

Guiding Graph: Explicit Query-Structure Guidance for Knowledge Graph Question Answering

Dehao Tao¹, Guangjie Liu², Jun Zhou¹, Junhao Chen¹

Wei Zheng¹, Rui Chen³, Junren Chen⁴

Yongfeng Huang^{1,*}, Minghu Jiang¹

¹Tsinghua University ²Xinjiang University

³China Railway Chengdu Group Co., Ltd.

⁴Sichuan University

tdh23@mails.tsinghua.edu.cn yfhuang@tsinghua.edu.cn

*Corresponding author

Abstract

Large Language Models (LLMs) can match local semantic clues in knowledge graphs (KGs), but complex KGQA often requires satisfying the full dependency structure of a question. Existing retrieval-based methods therefore risk producing reasoning paths that look plausible at each step but violate the query’s global intent. We propose **Guiding Graph**, a schema-light query-structure interface that turns KGQA from open-ended semantic traversal into structure-constrained query completion. A Guiding Graph exposes the entities, placeholders, and global dependencies that KG exploration must preserve, serving as an operational interface between natural language and graph traversal. The interface is training-free by default through prompt-based construction, and can optionally be strengthened with a cold-start fine-tuned constructor trained from limited bootstrapped supervision. Given the constructed Guiding Graph, exploration is governed by target selection from the query structure, structure-guided pruning before semantic matching, and target-scoped semantic alignment under the current Guiding Graph triple. Experiments on WebQSP, CWQ, and a domain-specific agricultural KG show that explicit query-structure guidance improves accuracy, robustness, and efficiency, with especially clear benefits for multi-hop reasoning and smaller LLMs.

1 Introduction

Grounding Large Language Models (LLMs) with knowledge graphs (KGs) has become a promising way to reduce hallucination and support knowledge-intensive question answering (Zhang et al., 2019; Yao et al., 2019; Wang et al., 2021; Luo et al., 2024). For complex KGQA, however, the retrieved facts must be not only semanti-

cally relevant, but also structurally consistent with the dependencies implied by the question. This requirement is especially important in multi-hop questions, where a locally plausible relation may lead to a path that violates another constraint in the query. As illustrated in Figure 1, the key failure is often not weak semantic matching alone, but the lack of an explicit global query structure that constrains local retrieval decisions.

Existing KG-grounded LLM reasoning methods expose a trade-off between precision, flexibility, and supervision. Executable query generation can be precise, but is sensitive to schema variation and parsing errors (Hu et al., 2023; Wang et al., 2023b). Iterative KG exploration is more flexible, but often makes locally semantic decisions that do not preserve the query’s global dependencies (Li et al., 2024; Zhao et al., 2023; Sun et al., 2024; Chen et al., 2024; Tao et al., 2025; He et al., 2025). Learned structural methods can encode such dependencies, but usually require substantial domain-specific supervision (Zhuang et al., 2024; Li et al., 2023b; Zhang et al., 2022; Mavromatis and Karypis, 2025). These trade-offs motivate an explicit query-structure interface that is operational enough to guide KG traversal, while remaining lightweight enough to transfer across schemas and domains.

We propose **Guiding Graph**, an explicit query-structure representation for KGQA. A Guiding Graph is a schema-light, machine-operable interface: it does not compile the question into a complete executable program, but exposes the entities, placeholders, and dependencies that KG exploration must preserve. With this interface, KGQA is reformulated from open-ended semantic traversal into structure-constrained query completion, where local retrieval decisions are scoped by the

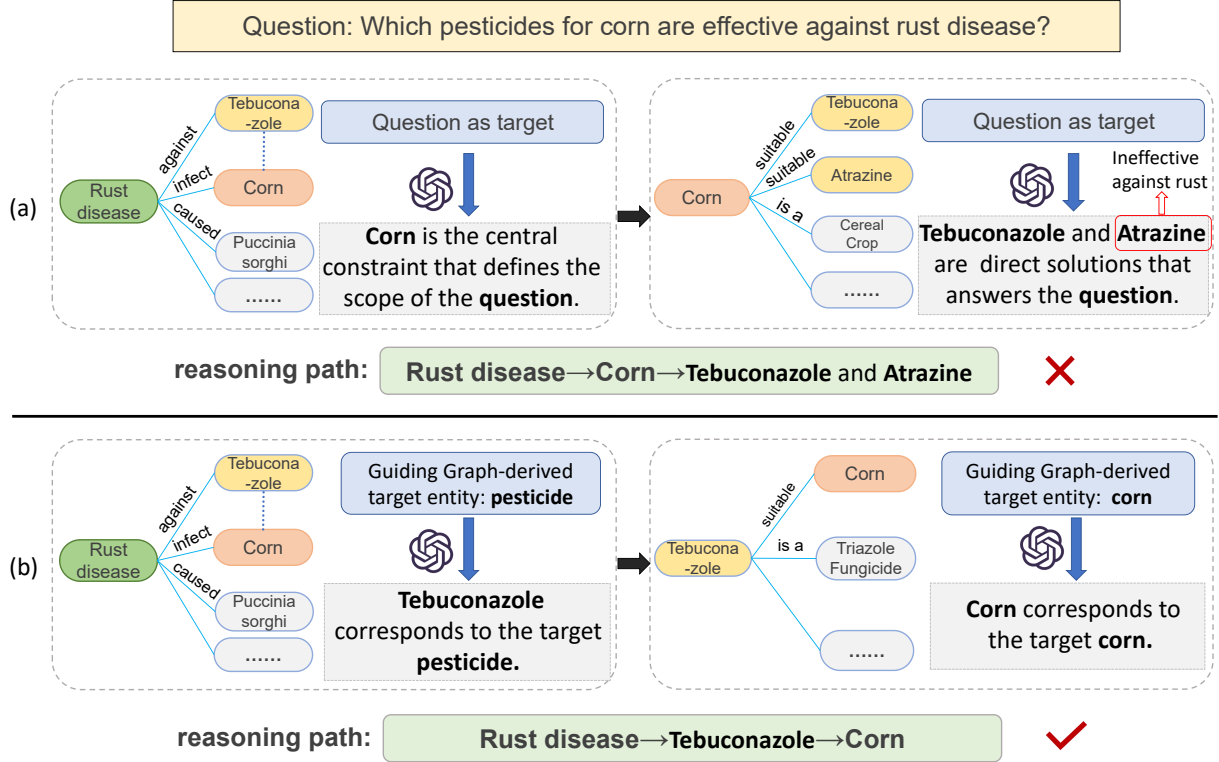


Figure 1: Motivating example comparing (a) whole-question semantic matching, which yields locally correct but globally inconsistent reasoning, and (b) step-wise matching guided by target constraints from the Guiding Graph, producing both local and global correctness. The displayed path illustrates the grounding/exploration process; the formal dependency constraints of the Guiding Graph are defined in Section 3.2.

global dependency pattern of the question.

Given a constructed Guiding Graph, our exploration algorithm maps it onto the KG through target-guided expansion. At each step, it determines the next target, prunes structurally invalid candidates before LLM scoring, and scopes semantic matching to the current Guiding Graph triple. This keeps local decisions compatible with the overall query structure, reducing semantic drift and cumulative multi-hop errors. The interface is training-free by default, and can optionally be made more robust through cold-start fine-tuning with limited bootstrapped supervision.

The contributions of this paper are as follows:

1. We introduce **Guiding Graph**, a schema-light query-structure interface for structurally consistent KGQA. It is training-free by default and can be optionally strengthened with a cold-start fine-tuned constructor.
2. We design a structure-constrained KG exploration algorithm that maps the Guiding Graph onto the KG through target selection, early structural pruning, and target-scoped semantic alignment.

3. We evaluate on WebQSP, CWQ, and a domain-specific agricultural KG, demonstrating gains in accuracy, robustness, and efficiency, especially for multi-hop reasoning and smaller LLMs.

2 Related Work

2.1 LLM Reasoning with Prompt

Structured prompting improves LLM reasoning by decomposing complex problems into intermediate states. Representative methods include Chain-of-Thought (Wei et al., 2022), decomposition-based prompting (He et al., 2021), Tree-of-Thought (Yao et al., 2023), Graph-of-Thought (Besta et al., 2024), Plan-and-Solve (Wang et al., 2023a), self-consistency (Wang et al., 2023c), Least-to-Most prompting (Zhou et al., 2023), and Program-Aided Reasoning (Gao et al., 2023). These methods organize reasoning in natural language or program-like traces, but their intermediate states are not necessarily tied to KG entities or reusable graph constraints. Our work inherits the benefit of explicit intermediate structure while making that structure directly operational for KG exploration.

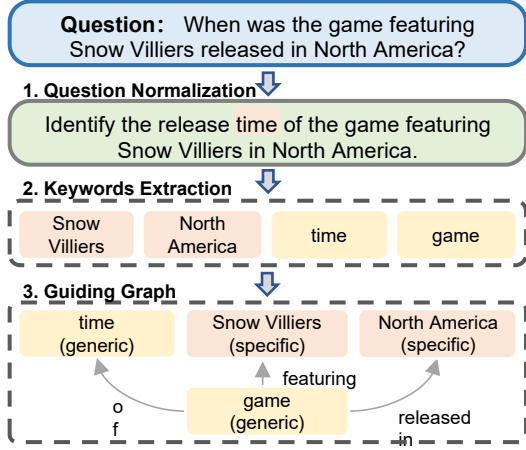


Figure 2: Guiding Graph Construction Workflow

2.2 Question Answering with KG-Augmented LLM

LLM-KG integration methods either inject KG structure during training (Zhang et al., 2019; Liu et al., 2020; Zhuang et al., 2024; Li et al., 2023b) or use KGs dynamically at inference time. Inference-time approaches include prompt- or query-based use of KG facts (Li et al., 2023a; Baek et al., 2023), retrieval-augmented systems (Wang et al., 2023b; Hu et al., 2023), graph-retrieval methods (Mavromatis and Karypis, 2025; Long et al., 2025; Sui et al., 2025), and KG-agent frameworks that iteratively explore graph neighborhoods, such as ToG (Sun et al., 2024), PoG (Chen et al., 2024), FiSKE (Tao et al., 2025), FastToG (Liang and Gu, 2025), and ReKG-MCTS (Song et al., 2025). These systems demonstrate the value of grounding LLMs in external graphs, but many still rely on locally selected relations or learned retrieval structures during traversal. Guiding Graph follows the inference-centric, structure-aware line of KGQA, while introducing a schema-light query-structure interface that constrains traversal without compiling the whole question into a target-schema-specific program.

3 Method

3.1 Task Definition

We study Knowledge Graph Question Answering (KGQA) with *explicit global structural guidance*. Given a natural language question Q and a knowledge graph $\mathcal{KG} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, the goal is to retrieve a query-specific set of KG triples that can ground answer generation.

From Q we construct a *Guiding Graph* $GG =$

$(\mathcal{V}_{GG}, \mathcal{E}_{GG})$. Its nodes are entities or placeholders mentioned in the question, and its edges encode desired relations and dependencies among them. KG exploration grounds the Guiding Graph by mapping its nodes to KG entity candidates and aligning its edges with semantically compatible KG triples.

The key constraint is *structural consistency*: if multiple Guiding Graph edges share a node, their grounded KG triples must share the corresponding grounded entity or candidate set. The output is a query-specific grounded subgraph, represented as a set of KG triples, which is provided to the LLM for final answer generation.

3.2 Guiding Graph Construction

Guiding Graph construction converts a natural-language question into the schema-light query-structure interface used by exploration. The interface provides step-wise target constraints and encodes dependency patterns rather than KG-specific predicates, allowing the same representation to transfer across domains with different relation names or schemas. We use prompt-based construction as the default training-free instantiation, with an optional cold-start fine-tuned variant for improved construction robustness. Figure 2 illustrates the construction process.

Prompt-Based Construction. The default constructor uses an LLM without task-specific training. It hierarchically processes the input question in three stages: 1. transform elliptical or under-specified questions into complete declarative statements; 2. extract fine-grained elements and categorize them into *specific keywords*, which usually denote named entities, and *generic keywords*, which usually denote concepts or relation clues; 3. infer semantic relations among the extracted keywords to capture the dependencies needed for constrained exploration. To instantiate Guiding Graph triples, we map specific keywords to entity nodes and use generic keywords as relation or placeholder cues, resolving co-reference and ambiguous generic terms according to the extracted dependencies. The final output is a list of Guiding Graph triples in the form $(subject, relation, object)$, where nodes may be specific entities or generic placeholders. Full operational heuristics and prompt templates are provided in Appendix A7 and Appendix A8.

Optional Cold-Start Fine-Tuned Construction.

The same interface can optionally be optimized to reduce prompt-chaining errors in keyword classification, intermediate formatting, or graph construction. Instead of requiring a separately annotated dataset, we use the training-free system itself as a cold-start data generator. Using only the training split, we run the full training-free pipeline and select 250 questions for which the system returns the correct answer; their generated Guiding Graphs are treated as high-confidence training targets. We further manually correct 50 failed cases, resulting in 300 question–Guiding Graph training pairs.

Using this bootstrapped data, we merge the information-extraction and graph-construction prompts into a single instruction and fine-tune an LLM to map each question directly to the final Guiding Graph triples while keeping the output format unchanged. The default and optional constructors therefore share the same schema and feed into the same KG exploration algorithm, so the framework remains training-free by default while offering a self-bootstrapping path when higher construction robustness is desired.

3.3 Knowledge Exploration

We design the knowledge exploration process around the Guiding Graph to ensure that every step adheres to the *global structural constraints* derived from the query. As illustrated in Figure 3, entities and relations in the Guiding Graph are progressively mapped to their counterparts in the KG through iterative, target-guided expansion. This process incrementally builds a KG subgraph that is both structurally consistent with the Guiding Graph and semantically aligned with the original query. The exploration loop repeats three operations until all reachable Guiding Graph nodes are grounded or no valid expansion remains: it selects an ungrounded target adjacent to the current grounded frontier, constructs a structurally valid candidate set for that target, and uses target-scoped semantic alignment to update the grounding dictionary and collected triples. The prompts used are shown in Table A9 A10 A11.

3.3.1 Initialization and Iterative Target Selection

Under the principle of **Global Structural Constraint**, initialization anchors the exploration in the global query structure encoded in the Guiding Graph. We first take all entities $e \in \mathcal{V}_{GG}^{\text{spec}}$

originating from specific keywords (Section 3.2) and perform string-based retrieval in the KG. Any matched KG entities are recorded in the initial mapping dictionary:

$$\mathcal{M} = \{e \in \mathcal{V}_{GG}^{\text{spec}} \mapsto \text{KG entities}\},$$

where $\mathcal{M}(e)$ denotes the set of KG entities currently associated with e . Restricting starting points to this subset avoids the pitfall that generic keywords often map to concept-type KG nodes, which can mislead subsequent reasoning. The dictionary $\mathcal{M}$ is then progressively expanded during exploration: in each round, the current target entity selected from the Guiding Graph is resolved to one or more KG entities and added to $\mathcal{M}$.

Exploration begins from the specific-origin entities already anchored in $\mathcal{M}$ and proceeds iteratively. In each iteration, the target entity $target_e \in \mathcal{V}_{GG}$ is determined by finding a triple pattern in the Guiding Graph of the form $(current_e, r, target_e)$, where $current_e \in \mathcal{V}_{GG}$ has been mapped in $\mathcal{M}$ and $target_e$ has not. Here r denotes the relation connecting $current_e$ and $target_e$ in the Guiding Graph. This ensures expansion only occurs from known anchors toward unexplored nodes, confining the search to subspaces defined by the connected structure of the Guiding Graph and preserving both local and global consistency.

3.3.2 Structure-Guided Early Pruning

In each exploration round, the selected $target_e$ inherits global structural constraints from the Guiding Graph through its mapped neighbors. These constraints define the structural subspace in which the next grounding must lie. We therefore perform *model-free* pruning before semantic evaluation, removing KG candidates that violate this structural pattern. This pre-semantic filtering reduces noise and LLM cost while increasing the likelihood that later semantic alignment remains globally coherent.

Let $\Gamma_{KG}(x)$ denote the neighborhood function returning KG entities adjacent to a given KG entity x . In implementation, this neighborhood respects the direction of the corresponding Guiding Graph edge; we omit direction in the notation for simplicity. Let $\mathcal{N}_{GG}^{\text{mapped}}(target_e)$ be the set of Guiding Graph neighbors of $target_e$ that have been mapped to one or more KG entities in $\mathcal{M}$. When a neighbor rel_e in this set maps to multiple KG entities $\mathcal{M}(rel_e) = \{m_1, m_2, \dots\}$, we first

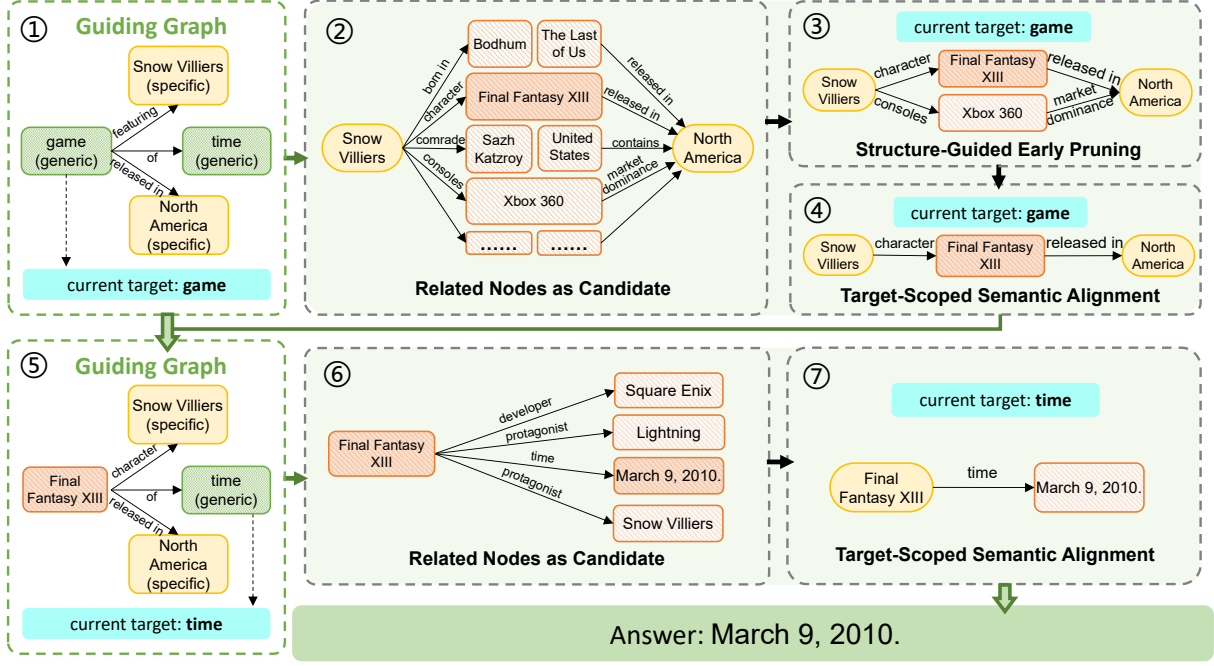


Figure 3: Iterative knowledge exploration with Guiding Graph: progressively mapping the Guiding Graph to the KG for question answering.

take the **union** of their KG neighborhoods as the structural support for that neighbor:

$$\mathcal{U}(\text{rel}_e) = \bigcup_{m \in \mathcal{M}(\text{rel}_e)} \Gamma_{\text{KG}}(m).$$

The structure-guided candidate set is then obtained by taking the **intersection** over all such neighbor unions:

$$\mathcal{E}'_{\text{cand}} = \bigcap_{\text{rel}_e \in \mathcal{N}_{\text{GG}}^{\text{mapped}}(\text{target}_e)} \mathcal{U}(\text{rel}_e).$$

This formulation implicitly includes *current_e* as one of the mapped neighbors. The pruning step is designed for scalability in large KGs, where high-degree entities can produce hundreds or thousands of adjacent candidates. Because the filtering is performed through set operations before any LLM scoring, it avoids applying model-based semantic ranking to the full neighborhood and reduces the need for heuristic candidate truncation. If the final intersection $\mathcal{E}'_{\text{cand}}$ is empty, which may occur when KG incompleteness prevents strict structural alignment, we trigger *dynamic branch selection*. In this fallback, each $\mathcal{U}(\text{rel}_e)$ is treated as an alternative structurally supported branch, and the LLM selects the branch whose context is most relevant to the current Guiding Graph triple. The selected branch is retained for continued exploration, while

the others are pruned. Thus the pruning is conditionally complete with respect to the currently mapped Guiding Graph constraints: any candidate that satisfies all mapped-neighbor constraints remains in the intersection. Dynamic branch selection is used only when the strict intersection is empty, providing a fallback rather than arbitrary termination.

3.3.3 Target-Scoped Semantic Alignment

Existing traversal methods often perform semantic pruning in multiple stages, such as relation matching followed by entity-level matching. Without an explicit target context, such pruning is applied to large candidate pools and may discard valid nodes early, propagating errors to later hops. In contrast, each exploration round in our method is scoped by the Guiding Graph triple (*current_e*, *r*, *target_e*): $\mathcal{M}(\text{current}_e)$ is already grounded, while *r* and *target_e* jointly define the semantic target to resolve. After structure-guided pruning, this context allows us to align relations in a single step over a compact candidate set.

Concretely, given $\mathcal{E}'_{\text{cand}}$, we gather KG relations $\mathcal{R}$ connecting any $e' \in \mathcal{E}'_{\text{cand}}$ to $\mathcal{M}(\text{current}_e)$. We then prompt the LLM to directly identify the (*current_e*, *can_r*, *target_e*) triple whose relation *can_r* $\in \mathcal{R}$ best matches

($current_e, r, target_e$) from the Guiding Graph. Let $\hat{r}$ be the chosen relation; we update the mapping as:

$$\mathcal{M}(target_e) = \mathcal{E}'_{cand} \cap \Gamma_{\hat{r}}(\mathcal{M}(current_e)),$$

where $\Gamma_{\hat{r}}(x)$ denotes the set of KG entities connected to x via relation $\hat{r}$. When its input is an entity set, $\Gamma_{\hat{r}}$ is applied to each entity and the results are unioned. Since the relation space is much smaller than the entity space, and the candidate entities have already been structurally filtered, this target-scoped relation alignment is both efficient and less prone to semantic drift.

Multiple parallel branches may arise from multi-hop combinations of nodes. Branches that fail to yield any new candidate entities in the current iteration—thus breaking the structural connectivity prescribed by the Guiding Graph—are removed from the search space, ensuring that subsequent exploration focuses on structurally feasible mappings.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation Metrics. We evaluate the proposed paradigm using two external knowledge bases: the open-source Freebase (Bollacker et al., 2008) and a self-constructed domain-specific knowledge graph. Details of the latter, referred to as the Agricultural Knowledge Graph, are provided in Appendix A.2.

Freebase provides a large and relation-rich testbed for knowledge-dense KGQA. We conduct experiments on two QA sets with Freebase as the external knowledge base: WebQSP (Yih et al., 2016) and CWQ (Talmor and Berant, 2018). For CWQ, we follow the commonly adopted LLM-based KGQA evaluation protocol used by ToG, PoG, and related work. In this protocol, each evaluation instance contains a single string-valued answer, and the metric is exact match / Hits@1. Consequently, partial and complete matching are equivalent for CWQ under this protocol.

We evaluate multi-answer questions through two distinct metrics: (1) partial match, where success requires retrieval of at least one correct answer, and (2) complete match, which necessitates identification of all correct answers. For single-answer evaluation instances, these two metrics reduce to the same criterion.

Baselines. We compare our method with a variety of approaches, including two knowledge-free prompting methods (IO (Brown et al., 2020), CoT (Wei et al., 2022)); four finetuned KB-QA models (RoG (Luo et al., 2024), GCR (Luo et al., 2025), GNN-RAG (Mavromatis and Karypis, 2025), EPERM (Long et al., 2025)); and several non-finetuned knowledge-enhanced methods such as FastToG (Liang and Gu, 2025), ReKG (Song et al., 2025), StructGPT (Jiang et al., 2023), KB-BINDER (Li et al., 2023a), ReKnoS (Wang et al., 2025), FiDeLis (Sui et al., 2025), ToG (Sun et al., 2024), PoG (Chen et al., 2024), and FiSKE (Tao et al., 2025). Table 1 combines quoted numbers from prior work for broad context with controlled reruns for the closest graph-exploration baselines under the same evaluation protocol.

4.2 Performance Comparison

The results of WebQSP and CWQ are shown in Table 1. On WebQSP, our approach with Llama3-8B-Instruct substantially outperforms the graph-exploration baselines evaluated with the same model, achieving performance competitive with methods that rely on much larger LLMs. This highlights a practical advantage of explicit query-structure guidance: it reduces the burden on the LLM by converting broad semantic exploration into constrained structural completion. With DeepSeek-V3 (DeepSeek-AI, 2024), our method achieves competitive partial-match performance and the best complete-match result among the compared non-finetuned methods, suggesting that structure-guided exploration is especially helpful for retrieving comprehensive answer sets.

Compared to WebQSP, CWQ contains more compositional multi-hop questions. On CWQ, our method achieves the strongest result among the compared non-finetuned methods and improves substantially over the controlled graph-exploration baselines PoG and FiSKE. This pattern supports our central claim: when the question contains nontrivial dependencies, explicitly preserving the query structure is more reliable than selecting locally plausible triples from fine-grained clues alone.

Overall, the gains are most pronounced in settings where structural consistency matters: complete answer retrieval, compositional questions, and smaller LLMs that benefit from a narrower and better constrained search space.

Method	WebQSP		CWQ
	partial	complete	Hits@1
<i>without external knowledge</i>			
IO prompt (Brown et al., 2020) w/DeepSeek-V3	63.3	35.3	44.8
COT (Wei et al., 2022) w/DeepSeek-V3	70.5	41.4	46.7
<i>with finetuning</i>			
RoG (Luo et al., 2024) w/Llama2-7B	85.7	–	62.6
GCR (Luo et al., 2025) w/Llama3.1-8B+ChatGPT	92.6	–	72.7
GNN-RAG (Mavromatis and Karypis, 2025) w/Llama2-7B	85.7	–	66.8
EPERM (Long et al., 2025) w/Llama2-7B	88.8	–	66.2
<i>without finetuning</i>			
FastToG (Liang and Gu, 2025) w/GPT-4o-mini	65.8	–	45.0
FastToG (Liang and Gu, 2025) w/Llama3-70B	66.4	–	46.2
ReKG (Song et al., 2025) w/Llama3-8B	72.2	–	59.8
ReKG (Song et al., 2025) w/Llama3-70B	78.9	–	64.4
StructGPT (Jiang et al., 2023) w/GPT-3.5	72.6	–	54.3
KB-BINDER (Li et al., 2023a) w/GPT-3.5	74.4	–	–
ReKnos (Wang et al., 2025) w/GPT-3.5	81.1	–	58.5
FiDeLis (Sui et al., 2025) w/GPT-4-turbo	84.4	–	71.5
ToG (Sun et al., 2024) w/Llama3-8B	55.6	32.3	–
ToG (Sun et al., 2024) w/GPT-3.5	76.2	–	57.1
PoG (Chen et al., 2024) w/Llama3-8B	63.4	34.4	–
PoG (Chen et al., 2024) w/DeepSeek-V3	80.9	60.7	55.7
FiSKE (Tao et al., 2025) w/Llama3-8B	70.8	40.4	–
FiSKE (Tao et al., 2025) w/DeepSeek-V3	82.5	61.1	50.2
Ours w/Llama3-8B	79.3	54.1	56.7
Ours w/DeepSeek-V3	81.8	64.5	71.8

Table 1: Results for WebQSP and CWQ. Bold results indicate the best performance among methods without finetuning.

Dataset	Method	LLM Call	Input Token	Output Token	Total Token
WebQSP	PoG	11.3	6590.2	427.0	7017.2
	FiSKE	7.9	3079.0	1379.7	4458.7
	Ours	8.6	3264.5	584.0	3848.5
CWQ	PoG	23.4	15483.4	694.8	16178.2
	FiSKE	9.4	3578.8	1828.7	5407.5
	Ours	10.2	4052.0	708.6	4760.6

Table 2: Efficiency Comparison Across Datasets.

4.3 Computational Cost

We conducted an efficiency study of our method, with results shown in the Table 2. Overall, our analysis reveals that our approach demonstrates comprehensive advantages over PoG in both the number of LLM calls and total token consumption. On WebQSP (dominated by single-hop questions), our method shows modest improvements, while on CWQ (containing primarily multi-hop questions), it achieves significant gains, highlighting its superior capability in handling complex problems. We attribute this to PoG’s reliance on heavily constrained prompts with strict rules to govern LLM outputs, whereas our approach embeds constraints primarily within the designed strategies rather than the prompts. This distinction may explain why our method is more friendly to smaller LLMs.

4.4 Construction Quality and Robustness

We next examine the Guiding Graph as a structural interface. Table 3 evaluates whether optional constructor refinement can improve this interface without changing downstream exploration. The refinement uses 250 high-confidence Guiding Graphs produced by the training-free pipeline on the training split, plus 50 manually corrected failed cases. Using Qwen3-8B for the full pipeline improves from 60.6 to 69.4 after refinement, showing that bootstrapped structural supervision benefits smaller models. In the hybrid setting, using the refined constructor only for Guiding Graph construction while keeping DeepSeek-V3 for exploration and answering improves performance from 71.8 to 73.7. This supports our design choice that construction quality can be enhanced as an optional robustness mechanism while the explo-

Setting	GG Constructor	Exploration/Answering	Hits@1
Training-free pipeline	DeepSeek-V3 prompting	DeepSeek-V3	71.8
Unfine-tuned Qwen3-8B pipeline	Qwen3-8B prompting	Qwen3-8B	60.6
Fully fine-tuned pipeline	Fine-tuned Qwen3-8B	Fine-tuned Qwen3-8B	69.4
Hybrid pipeline	Fine-tuned Qwen3-8B	DeepSeek-V3	73.7

Table 3: Effect of optional constructor refinement on CWQ.

ration algorithm remains unchanged. This separation indicates that the Guiding Graph constructor can be improved independently of the LLM used for graph exploration and answering. This suggests that errors in the query-structure interface, rather than only downstream KG exploration, are a meaningful source of failure. Fine-tuning details are provided in Appendix A.5. In addition to these end-task and perturbation analyses, Appendix A.3 reports a direct manual audit of generated Guiding Graphs, judging construction quality independently of final QA accuracy.

We further test robustness to imperfect Guiding Graphs through controlled perturbations. The perturbations replace specific entities or generic concepts with synonyms, insert a perturbed duplicate triple, or remove a peripheral triple. Detailed results are reported in Table 4. The results show that generic-concept substitutions have little impact, while errors affecting specific anchors or removing triples cause larger drops, consistent with the role of anchors and structural connectivity in our exploration process. This is expected because specific entities seed the initial KG retrieval, whereas generic concepts are later interpreted under target-scoped semantic alignment. Triple removal is more harmful because it can permanently block an otherwise valid reasoning path.

Variant	Partial Match
Original Guiding Graph	71.8
Specific-Entity Substitution	66.5
Generic-Concept Substitution	71.1
Triple Insertion	68.9
Triple Removal	63.2

Table 4: Ablation study on the robustness of the Guiding Graph.

4.5 Studies on starting points

We evaluated the impact of starting point retrieval on our method’s performance (Table 5). When no specific keyword can be linked to a KG entity, our method falls back to a direct LLM response rather than forcing unreliable graph exploration.

Filtered-CWQ removes such cases and therefore estimates performance when the structural interface has valid KG anchors. The improvement on Filtered-CWQ shows that starting-point quality remains an important practical factor; it also separates failures caused by entity linking from failures of the structure-guided exploration algorithm itself.

4.6 Ablation Studies

We conducted ablation experiments (Table 6) to evaluate the effects of *Structure-Guided Early Pruning* (with dynamic branch selection) and *Target-Scoped Semantic Alignment*.

For structural alignment, we tested (1) removing the module entirely, and (2) disabling dynamic branch selection. For semantic alignment, rich target phrases were replaced with single clues. Results show that reduced semantic context markedly lowers performance, confirming the role of precise semantics in pruning. Removing structural alignment decreases both accuracy and efficiency, while disabling dynamic selection slightly cuts tokens but costs accuracy—justifying the small overhead. In summary, the ablation experiments confirm the contribution of each component to the robustness and efficiency of the final model.

5 Conclusion

We presented **Guiding Graph**, a schema-light query-structure interface for KGQA. By exposing the entities, placeholders, and dependencies that KG exploration must preserve, Guiding Graph turns open-ended semantic traversal into structure-constrained query completion. The exploration algorithm maps this interface onto the KG through target selection, structure-guided early pruning, and target-scoped semantic alignment, keeping local decisions compatible with the global query structure. The framework is training-free by default through prompt-based construction, with an optional cold-start fine-tuned constructor for improved construction robustness. Experiments on WebQSP, CWQ, and a domain-specific agricul-

Dataset	Method	LLM Call	Input Tok.	Output Tok.	Total Tok.	Partial	Complete
CWQ	Ours	10.2	4052.0	708.6	4760.6	0.7179	0.7179
Filtered-CWQ	Ours	12.0	4875.4	769.9	5645.3	0.7460	0.7460

Table 5: Experimental results regarding starting points. In Filtered-CWQ, questions without identifiable starting points were removed. In our approach, such questions are directly answered by the LLM.

Variant	partial match	LLM Call	Input Token	Output Token	Total Token
Ours	71.8	10.2	4052.0	708.6	4760.6
w/o Target-Scoped Semantic Alignment	53.8	7.9	2908.5	591.1	3499.6
w/o Structure-Guided Early Pruning	67.9	11.6	4761.9	829.5	5591.4
w/o Dynamic Branch Selection	69.2	10.0	3983.8	701.4	4685.2

Table 6: Ablation Studies on CWQ set.

tural KG demonstrate gains in accuracy, robustness, and efficiency, with particular improvements in multi-hop reasoning and smaller-LLM settings. These results suggest that explicit query-structure interfaces are a useful abstraction for connecting LLM reasoning with structured external knowledge.

Limitations

The quality of the Guiding Graph remains important because construction errors can propagate to downstream exploration. The prompt-based constructor preserves the fully training-free setting, but may still suffer from entity extraction or dependency prediction errors. The cold-start fine-tuned constructor reduces this issue with only limited manual correction, but it still depends on the availability of training questions and answer supervision to identify high-confidence cases. Future work will investigate self-correction and uncertainty estimation mechanisms that further improve construction quality with less manual intervention.

Acknowledgement

This work was supported by the Henan Provincial Major Industrial Key Technology Research Open Bidding for Leading Talents Project (251000210400). This work was also supported by the Scientific Research Project of China Railway Chengdu Group Co., Ltd. in 2025 (CX25050).

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. [Knowledge-augmented language model prompting](#)

[for zero-shot knowledge graph question answering](#). In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106, Toronto, Canada. Association for Computational Linguistics.

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

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 B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

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*, volume 37, pages 37665–37691.

DeepSeek-AI. 2024. [DeepSeek-V3 technical report](#). *arXiv preprint arXiv:2412.19437*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [PAL: Program-aided language models](#). In *Proceedings of the 40th International*

- Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.
- 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*, pages 553–561.
- Jiashu He, Mingyu Ma, Jinxuan Fan, Dan Roth, Wei Wang, and Alejandro Ribeiro. 2025. [GIVE: Structured reasoning of large language models with knowledge graph inspired veracity extrapolation](#). In *Proceedings of the 42nd International Conference on Machine Learning*.
- Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. ChatDB: Augmenting LLMs with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*.
- Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. [StructGPT: A general framework for large language model to reason over structured data](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, Singapore. Association for Computational Linguistics.
- Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhui Chen. 2023a. [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, Toronto, Canada. Association for Computational Linguistics.
- Xin Li, Dongze Lian, Zhihe Lu, Jiawang Bai, Zhibo Chen, and Xinchao Wang. 2023b. GraphAdapter: Tuning vision-language models with dual knowledge graph. *Advances in Neural Information Processing Systems*, 36:13448–13466.
- Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. [Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources](#). In *The Twelfth International Conference on Learning Representations*.
- Xujian Liang and Zhaoquan Gu. 2025. [Fast think-on-graph: Wider, deeper and faster reasoning of large language model on knowledge graph](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24558–24566.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2901–2908.
- Xiao Long, Liansheng Zhuang, Aodi Li, Minghong Yao, and Shafei Wang. 2025. [EPERM: An evidence path enhanced reasoning model for knowledge graph question and answering](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12282–12290.
- Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *ICLR 2024: The Twelfth International Conference on Learning Representations*. ICLR.
- Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. 2025. [Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models](#). In *Proceedings of the 42nd International Conference on Machine Learning*, pages 41540–41565. PMLR.
- 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*, pages 16682–16699.
- Xiaozhuang Song, Shufei Zhang, and Tianshu Yu. 2025. [ReKG-MCTS: Reinforcing LLM reasoning on knowledge graphs via training-free monte carlo tree search](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9288–9306, Vienna, Austria. Association for Computational Linguistics.
- Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. 2025. [FiDeLiS: Faithful reasoning in large language models for knowledge graph question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8315–8330, Vienna, Austria. Association for Computational Linguistics.
- 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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.
- Dehao Tao, Congqi Wang, Feng Huang, Junhao Chen, Yongfeng Huang, and Minghu Jiang. 2025. [Fine-grained stateful knowledge exploration: Effective and efficient graph retrieval with large language models](#). *Knowledge-Based Systems*, 326:114011.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim.

- 2023a. [Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.
- Song Wang, Junhong Lin, Xiaojie Guo, Julian Shun, Jundong Li, and Yada Zhu. 2025. [Reasoning of large language models over knowledge graphs with super-relations](#). In *The Thirteenth International Conference on Learning Representations*.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. [KEPLER: A unified model for knowledge embedding and pre-trained language representation](#). *Transactions of the Association for Computational Linguistics*, 9:176–194.
- Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023b. [KnowledGPT: Enhancing large language models with retrieval and storage access on knowledge bases](#). *arXiv preprint arXiv:2308.11761*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 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. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). *Advances in neural information processing systems*, 36:11809–11822.
- Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. [The value of semantic parse labeling for knowledge base question answering](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.
- Minhao Zhang, Ruoyu Zhang, Yanzeng Li, and Lei Zou. 2022. [Crake: Causal-enhanced table-filler for question answering over large scale knowledge base](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1787–1798, Seattle, United States. Association for Computational Linguistics.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. [ERNIE: Enhanced language representation with informative entities](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.
- Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. [Verify-and-edit: A knowledge-enhanced chain-of-thought framework](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). In *The Eleventh International Conference on Learning Representations*.
- Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W. Huang, Jie Fu, Xiang Yue, and Wenhui Chen. 2024. [StructLM: Towards building generalist models for structured knowledge grounding](#). *Preprint*, arXiv:2402.16671.

A Appendix

A.1 LLM Usage Statement

The large language model was used solely for grammar checks and polishing, and no other purposes.

A.2 Experiment of Agricultural

The agricultural knowledge graph we constructed contains over 500,000 entities and 1.4 million triples. Compared to general-purpose knowledge graphs like Freebase, while the number of entities is relatively modest, its relational structure remains highly complex. Within agricultural knowledge graphs, diverse relationships can exist between different nodes, allowing for multiple possible pathways between the same pair of nodes. In contrast, general knowledge graphs often exhibit a simpler structure, where there might be only a single pathway between different nodes. This difference in relationship variety leads to varying levels of structural complexity. Based on this, we developed both single-hop and multi-hop question-answering datasets. These datasets have been released on the National Agriculture Science Data Center platform and are publicly accessible for research use.

The experimental results on the agricultural dataset are presented in the Table A1. For fair comparison, we equipped PoG with the same fine-grained information extraction method as ours, since its original implementation relies on dataset-provided topic entities which are unavailable in agricultural QA sets. Our method demonstrates significant improvements over existing approaches, particularly for multi-hop question answering, where the advantage reaches remarkable levels. A notable observation reveals that while existing methods show substantial performance gaps between single-hop and multi-hop QA tasks, our approach consistently maintains high performance across both scenarios. This demonstrates our method’s superior capability in handling complex problems within knowledge-intensive graphs, as also evidenced in Table 1.

The efficiency comparison is shown in Table A2, on the agricultural dataset, our method requires slightly fewer LLM calls than PoG, while the token consumption by the LLM is significantly reduced. FiSKE demonstrates high efficiency on WebQSP and CWQ, but incurs significant computational overhead when handling the structurally

complex agricultural dataset. In contrast, our approach consistently achieves stable and superior performance across all datasets.

A.3 Manual Audit of Guiding Graph Construction

To directly inspect the construction quality of Guiding Graphs independently of final QA accuracy, we manually audited 50 randomly sampled CWQ cases. Each generated Guiding Graph was labeled as *Good*, *Partial*, or *Bad* according to whether it preserved the intended compositional structure and relation constraints.

As shown in Table A3, 92% of the sampled cases preserve at least the main compositional skeleton. The main remaining construction errors involve temporal or superlative constraints, answer-type selection, and occasional over-specific entity identification. Table A4 provides representative examples.

A.4 Same-Backbone Stability Check

To complement the main comparison with a controlled run-to-run stability check, we evaluated ToG, PoG, FiSKE, and our method with the same DeepSeek-V3 backbone on the same random subset of 100 questions for three runs. As shown in Table A5, the advantage of our method is substantially larger than the observed run-to-run variance.

A.5 Fine-Tuning Details for Guiding Graph Construction

We fine-tune the unified Guiding Graph constructor with supervised fine-tuning using LlamaFactory. The base model is Qwen3-8B, and the output target is the final Guiding Graph triple sequence used by the exploration module. We use LoRA rather than full-parameter fine-tuning to keep the adaptation lightweight. The main training configuration is summarized in Table A6. For multi-GPU training, we launch the run through the distributed launcher by enabling `FORCE_TORCHRUN=1`; for single-GPU training, we set `CUDA_VISIBLE_DEVICES=0`.

We evaluate two ways of using the fine-tuned constructor. Before fine-tuning, Qwen3-8B achieves 60.6 on CWQ when used for the full pipeline. In the fully fine-tuned pipeline, the fine-tuned Qwen3-8B model is used for Guiding Graph construction, graph exploration, and answer generation, improving performance to 69.4 on CWQ. In the hybrid pipeline, the fine-tuned Qwen3-8B

Method	1-Hop Query		Multi-Hop Query	
	Partial	Complete	Partial	Complete
IO prompt (Brown et al., 2020) w/DeepSeek-V3	35.8	22.1	12.4	6.6
CoT (Wei et al., 2022) w/DeepSeek-V3	37.2	23.3	12.8	7.1
PoG (Chen et al., 2024) w/DeepSeek-V3	67.7	62.6	23.9	13.0
FiSKE (Tao et al., 2025) w/DeepSeek-V3	71.1	62.8	28.3	12.9
Ours w/DeepSeek-V3	89.9	75.8	81.8	56.8

Table A1: Results for the agricultural graph.

Dataset	Method	LLM Call	Input Token	Output Token	Total Token
agr-one-hop	PoG	5.6	2233.9	217.4	2451.4
	FiSKE	5.9	-	-	-
	Ours	4.6	1000.0	294.4	1294.4
agr-multi-hop	PoG	6.9	3243.6	379.6	3623.3
	FiSKE	13.1	-	-	-
	Ours	4.1	975.5	380.7	1356.2

Table A2: Efficiency Comparison on agricultural dataset.

Label	Count	Interpretation
Good	28/50	Preserves the intended compositional structure
Partial	18/50	Preserves the main skeleton with imperfect constraints
Bad	4/50	Misses a key entity, answer type, or dependency

Table A3: Manual audit of Guiding Graph construction on 50 randomly sampled CWQ cases.

model is used only for Guiding Graph construction, while graph exploration and answer generation still use DeepSeek-V3; this setting achieves 73.7 on CWQ. These results indicate that the optional fine-tuned constructor improves the structural interface, while stronger general-purpose LLMs can still be retained for downstream exploration and answering.

A.6 Prompt Templates

The prompts for Guidance Graph construction includes ‘information extraction’ A7 and ‘GG construction’ A8. The prompts for Knowledge Exploration includes ‘relation pruning’ A9, ‘choose’ A10 and ‘answering’ A11. The prompt ‘choose’ is for dynamic branch selection. When you need to switch between different datasets, you only need to modify the examples in the prompts.

Type	Question	Generated Guiding Graph	Analysis
Good	What’s the currency of the country that includes the location of Buenos Aires?	[(Buenos Aires, is located in, country); (country, has currency, currency)]	The graph preserves both the intermediate node and the two-hop relation constraints.
Partial	What is the most recent event in which the sports team owned by James L. Dolan won the championship?	[(sports team, owned by, James L. Dolan); (sports team, won, championship); (championship, in, event)]	The main dependency structure is preserved, but the temporal/superlative constraint is not explicitly represented.
Bad	What does the Missouri River bisect in the Central Time Zone?	[(Missouri River, bisects, Central Time Zone)]	The graph over-compresses the relation and misses the answer target node.

Table A4: Representative Guiding Graph construction cases from the manual audit.

Method	Run 1	Run 2	Run 3	Mean $\pm$ Std
Ours	72	73	70	71.7 $\pm$ 1.5
PoG	55	57	55	55.7 $\pm$ 1.2
ToG	50	55	48	51.0 $\pm$ 3.6
FiSKE	48	50	51	49.7 $\pm$ 1.5

Table A5: Same-backbone repeated-run stability check on a random subset of 100 questions using DeepSeek-V3.

Item	Setting
Base model	Qwen3-8B
Training framework	LlamaFactory
Training stage	SFT
Fine-tuning method	LoRA
LoRA rank	16
LoRA target modules	all
Template	qwen3_nothink
Cutoff length	2048
Per-device batch size	2
Gradient accumulation steps	16
Learning rate	1.0×10^{-4}
Scheduler	cosine
Warmup ratio	0.1
Epochs	10
Precision	FP16
Data loader workers	0
Pin memory	false

Table A6: Fine-tuning configuration for the unified Guiding Graph constructor.

information extraction	Rephrase the question as a statement, ensuring: 1. **All explicit and logically implied information is included** (e.g., location, time, scope if hinted in the question). 2. Split compound noun phrases into individual entities. 3. Classify each entity as generic/specific. Do not answer the question. question: What team did Payton Manning's father play for? answer: statement: Identify the team payton manning's father played for. entities: - team (generic) - payton manning (specific) - father (generic) question: what did james k polk do before he was president? answer: statement: Identify the job of James K. Polk before he became president. entities: - job (generic) - James K. Polk (specific) - president (generic) question: Where did the "Country Nation World Tour" concert artist go to college? answer: statement: Identify the college attended by the concert artist of the "Country Nation World Tour." entities: - college (generic) - Country Nation World Tour (specific) - concert artist (generic) Now answer with the format of the example above. Be brief and precise. question: {} answer:
------------------------	--

Table A7: information extraction.

GG construction	Please analyze the relationships between the following entities in the given sentence. Represent each relationship as a triple in the format (subject, relation, object). These keywords can be used as entities or relationships in the triples, but please do not modify the keywords. If a generic keyword and another keyword refer to the same entity, the generic keyword should be treated as a relation rather than an entity. sentence: Identify the father of Keyshia Cole. generic keywords: father specific keywords: Keyshia Cole answer: triples: [(father, of, Keyshia Cole)] sentence: Identify the main trading partner of China that appointed Abdelaziz Bouteflika to a governmental position. generic keywords: main trading partner; governmental position specific keywords: China; Abdelaziz Bouteflika answer: triples: [(main trading partner, of, China), (main trading partner, appointed to governmental position, Abdelaziz Bouteflika)] Now answer with the format of the example above. Be brief and precise. sentence: {} generic keywords: {} specific keywords: {} answer:
-----------------	---

Table A8: GG construction.

relation pruning	Select the most semantically matching relation from the candidate relations to replace the given relation in the theme information while maintaining its essential meaning. Rate it on a scale of 0 to 10. Be brief and precise. Rate it on a scale of 0 to 10. Be brief and precise. Below is an example: theme information: main spoken language of country relation: of candidate relations: ['language.human_language.main_country', 'language.human_language.human_language', 'language.human_language.language_family', 'language.human_language.iso_639_3_code', 'base.rosetta.languoid.parent', 'language.human_language.writing_system', 'base.rosetta.languoid.languoid_class'] assess: - 'language.human_language.human_language' can perfectly replace the relation, (9) score. - 'language.human_language.main_country' can replace the relation, (8) score. - other relations are not related to any information in the sentence, (0) score. Now answer with the format of the example above. Be brief and precise. theme information: {} relation: {} candidate relations: {} assess:
------------------	--

Table A9: relation pruning

choose	Based on the target sentence below, determine which of the provided pieces of knowledge better matches the target sentence. Be brief and precise. Target sentence: Identify the country that Turkey trades with which contains the Annaba Province. Knowledge 1: country location.adjoining_relationship.adjoins turkey Knowledge 2: country location.contain.contains Annaba Answer: Knowledge (2) better matches the target sentence. Target sentence: {} Knowledge 1: {} Knowledge 2: {} Answer:
--------	---

Table A10: choose.

answering	Please answer the question. Triples are available for reference. If there are not enough information in the triples, please answer with your own knowledge. question: Which place is the madam satan located? triplets: [('madam satan', 'film.film.country', 'the USA'), ('madam satan', 'film.film.language', 'English')] answer: 'madam satan' is located in 'the USA'. question: {} triplets: {} answer:
-----------	--

Table A11: answering.