Foresight-on-Graph: Vision-Guided Knowledge Graph Reasoning with Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) demonstrate strong reasoning ability but are constrained by outdated knowledge and hallucinations. Knowledge Graphs (KGs) can help mitigate these limitations by providing structured relational knowledge and capturing global multi-entity, multi-hop dependencies beyond linear text. However, existing KG-enhanced LLM reasoning methods mostly rely on hop-by-hop textual traversal, rather than making multi-hop decisions with a broader perspective as humans do, thus underusing the structural advantages of KGs. This leads to myopic exploration and fragmented reasoning, wasting effort on irrelevant entities while potentially missing critical ones. To address these limitations, we propose Foresight-on-Graph (FoG), a novel framework that introduces visual relation graphs to expand the reasoning horizon at each step. FoG enables Vision-Language Models (VLMs) to simultaneously attend to multi-entity, multi-hop contexts, supporting foresighted planning and more efficient graph exploration. To facilitate effective visual reasoning, we design relation-based fuzzy matching and union-find algorithms to filter out irrelevant relations and coarsen the graph, condensing subgraphs with millions of entities into compact relation graphs with only tens of nodes. Experiments on three real-world datasets demonstrate that FoG consistently outperforms existing methods while significantly improving reasoning efficiency.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance in text generation (Chen et al., 2023a,b; Gong and Sun, 2024; Ji et al., 2024), commonsense question answering (Li et al., 2024a; Wang et al., 2024; Zhao et al., 2024), complex reasoning (Kojima et al., 2022; DeepSeek-AI, 2025), and other NLP tasks. However, their static parameters make them prone to outdated knowledge and hallucinations, while retraining is often prohibitively costly. Retrieval-Augmented Generation (RAG) mitigates these issues by providing external knowledge as context, but still faces accuracy and efficiency challenges over large-scale dense knowledge bases. Knowledge Graphs (KGs), with their structured representation of entities and relations, offer a promising solution and have been applied to knowledge-intensive tasks such as recommendation, healthcare QA, and scientific literature comprehension (Wei et al., 2024a; Soman et al., 2023; Li and Tajbakhsh, 2023). Yet, LLMs are inherently unstructured and often struggle to exploit graph topology effectively, making graph understanding and knowledge graph question answering (KGQA) an important open challenge.

Existing LLM-based KGQA methods can be broadly divided into two paradigms. The first is single-pass graph retrieval augmentation, represented by G-Retriever (He et al., 2024) and GNN-RAG (Mavromatis and Karypis, 2025), which train graph retrievers such as GNNs or MLPs to fetch relevant triples, paths, or subgraphs as context for LLMs. However, their retrieval process is often implicit and difficult to interpret, and errors caused by insufficient or noisy evidence are hard to refine once retrieval is completed. The second paradigm is synergy-augmented reasoning, exemplified by Think-on-Graph (ToG) (Sun et al., 2024), where LLMs interactively explore entities and relations on KGs to produce interpretable reasoning paths and support more flexible graph reasoning.

However, existing synergy-augmented methods still suffer from myopic exploration. As shown in Figure 1, when answering "Which of Taylor Swift's songs has won American Music Awards (AMA)?", they may identify the relation "Songs" in the first hop but lack foresight into which candidate songs will later connect to "Award". Under a limited exploration width, they must select only a subset of entities, often wasting steps on irrelevant paths

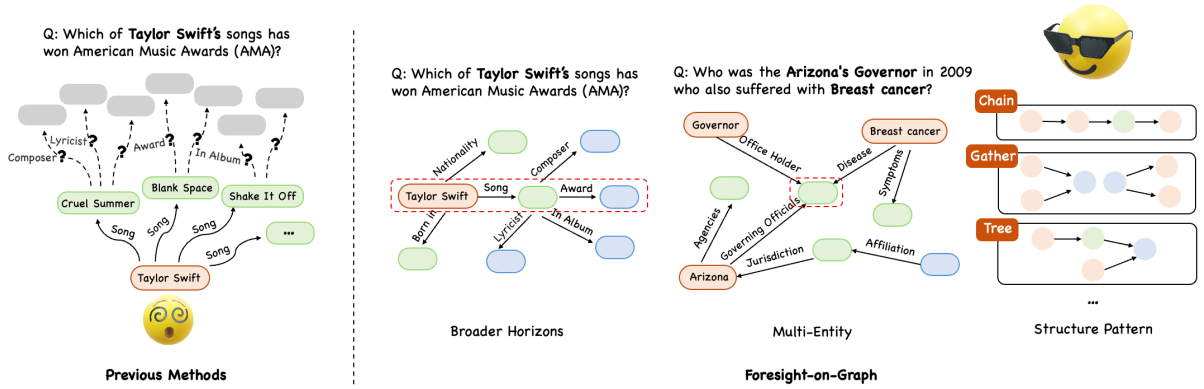

Figure 1: A sketched comparison with previous methods. Previous approaches blindly explore nodes with only first-order relations and entities visible at each exploration step. While FoG makes decisions from a high-level perspective with broader horizons, capturing the connections among multiple entities and leveraging structural patterns on the graph.

such as "Composer" or "In Album", even with reflection mechanisms such as PoG (Chen et al., 2024). Moreover, they often explore topic entities separately, producing fragmented evidence paths instead of capturing global inter-entity patterns.

To address these limitations, we propose a novel vision-guided framework called Foresight-on-Graph (FoG). By leveraging visual relation graphs, FoG is able to make more global and efficient decisions from a high level perspective. As shown in Figure 1, FoG offers three advantages:

Broader exploration horizon. In KGs, a single relation may connect thousands of entities, and subgraphs grow exponentially with hops. This is why previous methods only explore one-hop neighbors at each step. FoG overcomes this challenge by constructing highly condensed relation graphs to achieve a multi-hop field of view. Specifically, it extracts latent semantic relations from the question and filters question-irrelevant ones via fuzzy matching; it then applies a relation-based union-find to produce a coarsened relation graph, condensing subgraphs of millions of entities into merely tens of nodes. At each decision step, FoG thus has a wider field of view and can perform depth-first search guided by the task objective.

Holistic multi-entity reasoning. After visualizing the aforementioned relation graphs, FoG can simultaneously observe multiple topic entities and their interconnections within a single graph, rapidly locating relational paths between entities. By jointly attending to multi-hop, multi-entity information for holistic reasoning, FoG achieves more efficient and global reasoning, mirroring how humans reason over knowledge graphs.

Effective utilization of graph structure. Compared to text, vision is a more natural modality for representing graph structural information. Prior studies have shown that LLMs process textually described graph structures as a sequential context rather than implicitly constructing the underlying graph topology (Huang et al., 2024). FoG leverages VLMs to understand and analyze visual relation graphs, enabling the identification of structural patterns crucial for knowledge graph reasoning, such as chains, gather, scatter, and conjunction, thereby tackling more complex reasoning problems.

In summary, our contributions are as follows:

- We propose a novel vision-guided knowledge graph reasoning framework named FoG, which performs more efficient and foresighted reasoning from a higher-level perspective. By leveraging vision, FoG injects critical graph structural information into the model, enabling it to handle multi-hop and multi-entity questions.

- To enable visual reasoning on large-scale KGs, we specifically design a relation-based fuzzy selection and union-find algorithm to efficiently coarsen the graph into compact relation graphs, facilitating better comprehension and analysis by VLMs. Our coarsening method achieves nearly 99.9% reduction in the size of subgraphs.

- We conduct experiments on three real-world KGQA datasets, namely WebQSP, CWQ, and GrailQA. The results show the effectiveness and efficiency of FoG, revealing the potential of vision in guiding knowledge graph reasoning.

2 Preliminary

In this paper, we study LLMs for KGQA. A KG is represented as $\mathcal{G} = \{(e, r, e') | e, e' \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations, respectively. We use $\mathcal{F}$ to denote a set of triples (facts) from $\mathcal{G}$. Given a natural language question q, a Knowledge Graph $\mathcal{G}$, and topic entities $\mathcal{T}_q$ mentioned in q, the target of KGQA is to generate answers $\mathcal{A}_q$ to the question q. To better illustrate and organize the reasoning process, models usually attempt to generate reasoning paths from $\mathcal{T}_q$ to $\mathcal{A}_q$. A reasoning path can be defined as $p = e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \dots \xrightarrow{r_l} e_l$, where $e_i \in \mathcal{E}$ denotes the i-th entity and r_i denotes the i-th relation in the reasoning path. The corresponding relation path is the sequence of relations: $z = \{r_1, r_2, \dots, r_l\}$, where $r_i \in \mathcal{R}$ denotes the i-th relation in the path and l denotes the length of the path. Following previous works (Sun et al., 2024), we assume the topic entities $e_q \in \mathcal{T}_q$ mentioned in q and answers $a \in \mathcal{A}_q$ are labeled and linked to the corresponding entities in $\mathcal{G}$, i.e., $\mathcal{T}_q, \mathcal{A}_q \subseteq \mathcal{E}$.

3 Method

In this section, we present our proposed method, Foresight-on-Graph (FoG). Unlike existing methods that make decisions based on 1-hop neighbors, FoG leverages visual relation graph to enable multi-hop, multi-entity aware planning at each exploration step. As illustrated in Figure 2, FoG consists of four main stages: (1) Relation Path Prediction, (2) Relation Graph Construction, (3) Visual Graph Analysis, and (4) Dynamic Visual Exploration. We detail each stage in the following subsections. The prompt templates are provided in Appendix D.2.

3.1 Relation Path Prediction

For a knowledge graph question q, the underlying information consists of topic entities $\mathcal{T}_q$ and the implicit relation paths z from topic entities to the answer entities. While the topic entities are explicitly given, the relations implied by the question often differ significantly from the relation names in the knowledge graph. For instance, given the question "Who plays Bob Kelso in Scrubs?", the implied relations are *tv.tv_character.appeared_in_tv_program* and *tv.regular_tv_appearance.actor*. Directly performing semantic similarity matching based on q would fail to associate "plays" with the relevant relations. Instead, the model must first recognize that *Scrubs* is a TV show or *Bob Kelso* is a TV character, and then infer that the underlying reasoning path should be *Scrubs* $\xrightarrow{\text{tv_character}}$ *Bob Kelso* or *Bob Kelso* $\xrightarrow{\text{tv_actor}}$ *Scrubs*, in order to match the TV-related relations. To capture such subtle difference during mathcing, we use an LLM to infer the reasoning path underlying the question. We then extract the relations and formulate them into a relation path. Formally, we define the relation path prediction as:

$$z = \text{LLM}(q, \mathcal{T}_q) = \{r_1, r_2, \dots, r_l\} \tag{1}$$

where z denotes the predicted relation path consisting of l relations.

3.2 Relation Graph Construction

To make multi-hop neighborhoods tractable, FoG constructs a compact relation graph in two steps: it first filters question-irrelevant relations via fuzzy selection, then merges relation-equivalent entities with a union-find algorithm.

Fuzzy Selection. Guided by the predicted relation path z, we encode candidate relations with a pre-trained language model (e.g., Sentence-BERT (Reimers and Gurevych, 2019)) and select relations by semantic similarity. We match both the whole path and each relation in the path, then merge and deduplicate the results. For Freebase, we set $k_{\text{path}} = 20$ and $k_{\text{rel}} = 10$ by default.

Let $\phi(\cdot)$ denote the sentence encoder and $\mathcal{R}_{\text{cand}}$ the relations in the k-hop neighborhood of topic entities. We compute $s_{\text{path}}(r) = \cos(\phi(z), \phi(r))$ and $s_{\text{rel}}(r, r_i) = \cos(\phi(r_i), \phi(r))$, and select

$$\begin{aligned}\mathcal{R}_{\text{sel}} = &\text{Top}_{k_{\text{path}}}(\mathcal{R}_{\text{cand}}, s_{\text{path}}) \\ &\cup \bigcup_{r_i \in z} \text{Top}_{k_{\text{rel}}}(\mathcal{R}_{\text{cand}}, s_{\text{rel}}(\cdot, r_i)).\end{aligned} \tag{2}$$

Based on $\mathcal{R}_{\text{sel}}$, we extract triples whose relations are selected and retain the connected component reachable from the topic entities, obtaining the filtered subgraph $\mathcal{G}_{\text{sub}}$.

Relation-based Union-Find. Even after fuzzy selection, a single relation type may still include many instances. For example, the question subgraph in Figure 2 retains only 29 relation types yet still contains nearly 300,000 entities and 580,000 edges, motivating further coarsening.

Our coarsening principle focuses on relations rather than entity names. For non-topic entities

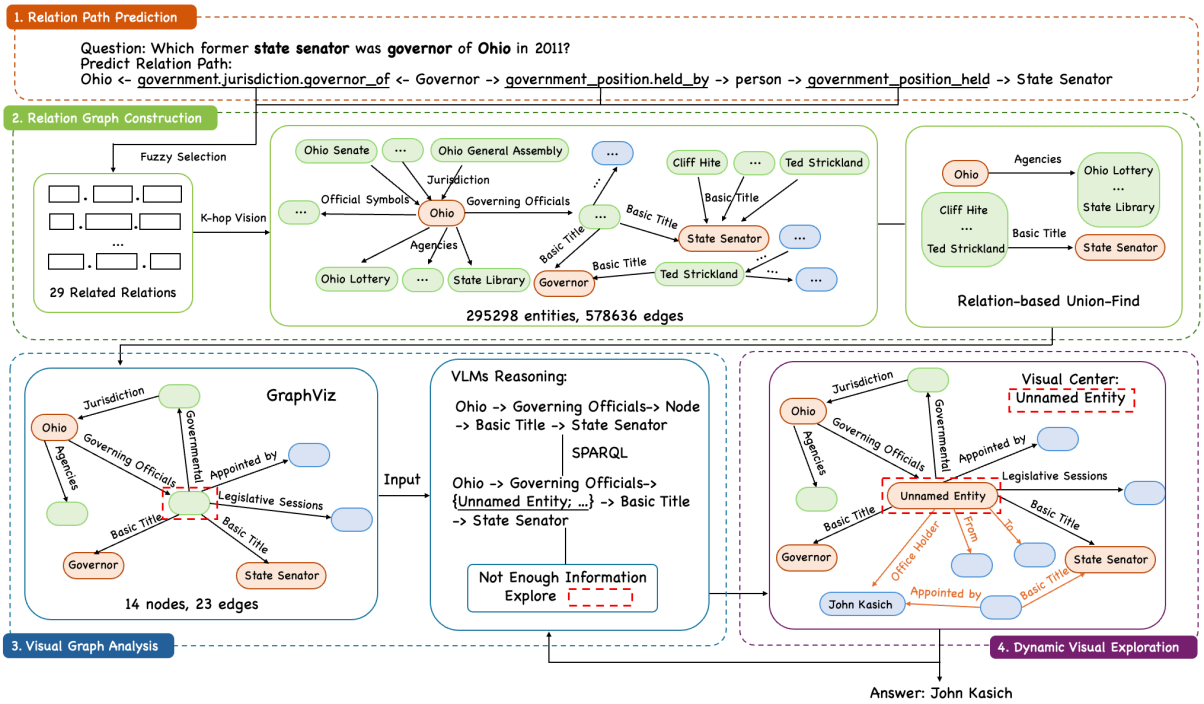

Figure 2: An example workflow of FoG, which includes four key components: Relation Path Prediction, Relation Graph Construction, Visual Graph Analysis, and Dynamic Visual Exploration.

$e_1, e_2 \in \mathcal{E} \setminus \mathcal{T}_q$, Relation-based Union-Find merges them when they share the same head-relation or tail-relation pattern:

$$e_1 \sim e_2 \Leftrightarrow \exists e, r: \ (e, r, e_1), (e, r, e_2) \in \mathcal{G}_{\text{sub}} \vee \exists e, r: \ (e_1, r, e), (e_2, r, e) \in \mathcal{G}_{\text{sub}}. \tag{3}$$

Topic entities $\mathcal{T}_q$ are excluded from merging and serve as anchors. After iterative union-find with path compression, the relation graph $\mathcal{G}_{\text{coarse}}$ consists of topic entities, collapsed entity equivalence classes, and relations. As shown in Figure 2, this reduces the example graph to 14 nodes and 23 edges.

3.3 Visual Graph Analysis.

After obtaining the coarsened relation graph $\mathcal{G}_{\text{coarse}}$, we visualize it as a 2D visual graph to better convey the graph structure and multi-hop, multi-entity information. On this basis, we invoke VLMs to reason over the visual graph, performing high-level analysis and path exploration as a human would.

Graph Visualization. We visualize the coarsened relation graph using Graphviz with the Dot layout, which preserves the hierarchical structure among nodes at different hops (Gansner and North, 2000). We retain only the names of topic entities and relations, while all other equivalent entity sets are represented as nodes. An illustration is provided at the bottom of Figure 2. This design emphasizes relations and reduces the influence of irrelevant entity names. Specific examples can be found in Appendix G.

Visual Path Exploration. Given the visual graph image I_{graph} and the question q, we prompt the VLM to identify relation paths from the topic entity to the answer entity. Unlike prior work that performs single-hop relation and entity exploration, visual path exploration enables multi-hop path discovery. Prior methods are analogous to breadth-first search, whereas FoG operates as a depth-first search with local visibility. We then replace unresolved equivalent entity sets with query variables and convert each VLM-selected path into a SPARQL query to retrieve instantiated reasoning paths $\mathcal{P}_{\text{inst}}$. The SPARQL templates are provided in Appendix D.1.

3.4 Dynamic Visual Exploration

Since multiple instantiated reasoning paths may contain diverse pieces of evidence for answering the question, we first summarize $\mathcal{P}_{\text{inst}}$ into $\mathcal{P}_{\text{sum}}$, then ask the LLM to judge whether the current evidence answers the question and to output the answer if sufficient. If the evidence is insufficient, the VLM selects exploration entities E_{next} from the current local relation graph. FoG recenters the local

view on E_{next}, supplements their one-hop relations for fine-grained search, and retains the shortest paths from the topic entities to the exploration entities to preserve context. The process repeats until the model finds sufficient evidence or reaches the maximum number of attempts.

4 Experiments

In this section, we first validate that vision-based VLMs are more effective than text-based LLMs for path analysis on graphs, thereby justifying our earlier choice of VLMs for graph analysis and path exploration. We then present extensive experiments of FoG on three real-world datasets. We first describe the setups of our experiments, and then provide the experimental results to demonstrate not only the effectiveness and but also the efficiency of FoG. We further conduct ablation experiments to demonstrate the necessity of each component.

4.1 VLMs in Path Tasks

To justify using VLMs for visual graph analysis, we evaluate whether visual inputs better support path-level reasoning than textual inputs. Following the setup of prior work (He et al., 2024), we compare GPT-4o and Qwen2.5-VL-72B on ExplaGraphs, a stance-prediction task over explanation graphs, by feeding each graph either as an image or as text and measuring accuracy. Figure 3 shows an illustrative example, with dataset details and example analysis provided in Appendix A. As shown in Table 1, both models achieve substantially higher accuracy with visual graphs, indicating that visual graph representations are more effective for multi-hop relational comprehension and path-level reasoning.

Table 1: Stance prediction accuracy (%) on ExplaGraphs under different input modalities.

Modality	GPT-4o	Qwen2.5-VL-72B
Text	89.7	80.4
Vision	97.1	93.7

4.2 Experimental Setups

Datasets & Evaluation Metrics. We evlauate FoG on the test splits of three widely-used multi-hop KGQA benchmarks: CWQ (Talmor and Berant, 2018), WebQSP (Yih et al., 2016), and GrailQA (Gu et al., 2021), all of which are grounded in the Freebase (Bollacker et al., 2008) knowledge graph. For GrailQA, we follow the same test subset adopted by ToG (Sun et al., 2024) for computational efficiency, and additionally report performance across its three generalization levels: I.I.D., Compositional, and Zero-shot. Consistent with prior work (Li et al., 2023; Jiang et al., 2023a; Sun et al., 2024), we adopt exact match accuracy (Hit) as the evaluation metric.

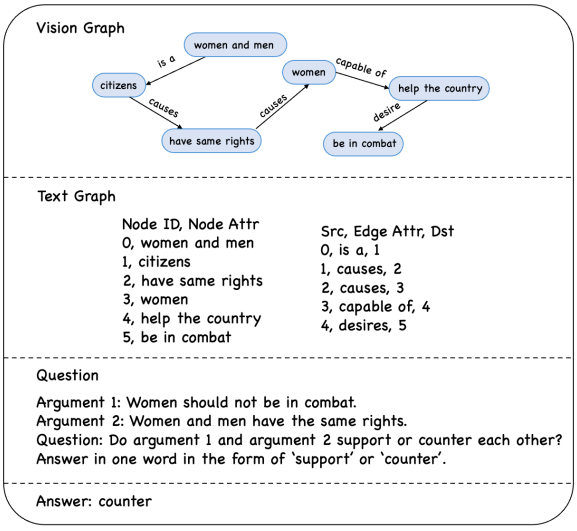

Figure 3: An illustrative example from ExplaGraphs.

Baselines & Settings. Following the LLMs-for-KGQA paradigms discussed in the Introduction, we categorize the comparison methods into three groups: (1) *LLM-only Reasoning*, which directly prompts LLMs without external KG access, represented by CoT (Wei et al., 2022). (2) *Fine-tuned Retrieval-Augmented*, which trains dedicated retrievers or semantic parsers to fetch relevant KG information for LLMs in a single pass, represented by GNN-RAG (Mavromatis and Karypis, 2025). (3) *Synergy-Augmented*, which enables LLMs to interactively explore KGs through multi-step prompting, represented by PoG (Chen et al., 2024). The complete baseline list is provided in Appendix B.

For each baseline, we report the best results from their original papers. For FoG, we set the order of the relation graph $k = 2$ by default. The fuzzy selection parameters are set to $k_{\text{path}} = 20$ and $k_{\text{rel}} = 10$ by default. We use GPT-4o as the underlying VLM, with the maximum number of attempts for dynamic visual exploration set to 3. Additionally, we introduce a variant **FoG-T**, which differs from FoG only in the graph representation modality: it replaces the visual relation graph with a textual representation while sharing all other com-

ponents. This variant is designed to highlight the critical role of vision in guiding graph comprehension and reasoning.

4.3 Performance Comparison

Table 2: Performance comparison of different methods on CWQ and WebQSP.

Method	CWQ	WebQSP
LLM-only Reasoning		
IO Prompt	37.6	63.3
CoT	38.8	62.2
SC	45.4	61.1
Fine-tuned Retrieval-Augmented		
UniKGQA	51.2	79.1
TIARA	-	75.2
RE-KBQA	50.3	74.6
DeCAF	70.4	82.1
RoG	62.6	85.7
GNN-RAG	66.8	85.7
GNN-RAG+RA	**68.7**	**90.7**
Synergy-Augmented w/ GPT-3.5 turbo or others		
KD-CoT	50.5	73.7
KB-BINDER	-	74.4
StructGPT	54.3	72.6
ToG	57.1	76.2
PoG	63.2	80.4
FoG-T	67.2	86.0
FoG	**72.3**	**88.7**
Synergy-Augmented w/ GPT-4		
InteractiveKBQA	59.2	72.5
ToG	67.6	82.6
PoG	75.0	87.3
FoG-T	76.7	89.3
FoG	**79.1**	**91.5**

Table 2 presents the comparison results between FoG and other KGQA methods on the CWQ and WebQSP datasets. FoG substantially outperforms all existing synergy-augmented methods, demonstrating that broadening the reasoning horizon brings significant benefits and further revealing the potential of vision in assisting LLMs with knowledge graph reasoning. Meanwhile, FoG-T, which relies solely on text, also consistently surpasses PoG. This highlights the importance of the relation graph, which effectively coarsens the massive k-order subgraph into a highly condensed and relevant relational graph. Although it cannot provide the model with graph structure as intuitively as vision does, it still offers a broader decision horizon, enabling the LLM to make foresighted decisions at each exploration step. Moreover, despite being a training-free method, FoG achieves performance on par with fine-tuned retrieval-augmented methods. Even when using GPT-3.5-turbo, FoG surpasses the strong baseline GNN-RAG+RA on CWQ, indicating that vision-guided knowledge graph exploration can match or even exceed the retrieval effectiveness of most fine-tuned retrievers.

Table 3 reports the results on GrailQA. Compared to fine-tuned methods, FoG achieves strong performance across all three generalization levels, demonstrating its ability to recognize and analyze diverse relational patterns with robust generalization capability. Unlike fine-tuned methods, which tend to suffer noticeable performance degradation when encountering out-of-distribution samples, FoG maintains consistent effectiveness, making it applicable to a broader range of scenarios.

4.4 Ablation Study

We disable each stage in turn and report the results on WebQSP and CWQ in Table 4, using GPT-3.5-turbo as the base LLM and GPT-4o as the VLM. Removing the Relation Path Prediction (RPP) stage, either by directly using the question as the matching query (RPP-Q) or by replacing the predicted relations with a random sample (RPP-R), causes a large drop, confirming that RPP supplies the semantic anchors that fuzzy selection relies on. Removing Dynamic Exploration leaves FoG with a single reasoning pass and consistently degrades performance, indicating that even with broad initial coverage, multi-round refinement is essential for multi-hop questions. Replacing the visual graph with a textual graph (FoG-T) also lowers accuracy, with a larger drop on the more complex CWQ dataset, confirming the critical role of vision in conveying graph structure. The Graph Coarsening stage cannot be ablated independently: without it, the filtered subgraph still contains thousands of entities and relations, which exceeds the context window of every VLM and forces the pipeline to abort.

4.5 Sensitivity Study

Hop Order k of the Relation Graph. Table 5 reports how performance changes as we vary the hop order k used to build the local subgraph. On WebQSP, accuracy is already near-saturated at $k = 2$,

Table 3: Performance comparison of different methods on GrailQA.

Method	GrailQA			
	Overall	I.I.D.	Comp.	Zero-shot
LLM-only Reasoning				
IO Prompt	29.4	-	-	-
CoT	28.1	-	-	-
SC	29.6	-	-	-
Fine-tuned Retrieval-Augmented				
RnG-KBQA	68.8	86.2	63.8	63.0
TIARA	73.0	87.8	69.2	68.0
FC-KBQA	73.2	88.5	70.0	67.6
Pangu	75.4	84.4	74.6	71.6
FlexKBQA	62.8	71.3	59.1	60.6
GAIN	76.3	88.5	73.7	71.8
Synergy-Augmented w/ GPT-3.5 or others				
KB-BINDER	50.6	-	-	-
ToG	68.7	70.1	56.1	72.7
PoG	76.5	76.3	62.1	81.7
FoG-T	80.7	84.6	73.7	81.5
FoG	**83.6**	**88.7**	**79.2**	**82.9**
Synergy-Augmented w/ GPT-4				
ToG	81.4	79.4	67.3	86.5
PoG	84.7	87.9	69.7	88.6
FoG-T	87.0	83.3	88.9	89.1
FoG	**90.6**	**86.4**	**92.0**	**93.2**

Table 4: Component ablation on WebQSP and CWQ. RPP-Q and RPP-R denote removing Relation Path Prediction with question-only and random-relation variants, respectively. *w/o Vision* corresponds to the FoG-T variant that replaces the visual relation graph with a textual representation. OOC denotes that the relation graph exceeds the VLM context window and the run cannot complete.

Variant	WebQSP	CWQ
FoG (full)	**88.7**	**72.3**
w/o RPP-Q	71.5	47.7
w/o RPP-R	48.5	23.3
w/o Vision	86.0	67.2
w/o Exploration	76.7	61.1
w/o Coarsening	OOC	OOC

with $k = 3$ yielding only a marginal change. On CWQ, $k = 1$ causes the answer entity to fall outside the reachable neighborhood for many questions and incurs a sharp accuracy drop, while $k = 3$ enlarges the candidate relation set and slightly hurts accuracy due to extra noise. Beyond accuracy, $k = 3$ also incurs substantially higher computational cost, as the subgraph size and the corresponding relation graph grow with k. Taking accuracy and efficiency together, $k = 2$ offers the best trade-off between coverage, noise, and cost.

Table 5: Sensitivity to the hop order k of the local relation graph.

k	**1**	**2**	**3**
WebQSP	87.8	88.7	**88.8**
CWQ	49.2	**72.3**	70.8

Fuzzy-Selection Budget $(k_{\text{path}}, k_{\text{rel}})$**.** We study the fuzzy-selection budget that decides how many relations survive after Relation Path Prediction. As shown in Table 6, accuracy peaks at $(20, 10)$ and degrades on either side. Smaller budgets miss critical relations, while larger budgets retain more noisy relations that distract visual path exploration. At $(20, 10)$, the predicted relation set covers the gold relation chain fully in 85.32% of WebQSP cases and at least partially in 91.70%, which is sufficient for downstream coarsening to receive the right structural skeleton.

Table 6: Sensitivity to the fuzzy-selection budget $(k_{\text{path}}, k_{\text{rel}})$.

$(k_{\text{path}}, k_{\text{rel}})$	**(10,5)**	**(20,10)**	**(30,15)**	**(40,20)**
WebQSP	82.5	**88.7**	87.2	85.9
CWQ	61.3	**72.3**	71.9	70.2

4.6 Efficiency Study

We compare the reasoning efficiency of FoG with two classic synergy-augmented methods, ToG and PoG. The results are summarized in Table 7, where for clarity we merge the text and vision LLM invocations into a single *Calls* column and the corresponding text and vision input tokens into a single *Tokens* column; the unmerged breakdown is provided in Appendix F. FoG achieves a substantial reduction in total LLM calls and input tokens compared with both baselines, since its broader single- and multi-hop exploration eliminates the need to traverse a large number of irrelevant entities and keeps the average number of visual reasoning

rounds low. Wall-clock time follows the same trend, with FoG being the fastest across both datasets, although the gap is smaller than in LLM calls because visual reasoning operates on larger relation graphs and SPARQL-based retrieval along relation paths incurs additional time overhead. Overall, FoG makes each exploration step more valuable, accomplishing precise path exploration with only a small number of LLM calls while delivering superior retrieval and reasoning performance.

Table 7: Efficiency comparison between FoG and the synergy-augmented baselines ToG and PoG.

Dataset	Method	Calls	Tokens	Time (s)
CWQ	ToG	22.6	8182.9	96.5
	PoG	13.3	7803.0	23.3
	FoG	**8.6**	**7760.1**	**21.2**
WebQSP	ToG	15.9	6031.2	63.1
	PoG	9.0	5234.8	16.8
	FoG	**5.3**	**5579.0**	**13.2**

5 Related Works

5.1 Retrieval-Augmented LLMs for KGQA

KGQA answers natural language questions by retrieving and reasoning over structured KG facts. Earlier work mainly relies on semantic parsing or learned entity-relation representations (Berant et al., 2013; Yih et al., 2016; Gu et al., 2021; Sun et al., 2019). Recent retrieval-augmented methods enhance LLMs with retrieved triples, subgraphs, or reasoning paths (Baek et al., 2023; Luo et al., 2024; Ye et al., 2022; Yu et al., 2023; He et al., 2024; Mavromatis and Karypis, 2025). Despite their effectiveness, these methods usually follow a retrieve-then-reason pipeline: KG evidence is selected before LLM reasoning and then provided as context, leaving limited room for the model to revise exploration decisions during reasoning and letting retrieval errors propagate downstream.

5.2 Synergy-Augmented LLMs for KGQA

Synergy-augmented methods allow LLMs to interact with KGs over multiple steps, using model decisions to guide graph exploration and newly retrieved structures to refine subsequent reasoning (Pan et al., 2024). Unlike retrieve-then-reason pipelines, this enables backtracking and refocusing on the fly. Representative systems perform beam-style triple selection, KG-guided retrieval, entity-relation navigation, graph-of-thought construction, or structured-data interaction (Sun et al., 2024; Ma et al., 2024; Guo et al., 2024; Wen et al., 2024; Jiang et al., 2023a; Luo et al., 2023; Kim et al., 2023). These approaches improve faithfulness and interpretability, but they still largely expose KG neighborhoods as textual triples or local candidates, which can make long-range multi-entity structure difficult for LLMs to perceive holistically, leading to shortsighted decisions that accumulate errors over long reasoning chains.

5.3 Vision for Graph Reasoning

Recent work studies whether VLMs can reason over graph drawings and whether visual representations can overcome the limitations of text-only graph descriptions (Wang et al., 2023a; Zhao et al., 2025). Benchmarks and systems for visual graph analysis show both the promise and limitations of current multimodal models on structural tasks (Williams and Carley, 2024; Zhu et al., 2025; Sartori et al., 2025). Wei et al. (2024b) propose GITA, which integrates graph-to-visual and textual representations for vision-language graph reasoning, demonstrating that combining visual and textual modalities can enhance graph reasoning performance. Unlike these studies, which mainly target abstract graph-theoretic tasks, we use visual graph representations to guide question answering over real-world knowledge graphs.

6 Conclusion

We propose Foresight-on-Graph (FoG), a vision-guided framework that uses visual relation graphs to broaden LLM-based KG reasoning beyond myopic hop-by-hop traversal. By combining relation-based fuzzy matching with union-find graph coarsening, FoG turns large local subgraphs into compact visual structures that VLMs can inspect for multi-hop, multi-entity path exploration. Experiments on CWQ, WebQSP, and GrailQA show that FoG consistently improves accuracy and reduces LLM calls compared with prior iterative methods. Its gains over the text-only FoG-T variant further indicate that visual graph structure provides useful signals for holistic KG reasoning. FoG therefore points to a broader direction for KGQA: combining symbolic graph access with multimodal structure perception to support more efficient and globally informed reasoning.

Limitations

FoG still depends on the visual reasoning ability of current VLMs (see Appendix E for details). When relation graphs become large or visually dense, models may miss valid paths or misread graph topology, so stronger graph-aware VLMs and more scalable visualization strategies remain important future directions.

Ethical considerations

This work studies FoG in a controlled research setting, aiming to improve the reliability and interpretability of LLM-based reasoning over knowledge graphs. All experiments are conducted on public KGQA benchmarks (CWQ, WebQSP, GrailQA) grounded in the Freebase knowledge graph, and contain no personally identifying information, private user records, or offensive content.

References

Jinheon Baek, Alham Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *EMNLP*, pages 1533–1544.

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

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.

Yong Cao, Xianzhi Li, Huiwen Liu, Wen Dai, Shuai Chen, Bin Wang, Min Chen, and Daniel Hershcovich. 2023. Pay more attention to relation exploration for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2119–2136.

Liyi Chen, Zhi Li, Weidong He, Gong Cheng, Tong Xu, Nicholas Jing Yuan, and Enhong Chen. 2023a. Entity summarization via exploiting description complementarity and salience. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8297–8309.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. 2024. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Xi Chen, Xinjiang Lu, Haoran Xin, Wenjun Peng, Haoyang Duan, Feihu Jiang, Jingbo Zhou, and Hui Xiong. 2023b. A table-to-text framework with heterogeneous multidominance attention and self-evaluated multi-pass deliberation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 607–620.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Emden R. Gansner and Stephen C. North. 2000. An open graph visualization system and its applications to software engineering. *Softw. Pract. Exp.*, 30(11):1203–1233.

Zheng Gong and Ying Sun. 2024. Graph reasoning enhanced language models for text-to-sql. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2447–2451.

Yu Gu, Xiang Deng, and Yu Su. 2023. Don't generate, discriminate: A proposal for grounding language models to real-world environments. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4928–4949.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases. In *WWW*, pages 3477–3488.

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

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. 2024. Can llms effectively leverage graph structural information through prompts, and why? *Trans. Mach. Learn. Res.*, 2024.

Bin Ji, Huijun Liu, Mingzhe Du, and See-Kiong Ng. 2024. Chain-of-thought improves text generation with citations in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18345–18353.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

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 International Conference on Learning Representations*.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. Kg-gpt: A general framework for reasoning on knowledge graphs using large language models. *arXiv preprint arXiv:2310.11220*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shengzhi Li and Nima Tajbakhsh. 2023. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *CoRR*, abs/2308.03349.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6980.

Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. 2024a. 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.

Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. 2024b. Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18608–18616.

Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538*.

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

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

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

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *TKDE*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP*.

Camilo Chacón Sartori, Christian Blum, and Filippo Bistaffa. 2025. Visgraphvar: A benchmark generator for assessing variability in graph analysis using large vision-language models. *IEEE Access*, 13:21788–21810.

Yiheng Shu and Zhiwei Yu. 2024. Distribution shifts are bottlenecks: Extensive evaluation for grounding language models to knowledge bases. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 71–88.

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.

Karthik Soman, Peter W. Rose, John H. Morris, Rabia E. Akbas, Brett Smith, Braian Peetoom, Catalina Villouta-Reyes, Gabriel Cerono, Yongmei Shi, Angela Rizk-Jackson, Sharat Israni, Charlotte A. Nelson, Sui Huang, and Sergio E. Baranzini. 2023. Biomedical knowledge graph-enhanced prompt generation for large language models. *CoRR*, abs/2311.17330.

Haitian Sun, Tania Bedrax-Weiss, and William W Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. *arXiv preprint arXiv:1904.09537*.

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

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL*.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023a. Can language models solve graph problems in natural language? In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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

Lei Wang, Yi Hu, Jiabang He, Xing Xu, Ning Liu, Hui Liu, and Heng Tao Shen. 2024. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19162–19170.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

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.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024a. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, pages 806–815. ACM.

Yanbin Wei, Shuai Fu, Weisen Jiang, Zejian Zhang, Zhixiong Zeng, Qi Wu, James T. Kwok, and Yu Zhang. 2024b. GITA: graph to visual and textual integration for vision-language graph reasoning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2024. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. In *ACL*.

Evan M. Williams and Kathleen M. Carley. 2024. Multimodal llms struggle with basic visual network analysis: A VNA benchmark. In *Social, Cultural, and Behavioral Modeling - 17th International Conference, SBP-BRiMS 2024, Pittsburgh, PA, USA, September 18-20, 2024, Proceedings*, volume 14972 of *Lecture Notes in Computer Science*, pages 15–24. Springer.

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

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.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *ACL*, pages 201–206.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. 2023. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. In *The International Conference on Learning Representations*.

Lingxi Zhang, Jing Zhang, Yanling Wang, Shulin Cao, Xinmei Huang, Cuiping Li, Hong Chen, and Juanzi Li. 2023. Fc-kbqa: A fine-to-coarse composition framework for knowledge base question answering. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Lili Zhao, Qi Liu, Linan Yue, Wei Chen, Liyi Chen, Ruijun Sun, and Chao Song. 2024. Comi: Correct and mitigate shortcut learning behavior in deep neural networks. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 218–228.

Xinjian Zhao, Wei Pang, Zhongkai Xue, Xiangru Jian, Lei Zhang, Yaoyao Xu, Xiaozhuang Song, Shu Wu, and Tianshu Yu. 2025. The underappreciated power of vision models for graph structural understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yingjie Zhu, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. 2025. Benchmarking and improving large vision-language models for fundamental visual graph understanding and reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 30678–30701. Association for Computational Linguistics.

A ExplaGraphs Modality Study Details

ExplaGraphs is a generative commonsense reasoning dataset for stance prediction in debates. It provides commonsense-augmented explanation graphs for evaluating whether arguments support or refute a given belief. In our modality study, the model determines whether an argument is supportive or contradictory, and accuracy is used as the evaluation metric. Each argument corresponds to an implicit path, so the task requires verifying whether the relevant path exists in the graph.

In Figure 3, Argument 1 conflicts with the path *women* $\rightarrow$ *help the country* $\rightarrow$ *be in combat*, while Argument 2 aligns with *women and men* $\rightarrow$ *citizens* $\rightarrow$ *have same rights*. Therefore, the answer is "counter".

B Complete Baseline List

We group comparison methods into three LLMs-for-KGQA paradigms. *LLM-only Reasoning* includes IO Prompt (Brown et al., 2020), Chain-of-Thought prompting (CoT) (Wei et al., 2022), and Self-Consistency (SC) (Wang et al., 2023c). *Fine-tuned Retrieval-Augmented* methods include UniKGQA (Jiang et al., 2023b), TIARA (Shu et al., 2022), RE-KBQA (Cao et al., 2023), FC-KBQA (Zhang et al., 2023), Pangu (Gu et al., 2023), FlexKBQA (Li et al., 2024b), GAIN (Shu and Yu, 2024), RoG (Luo et al., 2024), and GNN-RAG (Mavromatis and Karypis, 2025). *Synergy-Augmented* methods include KD-CoT (Wang et al., 2023b), InteractiveKBQA (Xiong et al., 2024), ToG (Sun et al., 2024), and PoG (Chen et al., 2024).

C Algorithm

In this section, we provide the complete pseudocode for the Foresight-on-Graph (FoG) algorithm.

Algorithm 1: Foresight-on-Graph (FoG)

Require: Question q, Topic entities $\mathcal{T}_q$, Knowledge Graph $\mathcal{G}$, Sentence encoder ϕ, Max iterations T, Top-k_{path} path relations, Top-k_{rel} individual relations, k-hop neighborhood
Ensure: Answer a

// Stage 1: Relation Path Prediction
$z \leftarrow \text{LLM}(q, \mathcal{T}_q)$ ▷ Predict relation path $\{r_1, r_2, \ldots, r_l\}$
// Stage 2: Relation Graph Construction
$\mathcal{R}_{\text{cand}} \leftarrow \text{GetRelations}(\mathcal{T}_q, \mathcal{G}, k)$ ▷ Get k-hop neighborhood relations
for each $r \in \mathcal{R}_{\text{cand}}$ **do**
  $s_{\text{path}}(r) \leftarrow \cos(\phi(z), \phi(r))$ ▷ Path-level similarity
  $s_{\text{rel}}(r, r_i) \leftarrow \cos(\phi(r_i), \phi(r)),\ \forall r_i \in z$ ▷ Relation-level similarity
end for
$\mathcal{R}_{\text{sel}} \leftarrow \text{Top}_{k_{\text{path}}}(\mathcal{R}_{\text{cand}}, s_{\text{path}}) \cup \bigcup_{r_i \in z} \text{Top}_{k_{\text{rel}}}(\mathcal{R}_{\text{cand}}, s_{\text{rel}}(\cdot, r_i))$
$\mathcal{G}_{\text{sub}} \leftarrow \text{ConnComp}(\{(e_h, r, e_t) \in \mathcal{F}_{k\text{-hop}} \mid r \in \mathcal{R}_{\text{sel}}\}, \mathcal{T}_q)$
$\mathcal{G}_{\text{coarse}} \leftarrow \text{UnionFindCoarsen}(\mathcal{G}_{\text{sub}}, \mathcal{T}_q)$ ▷ Graph coarsening
$e_{\text{center}} \leftarrow \mathcal{T}_q$ ▷ Initialize center entities
$\mathcal{P}_{\text{all}} \leftarrow \emptyset$ ▷ Initialize collected paths
for $t = 1$ to T **do**
  // Stage 3: Visual Graph Analysis
  $I_{\text{graph}} \leftarrow \text{Visualize}(\mathcal{G}_{\text{coarse}}, e_{\text{center}})$ ▷ Generate visual graph
  $p \leftarrow \text{VLM}(I_{\text{graph}}, q)$ ▷ Visual path exploration
  $\mathcal{P}_{\text{inst}} \leftarrow \text{SPARQL}(p, \mathcal{G})$ ▷ Retrieve instantiated paths
  $\mathcal{P}_{\text{all}} \leftarrow \mathcal{P}_{\text{all}} \cup \mathcal{P}_{\text{inst}}$
  // Stage 4: Dynamic Visual Exploration
  $\mathcal{P}_{\text{sum}} \leftarrow \text{Summary}(\mathcal{P}_{\text{all}})$
  $(c, a) \leftarrow \text{LLM}(q, \mathcal{P}_{\text{all}}, \mathcal{P}_{\text{sum}})$ ▷ Judge answerability
  if $c =$ yes **then**
    return a ▷ Return answer if sufficient evidence
  end if
  $E_{\text{next}} \leftarrow \text{VLM}(q, I_{\text{graph}}, \mathcal{P}_{\text{sum}}, \mathcal{P}_{\text{inst}})$ ▷ Select exploration entities
  $e_{\text{center}} \leftarrow E_{\text{next}}$ ▷ Shift local view
  $\mathcal{G}_{\text{coarse}} \leftarrow \text{UpdateGraph}(\mathcal{G}_{\text{coarse}}, E_{\text{next}}, \mathcal{T}_q)$
end for
return a ▷ Return best answer after max iterations

D SPARQL and Prompts

In this section, we show all the SPARQL queries and prompts used in our main experiments. First, we pre-define SPARQL templates for Freebase queries, which can be executed by filling in the appropriate entity ID (mid) and relation.

D.1 Pre-defined SPARQL

D.1.1 Relation Search

The following SPARQL queries retrieve all relations connected to a given entity, either as head (outgoing) or tail (incoming) relations.

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?relation
WHERE {
    ns:|\textcolor{purple}{mid}| ?
        relation ?x .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?relation
WHERE {
    ?x ?relation ns:|\textcolor{purple}{
        mid}| .
}
```

D.1.2 Entity Search

The following SPARQL queries retrieve entities connected to a given entity via a specific relation.

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
    ns:|\textcolor{purple}{mid}| ns:|\
        textcolor{cyan}{relation}| ?
        tailEntity .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?headEntity
WHERE {
    ?headEntity ns:|\textcolor{cyan}{
        relation}| ns:|\textcolor{purple
        }{mid}| .
}
```

D.1.3 Combined Relation and Entity Search

This query retrieves both connected entities and their relations in both directions simultaneously, along with entity names.

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?relation ?connectedEntity ?
    connectedEntityName ?direction
WHERE {
    {
        ns:|\textcolor{purple}{mid}| ?
            relation ?connectedEntity .
        OPTIONAL {
            ?connectedEntity ns:type.
                object.name ?name .
            FILTER(lang(?name) = 'en')
        }
        BIND(COALESCE(?name, "Unnamed
            Entity") AS ?
            connectedEntityName)
        BIND("tail" AS ?direction)
    }
    UNION
    {
        ?connectedEntity ?relation ns:|\
            textcolor{purple}{mid}| .
        OPTIONAL {
            ?connectedEntity ns:type.
                object.name ?name .
            FILTER(lang(?name) = 'en')
        }
        BIND(COALESCE(?name, "Unnamed
            Entity") AS ?
            connectedEntityName)
        BIND("head" AS ?direction)
    }
}
```

D.1.4 Path Verification Query

This SPARQL template is used to verify predicted reasoning paths from the vision model. The query dynamically constructs triple patterns based on the path structure, where variables (`?var0`, `?var1`, etc.) represent intermediate nodes.

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT |\textcolor{cyan}{?var0
    ?var1 ...}|
WHERE {
    ns:|\textcolor{purple}{entity\_0}|
        ns:|\textcolor{cyan}{relation\_1
        }| |\textcolor{cyan}{?var1}| .
    |\textcolor{cyan}{?var1}| ns:|\
        textcolor{cyan}{relation\_2}| |\
        textcolor{cyan}{?var2}| .
    ...
}
LIMIT 100
```

where `entity_0` represents known topic entities, `relation_i` represents the relations in the path, and `?var_i` represents variable positions for intermediate or answer entities.

D.2 Prompts

In this section, we detail the prompts required for our main experimental procedures.

D.2.1 Relation Path Prediction

This prompt instructs the LLM to predict the chain of relations that connects topic entities to the answer entity, serving as the foresight guidance for subsequent graph exploration.

Vision Path Generation Prompt Template

You will receive a multi-hop question along with a list of topic entities. Your task is to predict the chain of relations that connect the topic entities to the answer.
Focus on predicting the RELATIONS (not entities) that would be needed to traverse the knowledge graph from topic entities to the answer.
Please provide: 1. A predicted relation chain showing how to reach the answer 2. The expected answer type
Format your response as:
Thinking CoT: "topic_entity1" -> relation1 -> ? -> relation2 -> ... -> answer_type
OR for multiple topic entities:
Thinking CoT: "topic_entity1" -> relation1 -> ? -> relation2 -> ? <- relation3 <- "topic_entity2"
Note: Use "->" for forward relations (outgoing from entity) and "<-" for backward relations (incoming to entity).
Important: Your thinking CoT should try to include ALL the provided topic entities if possible. Connect them through the predicted relations to form a complete reasoning chain.

`In-Context Few-shot`

Q: {Question}
Topic Entities: {Topic Entities}
A:

D.2.2 Visual Path Selection

This prompt guides the vision-language model to analyze the visualized relation graph and identify the top-3 most promising reasoning paths that could lead to the answer.

Vision Path Selection Prompt Template

You are analyzing a knowledge graph visualization to find reasoning paths that can answer a question.
Graph Legend:

- Gold/Yellow boxes: Topic entities (starting points, labeled with names)
- Blue circles with numbers: Intermediate entity sets (Node 1, Node 2, etc.)
- Arrows: Relations connecting entities
  - Arrow pointing TO a node means the relation goes IN that direction
  - Arrow pointing FROM a node means the relation goes OUT from that direction

Your Task: Given the question and the graph image, identify the TOP 3 most promising paths that could lead to the answer.
Output Format: For each path, write in this format:
Path 1: TopicEntity1 -> relation1 -> Node X <- relation2 <- TopicEntity2 <- ...
Path 2: TopicEntity3 -> relation3 -> Node Y -> relation4 -> Node Z -> ...
Path 3: ...
Important:

- Always use the FULL relation names (e.g., "music.artist.genre" not just "genre"). Do not abbreviate or shorten relation names.
- Pay attention to the DIRECTION of arrows. Use "->" for edges pointing forward (outgoing) and "<-" for edges pointing backward (incoming).
- Do NOT output paths that do not exist in the graph. If you are unsure from the image, verify by checking the Graph Text Description.

Q: {Question}
Graph Text Description: {Text Description}
Thinking CoT: {Thinking CoT}
A:

D.2.3 Visual Path Selection Retry

When previously predicted paths fail SPARQL verification, this prompt asks the model to re-examine the graph and select alternative paths while avoiding the failed ones.

Vision Path Selection Retry Prompt Template

You are analyzing a knowledge graph visualization to find reasoning paths that can answer a question.
IMPORTANT: Previous attempts failed! The following paths were predicted before but could NOT be verified in the knowledge graph (they don't exist or have incorrect relation names):
{Failed Paths}
Please carefully re-examine the graph and select DIFFERENT paths. Make sure to:

1. Double-check that the relation names EXACTLY match what is shown in the graph
2. Verify the path connections actually exist in the Graph Text Description
3. Do NOT repeat any of the failed paths listed above

Graph Legend: (Same as Vision Path Selection Prompt)

Q: {Question}
Graph Text Description: {Text Description}
Thinking CoT: {Thinking CoT}
Based on the graph image and avoiding the failed paths, please identify the top 3 paths:

D.2.4 Text-Only Path Selection

This prompt serves as a text-only baseline variant that replaces the graph image with a textual edge list description, used in ablation studies to evaluate the contribution of visual graph analysis.

Vision Path Selection Text-Only Prompt Template

You are analyzing a knowledge graph to find reasoning paths that can answer a question.
Graph Structure: The graph contains:

- Topic entities (starting points): {Topic Entities}
- Intermediate nodes (entity sets): numbered as Node 1, Node 2, etc.
- Relations connecting these nodes

Edge List:
{Text Description}
Your Task: Given the question, identify the TOP 3 most promising paths that could lead to the answer.
Output Format:
Path 1: TopicEntity -> relation1 -> Node X -> relation2 -> Node Y -> ...
Path 2: ...
Path 3: ...

Q: {Question}
Thinking CoT: {Thinking CoT}
Please identify the top 3 paths:

D.2.5 Dynamic Exploration

This prompt directs the model to identify which graph nodes should be further explored in the next iteration, enabling the dynamic visual exploration mechanism.

Vision Exploration Prompt Template

Given a question, the current retrieved knowledge graph paths, and a summary of what has been found, your task is to identify which nodes or entities should be explored further to find the answer.

Q: {Question}
Current Retrieved Paths: {Current Paths}
Summarized Paths: {Summarized Paths}
Summary of Findings: {Summary}

Based on the above information, please select the nodes that are most likely to lead to the answer when explored further. These should be nodes that:

1. Appear in promising paths but need more connections explored
2. Are intermediate nodes that might connect to the answer
3. Are related to the predicted answer type

Please respond in the following format:
Exploration Required: {node1, node2, node3}

Reason: [Brief explanation of why these nodes should be explored]

D.2.6 Path Summarization

This prompt instructs the LLM to summarize the retrieved knowledge graph paths into concise chain-of-thought (CoT) reasoning chains that connect topic entities to potential answers.

Vision Summary CoT Prompt Template

Given a question, main topic entities, and retrieved knowledge paths, your task is to summarize the paths into a chain of thought (CoT) that connects topic entities to potential answers.
You must:

1. Use ONLY entities with IDs from the given paths
2. Do NOT skip steps - each hop must be explicitly shown
3. Simplify relation names for readability while keeping the meaning clear

Q: {Question}
Main Topic Entities: {Topic Entities}
Retrieved Paths: {Paths}

Output Format:
CoT1: {entity_id: name} - { <- relation <- } - {entity_id: name} - { -> relation -> } - {entity_id: name}
reason: [Brief explanation of how this chain answers the question]
If multiple CoT chains are needed:
CoT2: ...
reason: ...

`In-Context Few-shot`

Now summarize the paths for:

D.2.7 Single Path Correction

When a single predicted path fails SPARQL verification, this prompt asks the model to correct the path by checking relation names and arrow directions against the graph text description.

Single Path Correction Prompt Template

You predicted a path that could NOT be verified in the knowledge graph. The path may have incorrect relation names or wrong arrow directions.
Failed Path:
{Failed Path}
Your Task: Correct the path by checking it against the Graph Text Description below. Fix any issues with:

1. **Relation names**: Must EXACTLY match what appears in the Graph Text Description
2. **Arrow directions**:
   - Use "->" when the edge goes FROM source TO target (forward direction)
   - Use "<-" when the edge goes FROM target TO source (backward direction)
   - Check the Graph Text Description format: "A -> relation -> B" means A connects to B via relation

Graph Text Description:
{Text Description}
Important:

- Only output the CORRECTED path, nothing else
- The path format should be: Entity1 -> relation1 -> Node X <- relation2 <- Entity2
- If you cannot find a valid correction, output "INVALID"

Corrected Path:

D.2.8 Question Answering

This prompt asks the LLM to judge whether the currently retrieved paths provide sufficient evidence to answer the question, and if so, to produce the final answer.

Question Answering with Vision Paths Prompt Template

Given a main question, the predicted thinking chain, the retrieved knowledge graph paths, and the summarized CoT chain, determine if the paths are sufficient to answer the question.

Q: {Question}
Thinking CoT: {Thinking CoT}
Retrieved Paths: {Paths}
Summarized CoT: {Summarized CoT}

Your task:

1. If the paths are sufficient to answer the question, respond with {Yes} and provide the answer.
2. If not sufficient, respond with {No} and explain what information is missing.

Format:
response: {Yes} or {No}
answer: {your answer} (if Yes)
reason: [explanation]

E VLM Sensitivity

To check whether FoG's gains depend on a specific closed-source VLM, we replace GPT-4o with Qwen2.5-VL-72B while keeping the rest of the pipeline fixed. Table 8 shows that Qwen2.5-VL-72B achieves near-parity with GPT-4o on WebQSP (-0.9) and CWQ (-2.4), indicating that FoG is not tightly coupled to a single VLM. In WebQSP error inspection with GPT-3.5, 52% of failures come from the VLM missing the correct path in the visualized graph, 19% from the answer being absent from the coarsened subgraph, 25% from reasoning errors over otherwise correct paths, and 4% from the answer entity being absent from Freebase.

Table 8: Sensitivity to the underlying VLM.

VLM	WebQSP	CWQ
GPT-4o	**88.7**	**72.3**
Qwen2.5-VL-72B	87.8	69.9

F Full Efficiency Comparison

Table 9 reports the unmerged version of the efficiency comparison, separating the text-LLM and VLM invocations as well as their corresponding input tokens. Cells marked "–" are not reported by the corresponding baseline paper or were not measured in the same run.

G Visual Relation Graph Examples

In this section, we present concrete examples of the visual relation graphs generated by our framework. In the visualizations, topic entities are displayed as **yellow boxes** labeled with their names, equivalent entity sets are represented as **blue numbered circles** (e.g., Node 1, Node 2), and dynamically explored entities (introduced in subsequent iterations) are shown as **green boxes**. For relations, only the last attribute name is retained (e.g., *portrayed_in_films* instead of the full Freebase relation path) to improve readability.

G.1 WebQSP Examples

G.1.1 Example 1: Single-Iteration Graph

Figure 4 shows a visual relation graph from WebQSP with the topic entity "Coronation Street". The graph captures the 2-hop neighborhood relations around the topic entity. The hierarchical Dot layout clearly reveals the multi-hop structure: the first hop connects to entity sets via relations such as *episodes*, *regular_cast*, and *series*, while the second hop further extends through relations like *character*, *appeared_in_tv_program*, *actor*, and *starring_roles*. This compact visual representation enables the VLM to simultaneously observe all reachable paths from the topic entity and make informed path selections.

G.1.2 Example 2: Multi-Iteration Graph with Dynamic Exploration

Figures 5 and 6 illustrate the dynamic exploration process on a WebQSP question with the topic entity "Francis Bacon".

In the first iteration (Figure 5), the initial relation graph is constructed around the topic entity. The graph contains 9 entity set nodes connected by relations such as *gender*, *profession*, *founder_significant_follower*, *nationality*, and *films*. After analyzing this graph, the VLM identifies promising directions and selects entities for further exploration.

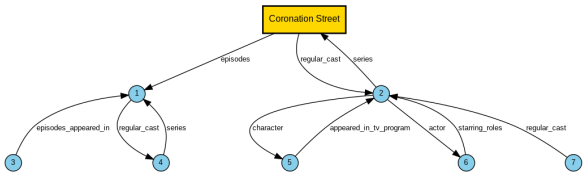

Figure 4: A visual relation graph from WebQSP with topic entity "Coronation Street". The graph shows the 2-hop relation neighborhood with 7 entity set nodes connected by relations related to TV series, episodes, and cast information.

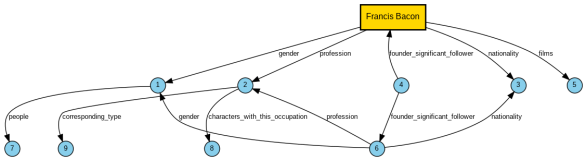

Figure 5: First iteration of a WebQSP example with topic entity "Francis Bacon". The initial relation graph presents relations related to profession, nationality, and followers.

In the second iteration (Figure 6), the dynamically explored entities "Scientist" and "England" are introduced as **green boxes**, expanding the graph with new relations such as *ethnicity*, *characters_of_this_ethnicity*, and *characters_based_on_this_person*. This demonstrates how FoG progressively enriches the relation graph by incorporating resolved entities from previous iterations, enabling deeper reasoning while maintaining a clear visual structure.

G.2 CWQ Examples

G.2.1 Example 1: Multi-Topic Entity Graph

Figure 7 shows a CWQ example involving two topic entities: "Country Nation World Tour" and "Bachelor's degree". The graph is notably more complex than the WebQSP examples, reflecting the multi-hop nature of CWQ questions. The two topic entities anchor separate subgraph regions connected through shared intermediate entity sets. Relations span diverse domains including music (*artist*, *concert_tours*, *concerts*), education (*degree*, *education*, *student*, *students_majoring*), and media (*album_or_release_supporting*). This

Table 9: Unmerged efficiency comparison between FoG and the synergy-augmented baselines ToG and PoG. *Text Calls* and *Vision Calls* are the average per-question invocations of the text LLM and the VLM, respectively; tokens are averaged similarly.

Dataset	Method	Text Calls	Vision Calls	Text Tokens	Vision Tokens	Time (s)
CWQ	ToG	22.6	–	8182.9	–	96.5
	PoG	13.3	–	7803.0	–	23.3
	FoG	**7.0**	**1.6**	**6208.6**	**1551.5**	**21.2**
WebQSP	ToG	15.9	–	6031.2	–	63.1
	PoG	9.0	–	5234.8	–	16.8
	FoG	**3.9**	**1.4**	**4234.9**	**1344.1**	**13.2**

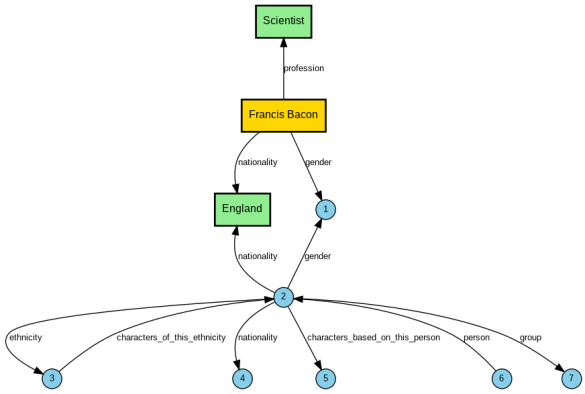

Figure 6: Second iteration of the same WebQSP example. Green boxes ("Scientist" and "England") represent dynamically explored entities that were resolved and added to expand the relation graph with new neighborhoods.

demonstrates the ability of visual relation graphs to present complex cross-domain reasoning structures in a single unified view.

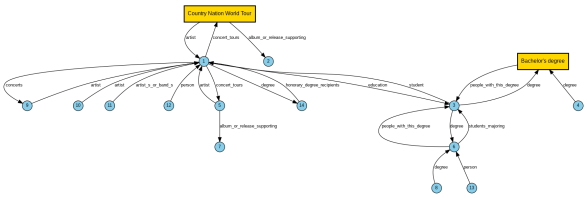

Figure 7: A visual relation graph from CWQ with two topic entities "Country Nation World Tour" and "Bachelor's degree". The graph captures complex multi-hop, cross-domain relations spanning music and education, with 14 entity set nodes.

G.2.2 Example 2: Multi-Iteration Graph with Dynamic Exploration

Figures 8 and 9 show the dynamic exploration process for a CWQ question with topic entities "Jenny's Father" and "Forrest Gump".

In the first iteration (Figure 8), the initial compact relation graph contains only 4 entity set nodes. Relations such as *portrayed_in_films*, *character*, *starring*, *film*, *film_casting_director*, and *films_casting_directed* capture the film-related connections between the two topic entities. The graph is concise and clearly shows the intersection structure between the two characters.

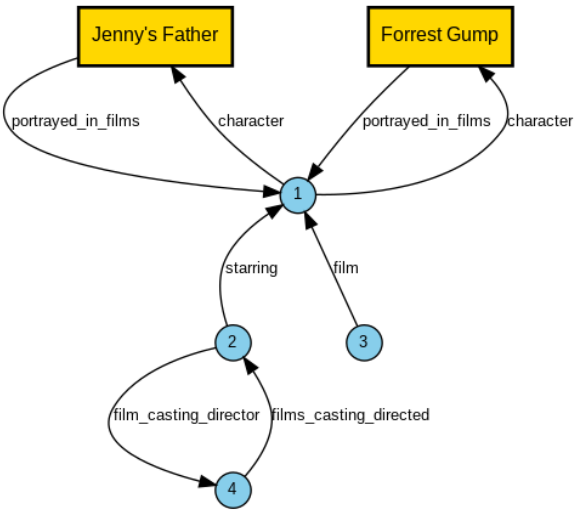

Figure 8: First iteration of a CWQ example with topic entities "Jenny's Father" and "Forrest Gump". The initial graph shows a compact structure with film-related relations connecting the two characters.

In the second iteration (Figure 9), the graph is significantly expanded with three dynamically explored entities shown as **green boxes** (labeled "Unnamed Entity", as they represent resolved entity sets whose specific names are displayed). New entity set nodes and relations are introduced, revealing deeper connections through additional *film*, *starring*, *character*, and *portrayed_in_films* relations. This expansion illustrates how the dynamic exploration mechanism progressively uncovers the

broader reasoning context needed for complex multi-hop questions.

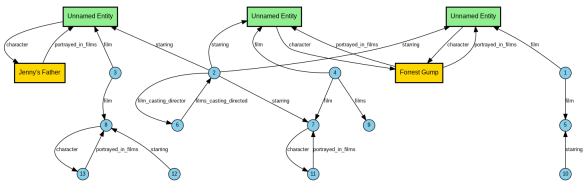

Figure 9: Second iteration of the same CWQ example. Three “Unnamed Entity” green boxes represent dynamically explored entities, significantly expanding the graph with deeper film-related connections and 13 entity set nodes.