r, e_o) in G^4. For single-hop structures, **similarity** and **symmetry** complement semantics. For example, for quadruples *("What areas does Astrid Fischel Volio govern or oversee?", Astrid Fischel Volio, Managed by, m.010ggwdg)*, LLM generates the reverse relation *Politician's roles* for *Astrid Fischel Volio*, aligning better with q. For multi-hop structures, **transitivity** reduces reasoning complexity. The 3-hop structure from *Astrid Fischel Volio* to *Costa Rica* is simplified to a direct relation through *Position jurisdiction*, aligning more closely with q and streamlining reasoning. Lastly, **hierarchy** enriches hierarchy structures of entities e_s and e_o by generating query-related ontology, reducing ambiguity and aiding precise reasoning. For example, for the entity *Astrid Fischel Volio*, its ontology triple *(Astrid Fischel Volio, is-a, Politician)* clarifies identity. The enriched graph is then used to assist LLM in question answering efficiently. (How EoG's aligned graphs help LLMs reason efficiently is detailed in Appendix E. Enriching and Question Answering prompts are in the Appendix B.3, B.4, B.5.)

3.3 Graph Quality Evaluation Metrics

To verify that our proposed method facilitates progress toward the optimization objective, we design graph evaluation metrics and theoretically prove their correlation with the optimization goal.

Relevance: As discussed in 3.2.3, KGs have structural and feature-based properties. For structural property, we aim to eliminate noisy focuses and address structure inconsistency in reasoning hops between the q and G. To measure relevance, we define a metric where v_q and v_t are the embeddings of query q and triple t, respectively.

$$S^r(q, G) = \sum_{t \in G} sim\left(v_q, v_t\right)$$

Semantic Richness: For the feature property of KG, we need to enirch the hierachy structures of entities in G to address the ambiguity issues, providing more semantically enriched ontological information. Therefore, we have designed a semantic richness metric:

$$S^e(G) = \sum_{t \in G} KGC\left(t_+\right),$$

where t_+ indicates positive triples, KGC is the semantic scoring model, such as KG-BERT (Yao et al., 2019), which evaluates the completeness score of the triples in the semantic space.

Redundancy: For the KGQA task, redundant and repetitive triples are meaningless, as they do not provide additional effective information for LLM reasoning but instead increase computational overhead (Liao et al., 2025; Liu et al., 2024; Yuan et al., 2024). Therefore, we design a redundancy metric.

$$S^d(G) = \sum_{G^{sub} \in G} \sum_{r_{j_1} \in G^{sub}(r)} \sum_{r_{j_2} \in G^{sub}(r)} sim\left(v_{r_{j_1}}, v_{r_{j_2}}\right),$$

$$j_1 \neq j_2, G^{sub} = \left\{(e_s, r, e_o) \middle| e_s = e_s^{fixed}, e_o = e_o^{fixed}\right\}$$

Here, $G(r)$ extracts the relationship in the triples, e_s^{fixed} and e_o^{fixed} denote subgraphs consisting of triples with the same head entity and tail entity, respectively.

Theorem 2 *The relationship between the graph quality evaluation metrics Relevance and Semantic Richness are positively correlated with the optimization objective $MI(Q, G)$.*

Model	CWQ		WebQSP	
	Hit@1	*F1*	*Hit@1*	*F1*
Information Retrieval				
KV-Mem	21.1	15.7	46.7	34.5
GraftNet	36.8	32.7	66.4	60.4
PullNet	47.2	-	68.1	-
EmbedKGQA	44.7	-	66.6	-
NSM+h	48.8	44.0	74.3	67.4
TransferNet	48.6	-	71.4	-
Subgraph Retrieval	50.2	47.1	69.5	64.1
Semantic Parsing				
SPARQL	31.6	-	-	-
UHop	-	29.8	-	68.5
Topic Units	39.3	36.5	68.2	67.9
QGG	44.1	40.4	73.0	74.0
UniKGQA	51.2	49.4	77.2	72.2
TIARA	-	-	75.2	-
LLMs				
Flan-T5-xl	14.7	-	31.0	-
Alpaca-7B	27.4	-	51.8	-
LLaMa2-Chat-7B	34.6	-	64.4	-
IO prompt w/ ChatGPT	37.6	-	63.3	-
CoT prompt w/ ChatGPT	38.8	-	62.2	-
SC prompt w/ ChatGPT	45.4	-	61.1	-
InteractiveKBQA	59.2	-	72.5	-
LLMs+KGs				
StructGPT	54.3	-	72.6	-
KD-CoT	50.5	-	73.7	50.2
DeCAF	70.4	-	82.1	-
KG-CoT	62.3	-	84.9	-
RoG	62.6	56.2	85.7	70.8
ToG	69.5	-	82.6	-
DoG	58.2	-	**91.0**	-
EoG (Ours)	**70.8**	**65.1**	85.0	**74.1**

Table 1: Performance comparison of EoG and various baselines on CWQ and WebQSP datasets, with the best results in bold.

$$P(q,G)=\frac{\sum_{j=1}^{l} n\left(q,t_j\right)}{l}\propto\frac{\sum_{j=1}^{l} sim\left(v_q,v_{t_j}\right)}{l}$$

$$P(q,G)=\frac{\sum_{j=1}^{l} n\left(q,t_j\right)}{l}\propto\frac{\sum_{t_+\in G} n\left(q,t_+\right)}{l}\propto\frac{\sum_{t_+\in G} KGC\left(t_+\right)}{l}$$

Therefore, we can obtain:

$$MI(q,G)\ \propto S^r,S^e$$

We prove that maximizing $MI(q,G)$ using LLM prior knowledge aligns the semantics of q and G, improving graph quality metrics. Experiments further validate that EoG generates higher-quality graphs, contributing to the achievement of the optimization goal.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate reasoning performance on benchmark KGQA datasets used by: WebQSP (Yih et al., 2016) includes 4,737 questions involving simple and two-hop reasoning, while CWQ (Talmor and Berant, 2018) features 34,689 questions requiring complex 2-4 hop reasoning. Both datasets use Freebase (Bollacker et al., 2008b) as the KG, with 88 million entities, 20,000 relations, and 126 million triples. More details are in the Appendix A.

Model	CWQ				WebQSP			
	Acc.	*Hit@1*	*F1*	*Prec.*	*Acc.*	*Hit@1*	*F1*	*Prec.*
w/o $\mathbb{E}_s$	62.5	68.3	61.5	66.6	71.8	84.1	72.5	86.3
w/o $\mathbb{E}_f$	64.8	70.2	64.2	69.2	72.5	84.5	73.7	87.6
w/o $\mathbb{E}_o$	52.4	58.8	50.9	56.1	66.2	81.4	67.2	83.0
w/o $\mathbb{P}$ & $\mathbb{E}_o$	49.8	55.6	47.9	52.1	64.1	79.8	64.1	77.7
EoG	**65.5**	**70.8**	**65.1**	**70.4**	**73.0**	**85.0**	**74.1**	**95.6**

Table 2: Ablation study of EoG's core modules, comparing the results after removing the Enrich and Prune modules separately.

Baselines. We compared KGQA baselines from Section 5: information retrieval, semantic parsing, LLM reasoning with and without KGs.

Evaluation Metrics. Following previous work, we use Hits@1 for top-answer accuracy and F1 to assess coverage for multi-answer questions, balancing Precision and Recall.

Implementations. During the pruning stage, we use sentence-transformers (Reimers and Gurevych, 2019) as the retriever, setting the top $k = 300$. GPT-4o mini serves as our base model, with temperature set to 0.2, generation count n and sampling parameter top p set to 1, and max tokens set to the model's maximum output length for reproducibility.

4.2 Main Result

Tab. 1 compares the performance of our proposed EoG method with existing SOTA approaches. The LLM for KG method achieves superior performance compared to other mainstream models. Our approach outperforms other LLM-based KGQA methods across nearly all metrics on the CWQ and WebQSP datasets. On WebQSP, we improve F1 by 4.7% over the advanced RoG. On the challenging CWQ dataset, our method demonstrates overall superiority, with Hits@1 and F1 improved by 13.1% and 15.8%, respectively, compared to RoG, and Hits@1 improved by 4.7% and 0.6% compared to ToG and DeCAF. These results highlight the SOTA performance of our method on KGQA tasks.

4.3 Abalation Study

We conducted ablation studies to assess the effectiveness of the Prune and Enrich modules, summarized in Tab. 2. The Enrich module includes Structural and Feature submodules, examined under four settings: (1) w/o Prune ($\mathbb{P}$), (2) w/o Enrich ($\mathbb{E}_o$), (3) w/o Feature Enrich ($\mathbb{E}_f$), and (4) w/o Structural Enrich ($\mathbb{E}_s$). **Alleviating focus mismatches:** The Prune module improves all performance metrics by reducing significant noise in vanilla KG. **Mitigating structure mismatches:** Using Structural Enrich alone increases Hits@1 and F1 by 3.8% and 9.7% on WebQSP, and 19.4% and 26.1% on CWQ. Feature Enrich alone boosts Hits@1 and F1 by 3.3% and 7.9% on WebQSP, and 16.2% and 21% on CWQ. Combining both submodules achieves even better results than using either individually. These findings demonstrate the effectiveness of the Prune and Enrich modules in query-graph alignment.

4.4 Graph Evaluation Metrics

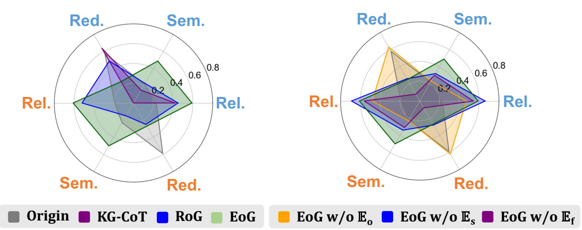

Figure 4: **Left:** Comparison of graph quality metrics between EoG and other methods. **Right:** Validation of graph quality improvement by Prune and Enrich modules. **Rel.**, **Sem.**, and **Red.** indicate Relevance, Semantic richness, and Redundancy, respectively, and these metrics in the blue and orange respectively represent the results on the CWQ and WebQSP.

To showcase the advantages of EoG-generated graphs, we evaluated their quality using three metrics: relevance, semantic richness, and redundancy. On the WebQSP and CWQ datasets, EoG was compared with the original dataset, RoG, and KG-CoT. KG-CoT results are only reported for CWQ due to incompatibility with WebQSP (Fig. 4 left). EoG achieves relavance scores above 0.6, outperforming RoG and KG-CoT with less noise. Its semantic richness (~0.5) is 20% higher, enhancing semantics for better LLM reasoning. Furthermore, EoG has the lowest redundancy (~0.2), ensuring concise graphs with minimal redundant information. We also analyzed the effects of EoG's Prune and Enrich modules (Fig. 4 right). The results show that the Prune improves the relevance score by over 0.25. Structural Enrich and Feature Enrich each increase the semantic richness score by over 0.1, while their combination improves it by over 0.26. Feature Enrich significantly reduces redundancy to below 0.1. This demonstrates the contribution of Prune and Enrich in improving subgraph quality and aligning KGs with the query.

Model	CWQ			WebQSP		
	# LLM Call	*Total Token*	*Total Cost*	*# LLM Call*	*Total Token*	*Total Cost*
CoT	1	409.7	8.00E-05	1	397.6	8.00E-05
ToG	9.2	11468.5	2.30E-03	8.8	10189.4	2.10E-03
DoG	5.7	37919.7	6.00E-03	2.7	6114.5	1.00E-03
EoG	4	6213.6	1.10E-03	4	6802.1	1.20E-03
w/o $\mathbb{E}_o$	2	4610.0	7.00E-04	2	4841.6	8.00E-04
w/o $\mathbb{P}$ & $\mathbb{E}_o$	1	63814.4	9.00E-03	1	66928.0	1.00E-02

Table 3: Efficiency comparison between EoG and other advanced methods.

Additionally, we visualized random CWQ cases to illustrate EoG's graph quality advantages across metrics in Fig. 5. **Relevance:** Using t-SNE, we plotted triples and question embeddings in 2D, with yellow dots marking question positions. EoG triples are closer and more concentrated around these points compared to the dispersed, noisy clusters of other methods, confirming higher relevance and reduced noise. **Semantic Richness:** With t-SNE and KG-BERT semantic scoring, circle sizes represent triple semantic scores (larger circles denote higher scores). EoG covers a significantly larger total area, illustrating richer semantic information. **Redundancy:** Triple redundancy was visualized as a density distribution of pairwise Euclidean distances. EoG triples exhibit greater spacing, indicating lower redundancy and more concise graphs.

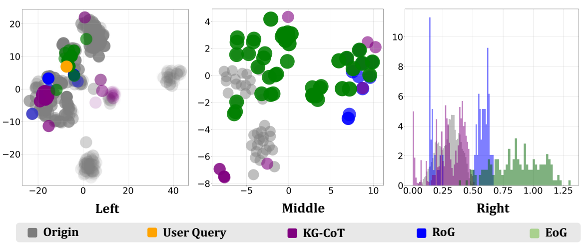

Figure 5: Graph visualization of EoG and advanced methods on the CWQ dataset. **Left:** t-SNE projection of **Relevance**. **Middle:** t-SNE visualization of **Semantic Richness**. **Right:** Comparison of **Redundancy**.

4.5 Computation Cost

We conducted a computation cost experiment to assess EoG's efficiency by comparing LLM call frequency, token usage, and cost metrics (Tab. 3).

Model	CWQ		WebQSP	
	Hit@1	*F1*	*Hit@1*	*F1*
RoG	62.2	55.4	86.4	70.8
RoG+$\mathbb{E}_o$	**75.4**$_{\uparrow 13.2}$	**68.7**$_{\uparrow 13.3}$	**91.5**$_{\uparrow 5.1}$	**77.1**$_{\uparrow 6.3}$
KG-CoT	62.3	52.7	-	-
KG-CoT+$\mathbb{E}_o$	**63.7**$_{\uparrow 1.4}$	**65.7**$_{\uparrow 13.0}$	-	-
ToG	67.6	-	82.6	-
ToG+$\mathbb{E}_o$	**70.7**$_{\uparrow 3.1}$	-	**89.4**$_{\uparrow 6.8}$	-
DoG	56.0	-	91.0	-
DoG+$\mathbb{E}_o$	**60.5**$_{\uparrow 4.5}$	-	**92.5**$_{\uparrow 1.5}$	-

Table 4: Impact of the Enrich module as a plugin enhancing other KGQA methods.

Results show EoG requires only 4 LLM calls per query, far fewer than ToG and DoG, and reduces token usage by 45.8% and 83.6% compared to ToG and DoG, respectively, with the lowest cost. Using the pruning module, EoG further cuts token usage by 92.8%, highlighting its superior computational efficiency.

4.6 Plug-and-Play

To validate EoG's plug-and-play capability, we applied the Enrich module to enhance other KGQA methods. As shown in Tab. 4, integrating Enrich with RoG, KG-CoT, ToG, and DoG significantly improved Hits@1 and F1, demonstrating its adaptability and robustness across KGQA methods. Additionally, experiments on base models with different reasoning capabilities and temperature parameters achieved excellent results, demonstrating EoG's strong reproducibility and plug-and-play flexibility (results in Appendix F).

5 Related Work

Traditional KGQA. Traditional KGQA methods are often categorized into information retrieval and semantic parsing approaches. Information retrieval methods (e.g., KV-Mem (Miller et al., 2016), GraftNet (Sun et al., 2018), PullNet (Sun et al., 2019), EmbedKGQA (Saxena et al., 2020), NSM+h (He et al., 2021), TransferNet (Shi et al., 2021), and Subgraph Retrieval (Zhang et al., 2022)) rely on retrieving query-relevant subgraphs from KGs but often retrieve irrelevant noises, leading to focus mismatch and reasoning errors. Semantic parsing (e.g., SPARQL (Sun et al., 2020), UHop (Chen et al., 2019), Topic Units (Lan et al., 2019), QGG (Lan and Jiang, 2020), UniKGQA (Jiang et al., 2023b), RnG-KBQA (Ye et al., 2022), and TIARA (Shu et al., 2022)) converts user queries into formal representations (e.g., SPARQL) to retrieve answers but struggles with complex queries, causing incomplete or failed answers.

LLMs for KGQA. LLMs introduce a transformative approach to KGQA, achieving state-of-the-art results through fine-tuned models and advanced prompting strategies. Earlier works leverage Flan-T5-xl (Chung et al., 2024), Alpaca-7B (Taori et al., 2023), and LLaMa2-Chat-7B (Touvron et al., 2023) with CoT (Chain-of-Thought) (Wei et al., 2022), ToT (Tree-of-Thought) (Yao et al., 2023a), and GoT (Graph-of-Thought) (Besta et al., 2024) to enhance reasoning reliability, showcasing strong semantic understanding and reasoning abilities. Advanced methods highlight these capabilities for KGQA but lead to resource-intensive workflows that lack scalability. StructGPT (Jiang et al., 2023a) unifies the discrepancies between queries and various data formats for collaborative reasoning. ReACT (Yao et al., 2023b), KD-CoT (Wang et al., 2023), DoG (Ma et al., 2025), and ToG (Sun et al., 2023) iteratively decompose query and perform step-by-step reasoning while focusing on single-hop subgraphs. However, these methods face challenges such as complex workflows with iterative interactions and increased expenses. KG-CoT (Zhao et al., 2024) and RoG (Luo et al., 2024) plan relational paths based on the query and extract relevant relational subgraphs from vanilla KGs as external knowledge to assist LLMs, but they are limited to specific KGs and incur high training costs.

Most LLM-based KGQA methods attempt to align query-graph semantics through reasoning, but still face a semantic gap with vanilla KGs, hindering reasoning.

6 Conclusion

In this paper, we propose EoG, which leverages LLMs as priors to generate query-aligned graphs for efficient reasoning. For KGQA tasks, we identify the key insight of the semantic gap between queries and graphs and tackle the limitations of vanilla KGs while avoiding the complex reasoning pipeline of existing methods. We also introduce three graph evaluation metrics with theoretical support. Extensive experiments show EoG achieves SOTA performance in KGQA while maintaining low computational costs, scalability, and adaptability across different methods.

Limitations

The limitations of our work at the current stage are mainly twofold:

(1) In the Feature Enrich part, we generate a general-domain ontology hierarchy. However, for tasks with varying ontology granularity across different domains, it may be necessary to design domain-specific ontology hierachy structures tailored to each domain.

(2) In our work, we use ChatGPT-series LLMs for query-graph alignment. Further exploration is needed to evaluate the alignment performance of LLMs with different parameter scales. In future work, we will investigate the performance of various LLMs at different parameter levels.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008a. Freebase: A collaboratively created graph database for structuring human knowledge. pages 1247–1250.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. 2008b. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*.

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.

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. UHop: An unrestricted-hop relation extraction framework for knowledge-based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 345–356, Minneapolis, Minnesota. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1).

Mingyu Guan, Anand Padmanabha Iyer, and Taesoo Kim. 2022. Dynagraph: dynamic graph neural networks at scale. In *Proceedings of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '22, New York, NY, USA. Association for Computing Machinery.

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

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 553–561, New York, NY, USA. Association for Computing Machinery.

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, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*.

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*.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974, Online. Association for Computational Linguistics.

Yunshi Lan, Shuohang Wang, and Jing Jiang. 2019. Knowledge base question answering with topic units. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5046–5052. International Joint Conferences on Artificial Intelligence Organization.

Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. 2024. Unigen: A unified generative framework for retrieval and question answering with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8688–8696.

Huanxuan Liao, Shizhu He, Yupu Hao, Xiang Li, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2025. SKIntern: Internalizing symbolic knowledge for distilling better CoT capabilities into small language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3203–3221, Abu Dhabi, UAE. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*.

Jie Ma, Zhitao Gao, Qi Chai, Wangchun Sun, Pinghui Wang, Hongbin Pei, Jing Tao, Lingyun Song, Jun Liu, Chen Zhang, and 1 others. 2025. Debate on graph: a flexible and reliable reasoning framework for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 24768–24776.

Deborah L McGuinness, Frank Van Harmelen, and 1 others. 2004. Owl web ontology language overview. *World Wide Web Consortium Recommendation*, 10(10):2004.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, and 1 others. 2024. Holmes: Hyper-relational knowledge graphs for multi-hop question answering using llms. *arXiv preprint arXiv:2406.06027*.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. 2022. TIARA: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8108–8121, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics.

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. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

Yawei Sun, Lingling Zhang, Gong Cheng, and Yuzhong Qu. 2020. Sparqa: skeleton-based semantic parsing for complex questions over knowledge bases. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8952–8959.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. *ArXiv*, abs/1803.06643.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*.

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.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

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*, pages 535–546, Online. Association for Computational Linguistics.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043, Dublin, Ireland. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Ye Yuan, Chengwu Liu, Jingyang Yuan, Gongbo Sun, Siqi Li, and Ming Zhang. 2024. A hybrid rag system with comprehensive enhancement on complex reasoning. *arXiv preprint arXiv:2408.05141*.

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)*, pages 5773–5784, Dublin, Ireland. Association for Computational Linguistics.

Kun Zhang, Jiali Zeng, Fandong Meng, Yuanzhuo Wang, Shiqi Sun, Long Bai, Huawei Shen, and Jie Zhou. 2024. Tree-of-reasoning question decomposition for complex question answering with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19560–19568. AAAI Press.

Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. 2024. Kg-cot: Chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering. pages 6642–6650.

Appendix

A Datasets Statistics

We use two KGQA benchmarks that leverage Freebase (Bollacker et al., 2008b) as the underlying knowledge graph: WebQuestionSP (Yih et al., 2016) and Complex WebQuestions (Talmor and Berant, 2018). To ensure fair comparison, the dataset preprocessing follows previous work (Luo et al., 2024). Dataset statistics are provided in Tab. 5, which shows the dataset splits as well as the number of reasoning hops required to answer the questions. WebQSP contains 2,826 and 1,628 questions for the training and test sets, respectively, while CWQ consists of 27,639 and 3,531 questions for the training and test sets, respectively. In WebQSP, the questions mainly involve 1-hop and 2-hop reasoning, with 2-hop questions accounting for 34.51%. CWQ includes questions requiring 1 to 4 hops, with those involving more than 2 hops making up 59.09%. As shown in Tab. 6, both datasets contain multi-answer samples, with 48.8% of WebQSP questions and 29.4% of CWQ questions having two or more answers.

Datasets	KB	#Train	#Test	1 hop	2 hop	$\geq$ 3 hop	Max #hop
WebQSP	Freebase	2826	1628	65.49%	34.51%	0.00%	2
CWQ	Freebase	27639	3531	40.91%	38.34%	20.75%	4

Table 5: Datasets statistics of WebQSP and CWQ

Datasets	#Ans=1	$2 \geq$ #Ans $\leq$	$5 \geq$ #Ans ≤ 9	#Ans ≥ 10
WebQSP	51.20%	27.40%	8.30%	12.10%
CWQ	70.60%	19.40%	6%	4%

Table 6: Answer distribution proportion of WebQSP and CWQ

B Prompts

Listing B.1, B.2, B.3, B.4, B.5 detail the specific designs of the Chain of Thought prompt, Query Structuring prompt, Structural Enrich prompt, Feature Enrich prompt, and Question Answering prompt. For the CoT prompt, we directly adopt the design from (Sun et al., 2023). Due to space constraints, some demonstrations for the Structural Enrich prompt and Feature Enrich prompt are omitted.

B.1 Chain of Thought Prompt

```
Q: What state is home to the university that is
represented in sports by George Washington Colonials
 men's basketball?\nA: First, the education
institution has a sports team named George
Washington Colonials men's basketball in is George
Washington University , Second, George Washington
University is in Washington D.C. The answer is {
Washington, D.C.}.

Q: Who lists Pramatha Chaudhuri as an influence and
wrote Jana Gana Mana?
A: First, Bharoto Bhagyo Bidhata wrote Jana Gana
Mana. Second, Bharoto Bhagyo Bidhata lists Pramatha
Chaudhuri as an influence. The answer is {Bharoto
Bhagyo Bidhata}.

Q: Who was the artist nominated for an award for You
 Drive Me Crazy?
A: First, the artist nominated for an award for You
Drive Me Crazy is Britney Spears. The answer is {
Jason Allen Alexander}.

Q: What person born in Siegen influenced the work of
 Vincent Van Gogh?
A: First, Peter Paul Rubens, Claude Monet and etc.
influenced the work of Vincent Van Gogh. Second,
Peter Paul Rubens born in Siegen. The answer is {
Peter Paul Rubens}.

Q: What is the country close to Russia where Mikheil
 Saakashvii holds a government position?
A: First, China, Norway, Finland, Estonia and
Georgia is close to Russia. Second, Mikheil
Saakashvii holds a government position at Georgia.
The answer is {Georgia}.

Q: What drug did the actor who portrayed the
character Urethane Wheels Guy overdosed on?
A: First, Mitchell Lee Hedberg portrayed character
Urethane Wheels Guy. Second, Mitchell Lee Hedberg
overdose Heroin. The answer is {Heroin}.

Q: {question}
```

Listing 1: Demonstration of CoT Prompt

B.2 Query Structuring Prompt

```
[INST] <<SYS>>
<</SYS>>
{instruction}
Given a question, decompose it step by step into
smaller components until it is broken down into unit
 queries that can be directly answered without
further reasoning or decomposition. A unit query is
a question that cannot be divided further and can be
 answered directly. If the given question is already
 a unit query, no decomposition is needed(If a
question can still be further broken down, it must
be divided into at least $two or more sub-questions$
. Otherwise, the question is considered a unit query
.). Your output should only include the tree
structure of the decomposed question, with sub-
questions indented using "-", and no additional
content should be provided, just the tree structure.
 The format for the decomposition tree is as follows
:
### decomposition tree format ###
question
-sub-question
--sub-sub-question
---sub-sub-sub-question
---sub-sub-sub-question
--sub-sub-question
-sub-question
### decomposition tree format ###
I will provide examples, please complete your task
after reviewing them.
{/instruction}
{demonstrations}
### Example 1:
Input:
What is the name of the scientist who developed the
theory that explains why objects fall to Earth?
Output:
What is the name of the scientist who developed the
theory that explains why objects fall to Earth?
```

```
-What is the theory that explains why objects fall
to Earth?
--Is there a theory for why objects fall to Earth?
--What is the name of this theory?
-Who developed this theory?
--Is this theory associated with a specific
scientist?
--What is the name of this scientist?
### Example 2:
Input:
What type of energy powers the device invented by
Thomas Edison that produces light?
Output:
What type of energy powers the device invented by
Thomas Edison that produces light?
-What device produces light and was invented by
Thomas Edison?
--Who is Thomas Edison?
--What devices did Thomas Edison invent?
--Is there a device invented by Thomas Edison that
produces light?
--What is the name of this device?
-What type of energy powers this device?
--What is energy in this context?
--What is the primary mechanism or process that
allows this device to produce light?
--What type of energy drives this mechanism?
### Example 3:
Input:
What inspired the author of the book "1984" to write
 it?
Output:
What inspired the author of the book "1984" to write
 it?
-Who is the author of the book "1984"?
--What is the book "1984"?
--Who wrote the book "1984"?
-What inspired this author to write "1984"?
--What was happening during the time this author
wrote "1984"?
--What personal experiences influenced the author?
--What political or social events might have
inspired the author?
--Did any other books or ideas inspire the author?
### Example 4:
Input:
What is the atomic number of the element discovered
in the laboratory where the youngest Nobel Physics
laureate worked?
Output:
What is the atomic number of the element discovered
in the laboratory where the youngest Nobel Physics
laureate worked?
-Who is the youngest Nobel Physics laureate?
--When did this person win the Nobel Prize?
--What was their age at the time of the award?
-Where did this laureate work?
--Did this laureate work in laboratory/institution?
--what's the name of the laboratory/institution?
-Was any chemical element discovered at this
laboratory?
--What is the name of the element?
--What is its atomic number?
### Example 5:
Input:
Which country hosted the sporting event where the
first female gold medalist in track and field
competed?
Output:
Which country hosted the sporting event where the
first female gold medalist in track and field
competed?
-What type of sporting event is being referred to (e
.g., Olympics)?
-Who was the first female gold medalist in track and
 field?
--What is track and field?
--Which female athlete won the first gold medal in
this category?
-In which event within track and field did she win?
-Which country hosted this sporting event?
{/demonstrations}
### Your Turn
Input:
{question}
[/INST]
```

Listing 2: Demonstration of Query Structuring Prompt

B.3 Structural Enrich Prompt

```
[INST] <<SYS>>You are a linguist capable of
understanding user queries and the semantic
information contained in triples. You can enhance
the semantic information of triples based on the
principles of similarity, symmetry, and transitivity
, bridging the semantic gap between the triples and
the user queries. Your goal is to augment the
triples semantically so they can better address the
 u s e r s  queries.
<</SYS>>
{instruction}
You are given a subgraph that consists of multiple
triples, where each triple corresponds to one or
more user queries. The format is as follows:
triple(entity1, relation, entity2) - query1 - query2
 - query3 - ...
Here, a query represents a question from the user.
The triple associated with a query may contain
information relevant to answering the question.
However, the semantic connections both within and
between triples are typically weak, lacking dense
links to clearly establish relationships between
entities. As a result, answers to the queries are
often implicit and require complex reasoning. To
bridge the semantic gap between the queries and the
triples, please perform semantic enhancement on the
subgraph by considering the user queries.
Specifically, you will utilize the following
properties for semantic enhancement: Similarity,
Symmetry, and Transitivity. Below, you will find
detailed explanations of these properties along with
 examples. Please keep these principles in mind as
you perform semantic enhancement.
{Similarity}
The property of similarity indicates that for a
given triple (e1, r1, e2), we can find an
alternative relationship r2 that is semantically
different from r1 but shares the same directional
connection between e1 and e2. In other words, the
entities e1 and e2 remain connected in the same
direction, but the meaning of the connection changes
 to reflect the semantics of r2. For example:
If (e1, r1, e2) exists, then (e1, r2, e2) may also
exist, where r1     r2, and r1 and r2 represent
different relationships.
{/Similarity}
{Symmetry}
The property of symmetry indicates that for a given
triple (e1, r1, e2), there exists a new relationship
 r2 that is semantically different from r1 and
reverses the direction of the connection between e1
and e2. In other words, the new triple swaps the
roles of e1 and e2 while preserving their semantic
relationship in a reversed form. For example:
If (e1, r1, e2) exists, then (e2, r2, e1) may also
exist, where r1     r2, and r1 and r2 represent
different relationships.
{/Symmetry}
{Transitivity}
The property of transitivity indicates that when two
 triples share a common entity, we can combine their
 relationships into a new relationship that connects
 the remaining two entities directly. Specifically,
if (e1, r1, e2) and (e2, r2, e3) exist, then there
exists a new relationship r3 that semantically
encompasses both r1 and r2, forming a new triple (e1
, r3, e3). The new relationship r3 captures the
combined semantics of r1 and r2. For example:
Given (e1, r1, e2) and (e2, r2, e3), we derive (e1,
r3, e3), where r3 combines the meanings of r1 and r2
.
{/Transitivity}
Your task will follow the four steps below:
{Procedure}
Step 1:
Examine the given triples and their associated
queries. Understand the information contained within
 the triples and the questions being asked in the
queries.
```

```
Step 2:
Using the similarity and symmetry properties,
perform semantic enhancement on the given 1-hop
subgraph (i.e., triples where connections involve
only directly linked entities).
Step 3:
Using the transitivity property, perform semantic
enhancement on the given multi-hop subgraph (i.e.,
triples where connections involve intermediary
entities).
Step 4:
Output the newly generated triples resulting from
the semantic enhancement process.
{/Procedure}
I will provide examples, please complete your task
after reviewing them.
{/instruction}
{demonstrations}
### Example 1:
Input:
(Michelle Bachelet,people.person.nationality,Chile)-
What is the location that appointed Michelle
Bachelet to a governmental position?-Who is Michelle
 Bachelet?-What governmental position was Michelle
Bachelet appointed to?-Where was Michelle Bachelet
appointed to this position?
(Chile,language.human_language.countries_spoken_in,
Spanish Language)-What language is spoken in this
location?
1-hop:
(Michelle Bachelet,people.person.nationality,Chile)
2-hop:
(Michelle Bachelet,people.person.nationality,Chile)
->(Chile,language.human_language.countries_spoken_in
,Spanish Language)
Output:
{thought}
step 1:
To prepare for the subsequent semantic enhancement
of the triples and bridge the semantic gap between
the triples and the user queries, let me review the
triples along with their corresponding user query(
ies):
(Michelle Bachelet,people.person.nationality,Chile)-
What is the location that appointed Michelle
Bachelet to a governmental position?-Who is Michelle
 Bachelet?-What governmental position was Michelle
Bachelet appointed to?-Where was Michelle Bachelet
appointed to this position?
(Chile,language.human_language.countries_spoken_in,
Spanish Language)-What language is spoken in this
location?
step 2:
Using the similarity property, perform semantic
enhancement on the given 1-hop subgraph:
For the 1-hop subgraph (Michelle Bachelet,people.
person.nationality,Chile),the related queries are "
What is the location that appointed Michelle
Bachelet to a governmental position?" , "Who is
Michelle Bachelet?" , "What governmental position
was Michelle Bachelet appointed to?" , "Where was
Michelle Bachelet appointed to this position?" .
Combining these queries, The newly added  triple(s)
is/are (Michelle Bachelet,
appointed_government_position_in,Chile)
Using the symmetry properties, perform semantic
enhancement on the given 1-hop subgraph:
For the 1-hop subgraph (Michelle Bachelet,people.
person.nationality,Chile),the related queries are "
What is the location that appointed Michelle
Bachelet to a governmental position?" , "Who is
Michelle Bachelet?" , "What governmental position
was Michelle Bachelet appointed to?" , "Where was
Michelle Bachelet appointed to this position?" .
Combining these queries, The newly added  triple(s)
is/are (Chile, appointed_as_government_official_by,
Michelle Bachelet)
step 3:
Using the transitivity properties, perform semantic
enhancement on the given 2-hop subgraph:
For the 2-hop subgraph (Michelle Bachelet,people.
person.nationality,Chile)->(Chile,language.
human_language.countries_spoken_in,Spanish Language)
,the related queries are "What is the location that
appointed Michelle Bachelet to a governmental
position?" , "Who is Michelle Bachelet?" , "What
governmental position was Michelle Bachelet
appointed to?" , "Where was Michelle Bachelet
appointed to this position?" , What language is
spoken in this location?. Combining these queries,
The newly added  triple(s) is/are (Michelle Bachelet
, language_of_the_country_where_appointed ,Spanish
Language)
step 4:
Final output:
(Michelle Bachelet,appointed_government_position_in,
Chile)
(Chile, appointed_as_government_official_by,
Michelle Bachelet)
(Michelle Bachelet,
language_of_the_country_where_appointed ,Spanish
Language)
{/thought}
### Example 2:
Input:
(Lou Seal,sports.mascot.team,San Francisco Giants)-
What team has Lou Seal as its mascot?-Which team is
associated with Lou Seal as its mascot?
(San Francisco Giants,sports.
sports_championship_event.champion,2014 World Series
)-When did this team last win the World Series?-What
 year or time frame did this team last win the World
 Series?
(San Francisco Giants,sports.sports_team.
championships,2012 World Series)-When did this team
last win the World Series?-What year or time frame
did this team last win the World Series?
1-hop:
(Lou Seal,sports.mascot.team,San Francisco Giants)
2-hop:
(Lou Seal,sports.mascot.team,San Francisco Giants)
->(San Francisco Giants,sports.
sports_championship_event.champion,2014 World Series
)
(Lou Seal,sports.mascot.team,San Francisco Giants)
->(San Francisco Giants,sports.sports_team.
championships,2012 World Series)
Output:
{thought}
step 1:
To prepare for the subsequent semantic enhancement
of the triples and bridge the semantic gap between
the triples and the user queries, let me review the
triples along with their corresponding user query(
ies):
(Lou Seal,sports.mascot.team,San Francisco Giants)-
What team has Lou Seal as its mascot?-Which team is
associated with Lou Seal as its mascot?
(San Francisco Giants,sports.
sports_championship_event.champion,2014 World Series
)-When did this team last win the World Series?-What
 year or time frame did this team last win the World
 Series?
(San Francisco Giants,sports.sports_team.
championships,2012 World Series)-When did this team
last win the World Series?-What year or time frame
did this team last win the World Series?
step 2:
Using the similarity property, perform semantic
enhancement on the given 1-hop subgraph:
For the 1-hop subgraph (Lou Seal,sports.mascot.team,
San Francisco Giants),the related queries are "What
team has Lou Seal as its mascot?" , "Which team is
associated with Lou Seal as its mascot?" . Combining
 these queries, The newly added  triple(s) is/are (
Lou Seal,associated_with_team,San Francisco Giants)
Using the symmetry properties, perform semantic
enhancement on the given 1-hop subgraph:
For the 1-hop subgraph (Lou Seal,sports.mascot.team,
San Francisco Giants),the related queries are "What
team has Lou Seal as its mascot?" , "Which team is
associated with Lou Seal as its mascot?" . Combining
 these queries, The newly added  triple(s) is/are (
San Francisco Giants,has_mascot,Lou Seal)
step 3:
Using the transitivity properties, perform semantic
enhancement on the given 2-hop subgraph:
For the 2-hop subgraph (Lou Seal,sports.mascot.team,
San Francisco Giants)->(San Francisco Giants,sports.
sports_championship_event.champion,2014 World Series
),the related queries are "What team has Lou Seal as
 its mascot?" and "Which team is associated with Lou
 Seal as its mascot?" , "When did this team last win
 the World Series?" , "What year or time frame did
```

this team last win the World Series?" . Combining these queries, The newly added triple(s) is/are (Lou Seal,mascot_of_team_that_last_won,2014 World Series)
For the 2-hop subgraph (Lou Seal,sports.mascot.team, San Francisco Giants)->(San Francisco Giants,sports.sports_team.championships,2012 World Series),the related queries are "What team has Lou Seal as its mascot?" and "Which team is associated with Lou Seal as its mascot?" , "When did this team last win the World Series?" , "What year or time frame did this team last win the World Series?" . Combining these queries, The newly added triple(s) is/are (Lou Seal ,mascot_of_team_that_won,2012 World Series)
step 4:
Final output:
(Lou Seal,associated_with_team,San Francisco Giants)
(San Francisco Giants,has_mascot,Lou Seal)
(Lou Seal,mascot_of_team_that_last_won,2014 World Series)
(Lou Seal,mascot_of_team_that_won,2012 World Series)
{/thought}
Example 3:
Input:
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)-Who was the champion of the 1931 World Series Championship?-Who won the World Series in 1931?
(St. Louis Cardinals, plays at, Busch Stadium)-Where does this champion team play?
(St. Louis Cardinals, has arena, Busch Stadium)-Where does this champion team play?-What is the name of the stadium associated with this team?
(St. Louis Cardinals, sports.sports_team.arena_stadium, Busch Stadium)-What is the name of the stadium associated with this team?
(St. Louis Cardinals, sports.sports_team.arena_stadium, Roger Dean Stadium)-Was this stadium their home stadium in 1931?
(St. Louis Cardinals, home ground of, Busch Stadium) -Was this stadium their home stadium in 1931?
(St. Louis Cardinals, home ground of, Roger Dean Stadium)-Was this stadium their home stadium in 1931?
1-hop:
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)
2-hop:
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)->(St. Louis Cardinals, plays at, Busch Stadium)
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)->(St. Louis Cardinals, has arena, Busch Stadium)
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)->(St. Louis Cardinals, sports .sports_team.arena_stadium, Busch Stadium)
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)->(St. Louis Cardinals, sports .sports_team.arena_stadium, Roger Dean Stadium)
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)->(St. Louis Cardinals, home ground of, Busch Stadium)
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)->(St. Louis Cardinals, home ground of, Roger Dean Stadium)
Output:
{thought}
step 1:
To prepare for the subsequent semantic enhancement of the triples and bridge the semantic gap between the triples and the user queries, let me review the triples along with their corresponding user query(ies):
(1931 World Series, sports.sports_team.championships , St. Louis Cardinals)-Who was the champion of the 1931 World Series Championship?-Who won the World Series in 1931?
(St. Louis Cardinals, plays at, Busch Stadium)-Where does this champion team play?
(St. Louis Cardinals, has arena, Busch Stadium)-Where does this champion team play?-What is the name of the stadium associated with this team?
(St. Louis Cardinals, sports.sports_team.arena_stadium, Busch Stadium)-What is the name of the stadium associated with this team?
(St. Louis Cardinals, sports.sports_team.arena_stadium, Roger Dean Stadium)-Was this stadium their home stadium in 1931?
(St. Louis Cardinals, home ground of, Busch Stadium) -Was this stadium their home stadium in 1931?
(St. Louis Cardinals, home ground of, Roger Dean Stadium)-Was this stadium their home stadium in 1931?
step 2:
Using the similarity property, perform semantic enhancement on the given 1-hop subgraph:
For the 1-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals),the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?".Combining these queries, The newly added triple(s) is/are (1931 World Series, champion_1931_World_Series, St. Louis Cardinals)
Using the symmetry properties, perform semantic enhancement on the given 1-hop subgraph:
For the 1-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals),the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?".Combining these queries, The newly added triple(s) is/are (St. Louis Cardinals, won_1931_World_Series_as_champion, 1931 World Series)
step 3:
Using the transitivity properties, perform semantic enhancement on the given 2-hop subgraph:
For the 2-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals)->(St . Louis Cardinals, plays at, Busch Stadium),the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?" and "Where does this champion team play?".Combining these queries, The newly added triple(s) is/are (1931 World Series, champion_team_plays_at, Busch Stadium)
For the 2-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals)->(St . Louis Cardinals, has arena, Busch Stadium),the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?" and "Where does this champion team play?" and "What is the name of the stadium associated with this team?".Combining these queries, The newly added triple(s) is/are (1931 World Series , stadium_related_to_champion_team, Busch Stadium)
For the 2-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals)->(St . Louis Cardinals, sports.sports_team.arena_stadium, Busch Stadium),the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?" and "What is the name of the stadium associated with this team ?".Combining these queries, The newly added triple(s) is/are (1931 World Series, stadium_related_to_champion_team, Busch Stadium)
For the 2-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals)->(St . Louis Cardinals, sports.sports_team.arena_stadium, Roger Dean Stadium),the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?" and "Was this stadium their home stadium in 1931?".Combining these queries, The newly added triple(s) is/are (1931 World Series, home_stadium_related_to_champion_team, Roger Dean Stadium)
For the 2-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals)->(St . Louis Cardinals, home ground of, Busch Stadium), the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?" and "Was this stadium their home stadium in 1931?".Combining these queries, The newly added triple(s) is/are (1931 World Series, home_stadium_related_to_champion_team, Busch Stadium)
For the 2-hop subgraph (1931 World Series, sports.sports_team.championships, St. Louis Cardinals)->(St . Louis Cardinals, home ground of, Roger Dean Stadium),the related queries are "Who was the champion of the 1931 World Series Championship?" and "Who won the World Series in 1931?" and "Was this

```
stadium their home stadium in 1931?".Combining these
 queries, The newly added triple(s) is/are (1931
World Series, home_stadium_related_to_champion_team,
 Roger Dean Stadium)
step 4:
Final output:
(1931 World Series, champion_1931_World_Series, St.
Louis Cardinals)
(St. Louis Cardinals,
won_1931_World_Series_as_champion, 1931 World Series
)
(1931 World Series, champion_team_plays_at, Busch
Stadium)
(1931 World Series, stadium_related_to_champion_team
, Busch Stadium)
(1931 World Series,
home_stadium_related_to_champion_team, Roger Dean
Stadium)
(1931 World Series,
home_stadium_related_to_champion_team, Busch Stadium
)
{/thought}
{/demonstrations}
### Your Turn
Input:
{quadruples}
1-hop:
{1-hop path}
2-hop:
{2-hop path}
[/INST]
```

Listing 3: Demonstration of Structural Enrich Prompt

B.4 Feature Enrich Prompt

```
[INST] <<SYS>>You are a linguist with extensive
expertise in ontology knowledge. You have the
ability to understand the context of entities,
including user queries and the semantic information
of triples. Based on the contextual information of
an entity, you can accurately and appropriately add
ontologies to the entity. Your goal is to bridge the
 semantic gap between entities and user queries by
adding relevant ontologies, thereby semantically
augmenting the entities to enable them to better
address the user's queries.
<</SYS>>
{instruction}
You are provided with an entity list. Each entity in
 the list is accompanied by its context information,
 which consists of two parts:
1.Relevant triples associated with the entity.
2.Relevant user queries associated with the entity.
The context format for each entity is as follows:
[$entity$ context]
relavent triple(s):triple1-triple2-triple3-...
relavent user query(ies):query1-query2-query3-...
[/$entity$ context]
Your task is to analyze the context information of
each entity, apply your expert knowledge, and assign
 an appropriate ontology to the entity. The added
ontology should:
1.Be semantically consistent with the entity's
context information.
2.Avoid any contradictions or irrelevance with the
given context.
3.Enhance the entity's ability to better answer the
user query(ies) in its context.
Store the ontology assignments in a set called {
result}. The {result} set is initially empty. Each
ontology assignment should follow the format:
{result}
(entity,ontology_relation,newly added ontology)
...
{/result}
You must choose an appropriate ontology relation
from the options provided below.
{ontology relation definition}
HYP: Hypernym relation
Refers to the relation where a broader or more
general concept includes or encompasses a more
specific or specialized concept.
Hypernym_isA
Is a type of...
Hypernym_locateAt
Is located at...
Hypernym_mannerOf
A is a specific implementation or way of B. Similar
to "isA," but used for verbs. For example, "auction"
 -> "sale"
IND: Induction relation
Refers to the relation between individual entities
and conceptual generalizations derived from a class
of entities with common characteristics.
Induction_belongTo
This relation is commonly used in SPG to describe
the classification relation from entity types to
concept types. For example, "company event" -> "
company event category".
INC: Inclusion relation
Expresses the relation between parts and the whole.
Inclusion_isPartOf
A is a part of B.
Inclusion_madeOf
A is made up of B. For example, "bottle" -> "plastic
".
Inclusion_derivedFrom
A is derived from or originated from B, used to
express composite concepts.
Inclusion_hasContext
A is a word used in the context of B, where B can be
 a subject area, technical field, or regional
dialect. For example, "astern" -> "ship".
{/ontology relation definition}
You can only choose an appropriate {
ontology_relation} from {Hypernym_isA,
Hypernym_locateAt, Hypernym_mannerOf,
Induction_belongTo, Inclusion_isPartOf,
Inclusion_madeOf, Inclusion_derivedFrom,
Inclusion_hasContext}.
Your task will follow the four steps below:
{Procedure}
step 1:
For the first $entity$ in the entity list, read its
context information carefully. Based on the provided
 context and your expert knowledge, answer the
following question:
Does this $entity$ already have an ontology assigned
?
If yes, proceed to Step 2.
If no, move to Step 3.
step 2:
Using the entity's context information and your
expert knowledge:
Identify a list of appropriate ontologies for the
entity.
For each ontology in this list, select the most
suitable ontology relation from the {
ontology_relation} options.
Format each assignment as (entity, ontology_relation
, newly added ontology) and add it to the {result}
set.
Remove this $entity$ and its corresponding context
information from the entity list.
Output the current {result} set in the following
format:
{result}
...
{/result}
Then move to Step 3.
step 3:
Check if the entity list is now empty:
If the list is empty, proceed to Step 4.
If the list is not empty, return to Step 1 and
repeat the process for the next $entity$.
step 4:
Once all entities have been processed, output the
final {result} set containing all (entity,
ontology_relation, newly added ontology) assignments
 in the following format:
{result}
(entity,ontology_relation,newly added ontology)
...
{/result}
{/Procedure}
I will provide examples, please complete your task
after reviewing them.
{/instruction}
{demonstrations}
### Example 1:
Input:
```

```
entity List:
[$Michelle Bachelet$ context]
relavent triple(s):(Michelle Bachelet,people.person.
nationality,Chile)
relavent user query(ies):What is the location that
appointed Michelle Bachelet to a governmental
position?-Who is Michelle Bachelet?-What
governmental position was Michelle Bachelet
appointed to?-Where was Michelle Bachelet appointed
to this position?
[/$Michelle Bachelet$ context]
[$Chile$ context]
relavent triple(s):(Michelle Bachelet,people.person.
nationality,Chile)-(Chile,language.human_language.
countries_spoken_in,Spanish Language)
relavent user query(ies):What is the location that
appointed Michelle Bachelet to a governmental
position?-Who is Michelle Bachelet?-What
governmental position was Michelle Bachelet
appointed to?-Where was Michelle Bachelet appointed
to this position?-What language is spoken in this
location?
[/$Chile$ context]
Output:
{result}
(Michelle Bachelet, Hypernym_isA, Political Figure)
(Michelle Bachelet, Hypernym_isA, President)
(Chile, Hypernym_isA, Country)
(Chile, Hypernym_locateAt, South America)
(Chile, Inclusion_hasContext, Spanish Language)
{/result}
### Example 2:
Input:
entity List:
[$Ovadia Yosef$ context]
relavent triple(s):(Ovadia Yosef, people.person.
religion, Judaism)-(Ovadia Yosef, people.person.
nationality, Israel)
relavent user query(ies):Who is Ovadia Yosef?-Did
Ovadia Yosef hold a leadership role in a religion?-
What religion is associated with Ovadia Yosef?-Is
there a predominant religion associated with Ovadia
Yosef?-What is the name of this religion?
[/$Ovadia Yosef$ context]
[$Judaism$ context]
relavent triple(s):(Ovadia Yosef, people.person.
religion, Judaism)
relavent user query(ies):Who is Ovadia Yosef?-Did
Ovadia Yosef hold a leadership role in a religion?-
What religion is associated with Ovadia Yosef?-Is
there a predominant religion associated with Ovadia
Yosef?-What is the name of this religion?
[/$Judaism$ context]
[$Israel$ context]
relavent triple(s):(Ovadia Yosef, people.person.
nationality, Israel)
relavent user query(ies):Who is Ovadia Yosef?
[/$Israel$ context]
Output:
{result}
(Ovadia Yosef, Hypernym_isA, Religious Leader)
(Ovadia Yosef, Induction_belongTo, Jewish Religious
Leaders)
(Judaism, Hypernym_isA, Religion)
(Judaism, Inclusion_hasContext, Jewish Culture)
(Israel, Hypernym_isA, Country)
(Israel, Hypernym_locateAt, Middle East)
{/result}
### Example 3:
Input:
entity List:
[$1931 World Series$ context]
relavent triple(s):(1931 World Series, sports.
sports_team.championships, St. Louis Cardinals)
relavent user query(ies):Who was the champion of the
 1931 World Series Championship?-Who won the World
Series in 1931?
[/$1931 World Series$ context]
[$St. Louis Cardinals$ context]
relavent triple(s):(1931 World Series, sports.
sports_team.championships, St. Louis Cardinals)-(St.
 Louis Cardinals, plays at, Busch Stadium)-(St.
Louis Cardinals, has arena, Busch Stadium)-(St.
Louis Cardinals, sports.sports_team.arena_stadium,
Busch Stadium)-(St. Louis Cardinals, sports.
sports_team.arena_stadium, Roger Dean Stadium)-(St.
Louis Cardinals, home ground of, Busch Stadium)-(St.
 Louis Cardinals, home ground of, Roger Dean Stadium
)
relavent user query(ies):Who was the champion of the
 1931 World Series Championship?-Who won the World
Series in 1931?-Where does this champion team play?-
Where does this champion team play?-What is the name
 of the stadium associated with this team?-What is
the name of the stadium associated with this team?-
Was this stadium their home stadium in 1931?-Was
this stadium their home stadium in 1931?-Was this
stadium their home stadium in 1931?
[/$St. Louis Cardinals$ context]
[$Busch Stadium$ context]
relavent triple(s):(St. Louis Cardinals, plays at,
Busch Stadium)-(St. Louis Cardinals, has arena,
Busch Stadium)-(St. Louis Cardinals, sports.
sports_team.arena_stadium, Busch Stadium)-(St. Louis
 Cardinals, home ground of, Busch Stadium)
relavent user query(ies):Where does this champion
team play?-Where does this champion team play?-What
is the name of the stadium associated with this team
?-Was this stadium their home stadium in 1931?
[/$Busch Stadium$ context]
[$Roger Dean Stadium$ context]
relavent triple(s):(St. Louis Cardinals, sports.
sports_team.arena_stadium, Roger Dean Stadium)-(St.
Louis Cardinals, home ground of, Roger Dean Stadium)
relavent user query(ies):Was this stadium their home
 stadium in 1931?-Was this stadium their home
stadium in 1931?
[/$Roger Dean Stadium$ context]
Output:
{result}
(1931 World Series, Hypernym_isA, Sports
Championship)
(1931 World Series, Induction_belongTo, Baseball
Championships)
(St. Louis Cardinals, Hypernym_isA, Baseball Team)
(St. Louis Cardinals, Hypernym_locateAt, United
States)
(St. Louis Cardinals, Inclusion_isPartOf, Major
League Baseball)
(Busch Stadium, Hypernym_isA, Sports Arena)
(Busch Stadium, Induction_belongTo, Baseball
Stadiums)
(Roger Dean Stadium, Hypernym_isA, Sports Arena)
(Roger Dean Stadium, Induction_belongTo, Baseball
Stadiums)
{/result}
{/demonstrations}
### Your Turn
Input:
entity List:
[$entity$ context]
relavent triple(s):
relavent user query(ies):
[/$entity$ context]
[/INST]
```

Listing 4: Demonstration of Feature Enrich Prompt

B.5 Question Answering Prompt

```
[INST] <<SYS>>
<</SYS>>
{instruction}
You are given an input question along with
additional information to assist in answering the
question. The additional information includes a set
of relevant triples. The format of a triple is as
follows:
(e1, r, e2)
Your task is to perform step-by-step reasoning based
 on the provided additional information to arrive at
 the answer. You should output your process of
thinking and reasoning then the final answer. Each
answer should be as close to a single entity as
possible, rather than a long sentence. Each answer
should not be in the form of an ID, such as: "m.0
n1v8cy", "m.0jx21d", or similar formats. If the
provided information is insufficient to infer the
correct answer, please use your internal knowledge
to generate a response. Please try to output all
possible answers you consider correct. If there is
only one answer, directly output that answer. If
```

```
there are multiple answers, separate them using <SEP
>.
### Input Format ###
Input:
question:
{input question}
information:
{triple1}
{triple2}
{triple3}
...
### Output Format ###
Output:
{thoughts & reason}
Your process of thinking and reasoning
...
{/thoughts & reason}
Final answer:
{answer}
{/instruction}
### Your Turn
Input:
question:
{question}
information:
{knowledge graph}
[/INST]
```

Listing 5: Demonstration of Question Answering Prompt

C Details of our Theoretical Proof

In this section, we provide a detailed derivation of the solution for EoG.

As stated in 3.1, our goal is to find an optimized graph G^* by maximizing the expected posterior probability:

$$G^* = \underset{G}{argmax}\, \mathbb{E}_{P(q,G)}\left[P\left(M_\theta, q|G\right)\right]$$

It is equivalent to maximizing the mutual information between q and G. The detailed derivation is as follows.

Since $P(q)$ and $P(G)$ are constants, we have:

$$P\left(M_\theta, q|G\right) \propto \log\left(\frac{P\left(M_\theta, q|G\right)}{P(q)}\right)$$

The equivalent transformation of the conditional probability is:

$$P\left(M_\theta, q|G\right) = \frac{P\left(M_\theta, G|q\right)P(q)}{P(G)}$$

We can obtain:

$$P\left(M_\theta, q|G\right) \propto \log\left(\frac{P\left(M_\theta, q|G\right)}{P(q)}\right) = \log\left(\frac{P\left(M_\theta, G|q\right)}{P(G)}\right)$$

We eliminate the variable M_θ using the marginalization formula:

$$\log\left(\frac{P\left(M_\theta, G|q\right)}{P(G)}\right) \propto \log\left(\frac{\int_{M_\theta} P\left(M_\theta, G|q\right) dM_\theta}{P(G)}\right)$$

We can obtain:

$$\log\left(\frac{\int_{M_\theta} P\left(M_\theta, G|q\right) dM_\theta}{P(G)}\right) = \log\left(\frac{P\left(G|q\right)}{P(G)}\right) = \log\left(\frac{P\left(q, G\right)}{P(q)P(G)}\right)$$

Thus, the expectation of the original posterior probability is equivalent to:

$$\mathbb{E}_{P(Q,G)}\left[P\left(M_\theta, q|G\right)\right] \propto \mathbb{E}_{P(q,G)}\left[\log\left(\frac{P\left(q, G\right)}{P(q)P(G)}\right)\right]$$

Finally we can get:

$$\mathbb{E}_{P(q,G)}\left[P\left(M_\theta, q|G\right)\right] \propto MI(q, G)$$

D Optimal k-value Exploration

To explore the optimal number of retrieved triples(k) as discussed in Section 3.2.2, and to demonstrate the superiority of our Structure-Aware Multi-Channel Pruning method compared to the vanilla pruning approach, we evaluated the impact of different numbers of retrieved triples on **Answer Coverage** and **Token Cost** (using the default GPT-4o-mini) on the WebQSP and CWQ datasets. For answer coverage, it is calculated as: # Answers in Pruned Graph / # Total Answers. For Token Cost, it is determined by converting triples directly into natural language text, e.g., converting (*Beijing, located in, China*) into *Beijing located in China*. We then compute the total token number of the converted text and multiply it by the OpenAI API price to get the token cost. The results are shown in Fig. 6 and Fig. 7, where the values represented by the green dashed lines indicate the answer coverage before pruning performed by (Luo et al., 2024) on the dataset. As the number of retrieved triples increases, the number of answers mistakenly removed by pruning decreases, but the token length grows, leading to a continuous increase in API invocation costs. When the number of retrieved triples is 300, the pruned graph obtained by Structure-Aware Multi-Channel Pruning has only a 5% reduction in answer coverage compared to the unpruned graph. Compared to the vanilla pruning method, our pruning strategy achieves approximately 7% higher answer coverage across all triple counts, which demonstrates the superiority of our approach.

Considering the trade-off between token cost and answer coverage, we chose $k = 300$ as the optimal number of retrieved triples.

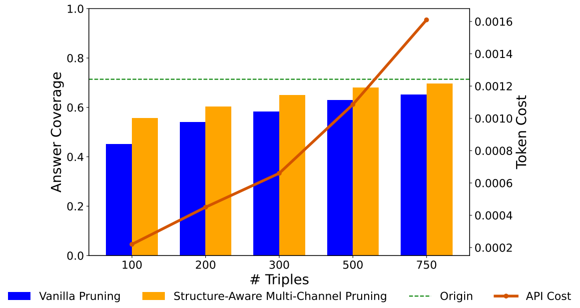

Figure 6: Impact of the Number of Retrieved Triples on Answer Coverage Ratio and API Call Cost ($) in the CWQ Dataset

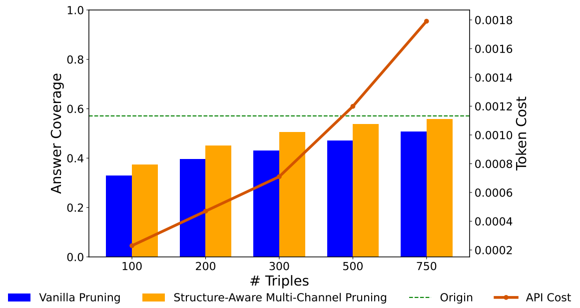

Figure 7: Impact of the Number of Retrieved Triples on Answer Coverage Ratio and API Call Cost ($) in the WebQSP Dataset

E Case Study

In this section, we present case study on the CWQ dataset to illustrate the critical role of EoG-generated query-aligned graphs in reasoning. We provide two examples, as shown in Fig. 8 and Fig. 9, comparing the performance of EoG with the vanilla KG method that assists ChatGPT in step-by-step reasoning. In the figures, green highlights the correct answers or key triples, red indicates misleading triples, and orange represents triples relevant to the query.

In the first example, the query is *What is the currency of the place whose religious organization leadership is the Society of Jesus?* The retrieved triples include misleading information: (*Pope Francis*, *people.person.birth*, *Italy*), (*Italy*, *location.country.currency_used*, *Euro*). Using the vanilla KG, ChatGPT retrieves the entity *m.0rgks1y* related to *Society of Jesus*, but since it cannot interpret the entity's semantics, it overlooks the correct triple: (*m.0rgks1y*, *religion.religious_organization_leadership.jurisdiction*, *Argentina*). Additionally, (*m.0rgks1y*, *religion.religious_organization_leadership.leader*, *Pope Francis*), (*Pope Francis*, *people.person.birth*, *Italy*), and (*Italy*, *location.country.currency_used*, *Euro*) lead ChatGPT to assume that Pope Francis, as a leader of the Society of Jesus, is the key to solving the query and conclude that the answer is Euro based on Pope Francis' birthplace, Italy. This error highlights the semantic gap between the query and the vanilla KG, resulting in incorrect reasoning. EoG bridges this semantic gap by leveraging structural and feature attributes like similarity, symmetry, transitivity, and hierarchy. Through symmetry, EoG generates the triple: (*m.0rgks1y*, *religion.religious_organization_leadership.organization*, *Society of Jesus*), indicating the leadership relationship between *Society of Jesus* and *m.0rgks1y*. Hierarchy generates: (*m.0rgks1y*, *Hypernym_isA*, *Religious Organization*), clarifying the meaning of *m.0rgks1y*. Transitivity further establishes the relationship: (*Society of Jesus*, *religious_leadership_of_place*, *Argentina*), which aligns semantically with the query, reduces reasoning hops, and enables LLMs to correctly infer that *Society of Jesus* is based in *Argentina*. Finally, the hierarchical triple: (*Argentina*, *Inclusion_hasContext*, *Argentine Peso*) helps the LLM deduce that the currency of *Argentina* is *Argentine Peso*. This proves that EoG's query-aligned graph effectively bridges the semantic gap, simplifies reasoning complexity, and enhances LLM efficiency.

In the second example, the vanilla KG contains ambiguous information: (*Eclipse, film.film.starring, m.075wxc2*), and incorrect triples: (*Eclipse, book.book.characters, Leah Clearwater*), (*New Moon, book.book.characters, Leah Clearwater*). These lead ChatGPT to assume that *Leah Clearwater's* presence in both *Eclipse* and *New Moon* implies a chronological relationship, resulting in the wrong answer: *New Moon*. In contrast, EoG's query-aligned graph generates: (*Eclipse*, *film.film.prequel*, *The Twilight Saga: New Moon*), which clarifies the prequel relationship between *Eclipse* and the correct answer, *The Twilight Saga: New Moon*. Additionally, the transitive triple: (*Eclipse*, *features_actor*, *Taylor Lautner*) helps LLMs verify the connection between *Taylor Lautner* and *Eclipse* from the query, leading to the correct reasoning and answer.

These examples demonstrate that EoG-generated query-aligned graphs not only bridge semantic gaps but also reduce reasoning complexity, enabling LLMs to perform more accurate and efficient reasoning.

Question	What is the currency of the place whose religious organization leadership is the Society of Jesus?
Answer	Argentine Peso
Vanilla Graph	("m.0rgks1y","religion.organization_conducting_ceremonies","Society of Jesus") ("m.0rgks1y","religion.religious_organization_leadership.jurisdiction","Argentina") **("m.0rgks1y","religion.religious_organization_leadership.leader","Pope Francis")** ("Pope Francis","people.person.religion","Catholicism") **("Pope Francis","people.person.birth","Italy")** **("Italy","location.country.currency_used","Euro")**
ChatGPT+CoT	The currency of the place where the religious organization leadership (Society of Jesus) is located is Euro. Here's the reasoning: The religious organization (associated with "m.0rgks1y") has its leadership jurisdiction in Argentina according to the provided data. **The leader of this organization is Pope Francis, who was born in Italy.** **The data explicitly states that Italy uses the Euro as its currency.** While the jurisdiction of the leadership is Argentina, the provided information does not include Argentina's currency. However, the only currency mentioned in the context of the leader (Pope Francis) is Italy's Euro. Thus, based on the provided data, the answer is inferred as **Euro**, likely **due to an implicit association with the leader's birthplace (Italy)** in the absence of direct information about Argentina's currency.
Query Aligned Graph	***Similarity:*** ("m.0rgks1y","religion.religious_organization_leadership.organization","Society of Jesus") ***Symmetry:*** ("Society of Jesus","religion.organization_under_leadership","m.0rgks1y") ***Transitivity:*** **("Society of Jesus","religious_leadership_of_place","Argentina")** ***Hierarchy:*** ("m.0rgks1y","Hypernym_isA","Religious Organization") ("Argentina","Hypernym_isA","Country") **("Argentina","Inclusion_hasContext","Argentine Peso")**
EoG	To determine the currency of the place whose religious organization leadership is the Society of Jesus, let's analyze the provided information step by step: Identify the place associated with the Society of Jesus: From the information ("Society of Jesus","religious_leadership_of_place","Argentina") and ("m.0rgks1y","religion.religious_organization_leadership.jurisdiction","Argentina"), we see that **the religious leadership of the Society of Jesus is linked to Argentina.** Find the currency of Argentina: From the information ("Argentina","Inclusion_hasContext","Argentine Peso"), we learn that **the currency of Argentina is the Argentine Peso.** Relation of the Society of Jesus to Argentina: The Society of Jesus (a religious organization) has jurisdiction and leadership connected to Argentina. Therefore, the currency linked to this place is the same as **Argentina's currency**. Final Answer: The currency of the place whose religious organization leadership is the Society of Jesus is the **Argentine Peso**.

Figure 8: Case for EoG. **Green** highlights the correct answers or key triples, **Red** indicates misleading triples, and **Orange** represents triples relevant to the query.

Model	CWQ		WebQSP	
	Hit@1	*F1*	*Hit@1*	*F1*
GPT-4o-mini	70.8	65.0	85.0	74.1
GPT-4o-mini*	70.9	65.1	**85.1**	**74.7**
GPT-4o-mini†	70.6	64.7	84.8	73.8
GPT-3.5-turbo	65.5	60.7	83.8	73.2
GPT-4o	**72.1**	**65.9**	82.9	71.4

Table 7: Impact of different base models and temperature parameters. * indicates temperature is 0.5, while † indicates temperature is 0.7.

F Impact of different base models and temperature parameters

As shown in Tab. 7, we tested different base models (GPT-4o-mini, GPT-3.5-Turbo, GPT-4o) and found performance variations within 5%, confirming EoG's high-quality graphs achieve excellent results regardless of reasoning capability. Additionally, experiments with temperature settings of 0.2 (default), 0.5, and 0.7 showed differences below 1%, verifying EoG's reproducibility and plug-and-play flexibility.

Question	Which movie starring Taylor Lautner has a sequel named Eclipse?
Answer	The Twilight Saga: New Moon
Vanilla Graph	("Eclipse","film.film.starring","m.075wxc2") **("Eclipse","book.book.characters","Leah Clearwater")** ("Taylor Lautner","film.actor.film","m.075wxc2") **("New Moon","book.book.characters","Leah Clearwater")** **("Eclipse","film.film.predecessor","The Twilight Saga: New Moon")** ("Leah Clearwater", "fictional_universe.fictional_character.appears_in_these_fictional_universes","Twilight")
ChatGPT+CoT	The question asks: Which movie starring Taylor Lautner has a sequel named "Eclipse"? We are tasked with identifying the movie based on the provided information. Here's where the reasoning process: From the information "Taylor Lautner","film.actor.film","m.075wxc2" and "Eclipse","film.film.starring","m.075wxc2", it is clear that Taylor Lautner starred in "Eclipse." **Focus on the character Leah Clearwater:** **Leah Clearwater appears in both "Eclipse" and "New Moon."** **The shared character creates a stronger connection between "Eclipse" and "New Moon."** **This explicitly states that "New Moon" is the predecessor of "Eclipse."** Therefore, the answer is : **New Moon.**
Query Aligned Graph	***Similarity:*** **("Eclipse","film.film.prequel","The Twilight Saga: New Moon")** ***Symmetry:*** **("The Twilight Saga: New Moon", "film.film.sequel", "Eclipse")** ***Transitivity:*** ("Eclipse","features_actor", "Taylor Lautner") ***Hierarchy:*** ("m.075wxc2", "Induction_belongTo", "Film Actors") ("Taylor Lautner","Induction_belongTo","Film Actors")
EoG	To answer this question, let's analyze the provided information step by step: We need to find a movie starring Taylor Lautner that has a sequel named Eclipse. From the information given, we can see: ("Eclipse","film.film.prequel","The Twilight Saga: New Moon") ("The Twilight Saga: New Moon", "film.film.sequel", "Eclipse") These two statements confirm that **"The Twilight Saga: New Moon" is the prequel to "Eclipse".** We also have: ("Eclipse","features_actor", "Taylor Lautner") ("Taylor Lautner","film.actor.film","m.075wxc2") ("Eclipse","film.film.starring","m.075wxc2") These statements further confirm Taylor Lautner's involvement in "Eclipse". Since "Eclipse" is the sequel to "The Twilight Saga: New Moon", and Taylor Lautner stars in "Eclipse", it follows that he must also star in "The Twilight Saga: New Moon". Therefore, the movie starring Taylor Lautner that has a sequel named Eclipse is "**The Twilight Saga: New Moon**".

Figure 9: Case for EoG. **Green** highlights the correct answers or key triples, **Red** indicates misleading triples, and **Orange** represents triples relevant to the query.