SRG-KC: Structured Reasoning Graphs and Knowledge Chains for Complex Knowledge Graph Question Answering

Anonymous Submission

Abstract

Large language models have demonstrated strong language understanding capabilities in open-domain question answering and knowledge-augmented generation, but in complex knowledge graph question answering they still face key challenges, including substantially different reasoning-depth requirements across questions, the risk of under-reasoning or excessive diffusion under fixed-depth strategies, and vulnerability to noisy paths when candidate reasoning paths vary widely in quality and lack explicit reliability modeling. To address these issues, this paper proposes SRG-KC, a structured reasoning graph framework. The proposed method performs joint multi-path and multi-step reasoning over knowledge graphs and explicitly models the chained process of question semantics, relation selection, and entity state evolution. Building on this process, we further introduce two core mechanisms: semantic-aware path priority modeling and path reliability fusion. The former recalibrates intermediate entity states through question-conditioned modulation, while the latter achieves robust fusion of final entity scores via dynamic intra-path step integration and inter-path reliability weighting. Experimental results demonstrate the effectiveness of the proposed method on complex multi-hop question answering tasks. On the WebQSP and CWQ datasets, the final answer accuracy reaches 86.18 and 69.01, respectively.

1 Introduction

Complex knowledge graph question answering requires a model to identify topic entities and relation semantics and to perform relation selection, entity expansion, evidence aggregation, and answer determination over multi-hop graph structures (Mitra et al., 2022; Qiao et al., 2022; Dong et al., 2023). Compared with single-hop question answering, it places stronger demands on compositional generalization, since the model must distinguish useful from useless reasoning branches and organize scattered evidence into structured support for answer inference (Shi et al., 2021; Cai et al., 2021).

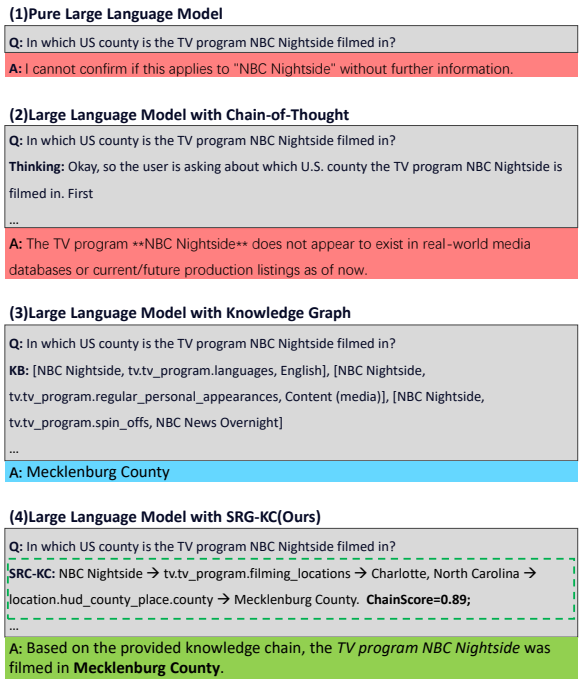

Figure 1: Comparison of four question answering paradigms: direct LLM QA, CoT-based LLM QA, KG-augmented LLM QA, and SRG-KC.

Although large language models possess non-trivial reasoning ability (OpenAI, 2023; Anthropic, 2024), their free-form textual reasoning is often not explicitly aligned with graph structure. When graph evidence is simply linearized into text, multi-hop topological constraints can be weakened, leading to missing evidence, logical confusion, and answer drift (Li et al., 2024). In contrast, graph reasoning methods can search and propagate directly over knowledge graphs, but their outputs often remain at the level of scores or query structures and are not readily usable as interpretable evidence for language models (Lan et al., 2021).

Existing methods mainly suffer from three lim-

itations: fixed reasoning depth cannot adapt well to varying question complexity; intermediate states lack explicit priority modeling under semantic constraints; and multi-path results usually lack reliability fusion, allowing local errors to propagate to final decisions (Shi et al., 2021; Zhao et al., 2024).

These observations motivate a semantically driven structured reasoning process in which question semantics guide relation selection and entity retention at each step, while intermediate propagation states constrain later reasoning. Based on this idea, we propose SRG-KC, which bridges graph reasoning and language generation through a learnable structured reasoning process.

The main contributions of this paper are as follows:

1. We propose SRG-KC, a multi-path and multi-step structured reasoning framework for complex knowledge graph question answering, which represents the reasoning process as interpretable and serializable intermediate graph states.

2. We introduce a semantic-aware path priority mechanism that re-scores intermediate entities under question-conditioned constraints, thereby improving the preservation of critical evidence.

3. We develop a path reliability fusion mechanism that combines intra-path step integration with inter-path weighted fusion, mitigating interference from noisy paths and improving the robustness of final predictions.

2 Related Work

2.1 Neural Graph Reasoning for Complex Knowledge Graph Question Answering

Early methods for complex knowledge graph question answering often relied on semantic parsing or executable queries. They are interpretable under fixed schemas and explicit constraints, but depend heavily on annotations and generalize poorly in open settings (Berant et al., 2013; Yih et al., 2015; Lan et al., 2021). Neural graph reasoning later improved multi-hop modeling through end-to-end relation prediction and entity propagation, yet still tends to rely on fixed reasoning depth and lacks explicit reliability assessment across candidate paths (Shi et al., 2021; Ye et al., 2022; Qiao et al., 2022).

2.2 Knowledge-Augmented Question Answering Driven by Large Language Models

Prompt-based question answering with large language models has achieved strong performance on many tasks (Wei et al., 2022). More recently, researchers have introduced knowledge graph evidence into prompts to mitigate hallucinations caused by purely parametric memory (Baek et al., 2023; Wang et al., 2023b). The main challenge is that flat textual concatenation often loses directional constraints and multi-hop dependencies, making it difficult to preserve coherent reasoning structure (Panda et al., 2024). Recent work has further explored constraining reasoning directly with graph structure (Luo et al., 2024) and analyzing the mismatch between retrieved graph context and downstream answer generation (Zarrinkia et al., 2026). Thus, converting graph evidence into a structured, noise-controlled representation usable by language models has become a central issue.

2.3 Differences Between Our Method and Existing Approaches

Prior work has made progress in relation matching, path search, and prompting knowledge graph evidence (Shi et al., 2021; Qiao et al., 2022; Wang et al., 2023b; Zhao et al., 2024), but provides limited modeling of how intermediate reasoning states should be preserved, filtered, and fused under question-conditioned semantics. Under multi-step propagation and multi-path reasoning, without explicit characterization of intermediate entity importance and path reliability, local noise may still affect later decisions (He et al., 2021; Du et al., 2023; Panda et al., 2024). Our method addresses this gap through semantic-aware path priority modeling and path reliability fusion, providing a more stable connection between structured graph reasoning and knowledge-chain prompting for language models.

3 Method

3.1 Task Definition

Let the knowledge graph be defined as

$$G = (\mathcal{E}, \mathcal{R}, \mathcal{T}), \tag{1}$$

where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote the sets of entities, relations, and triples, respectively. Given a natural language question q and a topic entity set $H \subseteq \mathcal{E}$,

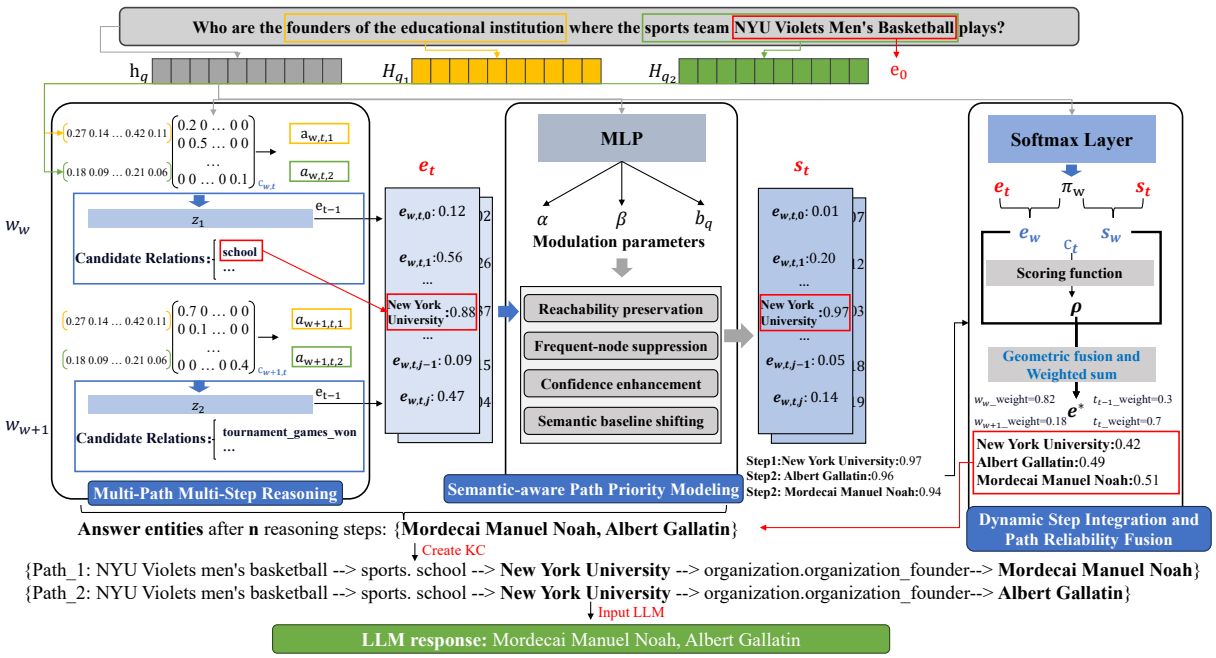

Figure 2: Overall architecture of SRG-KC. The framework first encodes the question, then performs multi-path multi-step graph reasoning with semantic-aware path priority modeling, aggregates evidence through dynamic step integration and path reliability fusion, and finally converts high-confidence paths into knowledge-chain prompts for answer generation.

the goal is to predict the answer entity set $A \subseteq \mathcal{E}$; based on this formulation, our method further seeks to locate answer entities through multi-step, multi-path structured reasoning over a candidate subgraph and to generate high-quality knowledge-chain evidence for language models.

3.2 Multi-Path Multi-Step Semantically Driven Reasoning

Complex knowledge graph question answering involves multiple semantic constraints, and relation selection is tightly coupled with entity expansion. Once an early decision deviates, the error can accumulate through later propagation. We therefore adopt a multi-path multi-step reasoning paradigm that dynamically focuses on question semantics at each step and performs constrained relation prediction and entity propagation accordingly. Section 3.3 then determines which intermediate entities should be retained, and Section 3.4 performs hierarchical fusion across steps and paths.

Given an input question q, the model first encodes it with a pretrained language model to obtain a global semantic representation $\mathbf{h}_q$ and token-level contextual representations $\mathbf{H}_q$. Based on $\mathbf{h}_q$, the model maintains W reasoning paths and T reasoning steps simultaneously, and at each step constructs a path-step-specific semantic query $\mathbf{c}_{w,t}$. This query attends to the token-level question representations to produce the current semantic context, from which multi-label relation activation probabilities $p_{w,t,r}$ are predicted for parallel candidate relations.

To reduce ineffective propagation, we introduce a dynamic relation reachability constraint that retains only those candidate relations reachable from the currently activated head entities, thereby suppressing clearly implausible relation branches at early stages and reducing propagation noise in subsequent steps. Let $\mathbf{e}_{w,t-1} \in [0,1]^{|\mathcal{E}|}$ denote the entity state at the previous step, let $\mathbf{e}_{w,0}$ be initialized by the topic entities, and perform parallel propagation based on relation probabilities:

$$\tilde{\mathbf{e}}_{w,t,j} = \sum_{(u,r,j)\in\mathcal{T}_b} e_{w,t-1,u}\, p_{w,t,r}. \tag{2}$$

Here, $\mathcal{T}_b$ denotes the triple set of the subgraph corresponding to the current sample, while u and j denote head and tail entity indices, respectively. $e_{w,t-1,u}$ is the activation strength of the head entity at the previous step, and $\tilde{\mathbf{e}}_{w,t,j}$ is the intermediate activation value of the propagated tail entity.

This propagation jointly maps head entity activation strengths and relation activation probabilities

to object entities. To balance new evidence and historical states, a gated residual update is further adopted:

$$\begin{aligned} g_{w,t} &= \sigma(\mathbf{u}_w^\top \mathbf{c}_{w,t} + c_w), \\ \mathbf{e}_{w,t} &= g_{w,t}\tilde{\mathbf{e}}_{w,t} + (1 - g_{w,t})\mathbf{e}_{w,t-1}. \end{aligned} \tag{3}$$

Here, $\mathbf{u}_w \in \mathbb{R}^d$ and $c_w \in \mathbb{R}$ are path-level gating parameters, and $g_{w,t} \in (0, 1)$ is the gating coefficient at the current step, which controls the fusion ratio between newly propagated evidence and historical states.

To avoid excessive score accumulation during multi-hop propagation, the model applies a differentiable upper-bound constraint to entries larger than 1 through overflow masking, stop-gradient normalization, and element-wise rescaling. This keeps propagation numerically stable without introducing extra gradient interference.

This subsection outputs relation activation probabilities and entity state sequences for each path and reasoning step, which serve as the inputs to later priority re-scoring and hierarchical fusion.

3.3 Semantic-Aware Path Priority Modeling

During multi-hop propagation, the entity state $\mathbf{e}_{w,t}$ obtained in the previous subsection mainly reflects the strength with which an entity is reached from the topic entity through the current combination of relations. To emphasize its role as the input to the priority re-scoring module, we denote $\mathbf{d}_{w,t} = \mathbf{e}_{w,t}$ in this subsection. However, this propagation strength is not equivalent to whether the entity is worth retaining for the current question. On the one hand, high-degree nodes may obtain high propagation values due to structural advantages. On the other hand, different questions may prefer different types of intermediate states; for example, constraint-oriented questions prefer concentrated and high-confidence candidate sets, whereas set-valued questions allow moderate dispersion. Therefore, ranking solely by reachability can introduce semantic mismatch.

To address this issue, we introduce a question-conditioned modulation mechanism. The core idea is to use question semantics as a conditioning signal to recalibrate propagation states, so that entity scores are transformed from whether an entity is reachable to whether it should be preferentially retained under the current question. Specifically, modulation parameters are generated from the global question representation:

$$[\tilde{\alpha}, \tilde{\beta}, b_q] = \mathrm{MLP}_q(\mathbf{h}_q), \tag{4}$$

where $\alpha = \tanh(\tilde{\alpha})$ and $\beta = \tanh(\tilde{\beta})$ are used as bounded modulation coefficients. The priority score is defined as

$$\begin{aligned} \mathbf{s}_{w,t} = \mathbf{d}_{w,t} \\ &+ \eta\big[\alpha \tanh(\mathbf{d}_{w,t}) + \beta\big(\mathbf{d}_{w,t} \odot \sigma(\mathbf{d}_{w,t})\big)\big] \\ &+ b_0 + b_q, \end{aligned} \tag{5}$$

where η and b_0 are learnable scalars. This equation can be decomposed into three components. The first term $\mathbf{d}_{w,t}$ preserves prior information about propagation reachability. The second branch, $\tanh(\mathbf{d}_{w,t})$, compresses extreme values and suppresses abnormal amplification caused by structurally frequent nodes. The third branch, $\mathbf{d}_{w,t} \odot \sigma(\mathbf{d}_{w,t})$, smoothly enhances the medium-to-high confidence range, thereby improving the separability of key candidates. The bias term $b_0 + b_q$ dynamically shifts the scoring baseline at the current step under question-conditioned semantics. After scoring, non-negative clipping and row-wise max normalization are applied.

$\mathbf{s}_{w,t}$ can be viewed as the semantic priority distribution of the w-th path at the t-th reasoning step, where $s_{w,t,j}$ indicates the relative retention value of entity e_j under the current question. Compared with the original propagation state $\mathbf{d}_{w,t}$, it captures not only which entities are reachable, but also which reached entities should be retained and reinforced. The normalization improves comparability across paths and steps and suppresses local outliers during later fusion.

3.4 Dynamic Step Integration and Path Reliability Fusion

After multi-step propagation, the model obtains many intermediate states across paths and steps, but their contributions to the final answer are unequal. Effective reasoning depth varies across questions, and path reliability is also uneven. We therefore introduce a hierarchical fusion strategy: dynamic step-level integration within each path, followed by reliability-weighted fusion across paths.

We first predict step-level weights from the global question representation and normalize them within each path. This distribution reflects the relative contribution of different reasoning steps within the same path for the current question. Unlike fixed termination steps or uniform averaging, it allows

the model to adaptively emphasize shallow or deep evidence depending on the question, thereby alleviating under-reasoning and excessive diffusion caused by fixed reasoning depth. Based on these weights, the entity states and priority signals within each path are aggregated as

$$\begin{aligned} \mathbf{e}_w &= \sum_{t=1}^{T} \pi_{w,t} \mathbf{e}_{w,t}, \\ \mathbf{s}_w &= \sum_{t=1}^{T} \pi_{w,t} \mathbf{s}_{w,t}. \end{aligned} \tag{6}$$

Here, $\mathbf{e}_w$ denotes the aggregated entity evidence of path w, and $\mathbf{s}_w$ denotes its aggregated priority signal. The model then estimates reliability across candidate paths by summarizing each path representation and normalizing the resulting reliability logits with a softmax distribution $\boldsymbol{\rho}$. This design reduces the influence of noisy or unstable paths. We first define geometric fusion as

$$\hat{\mathbf{e}}_{\text{geo}} = \exp\left(\sum_{w=1}^{W} \rho_w \log(\text{clip}_\epsilon(\mathbf{e}_w)) \right). \tag{7}$$

This term emphasizes multi-path consistency by assigning higher scores to candidates that are supported more uniformly across reliable paths. We then define weighted summation as

$$\hat{\mathbf{e}}_{\text{sum}} = \sum_{w=1}^{W} \rho_w \mathbf{e}_w. \tag{8}$$

This term preserves broader candidate coverage and reduces over-penalization of entities that receive strong support from only a subset of paths. Based on these two components, the final entity score is computed by interpolation:

$$\begin{aligned} \mathbf{e}^* &= \lambda \hat{\mathbf{e}}_{\text{geo}} + (1-\lambda) \hat{\mathbf{e}}_{\text{sum}}, \\ \mathbf{s}^* &= \sum_{w=1}^{W} \rho_w \mathbf{s}_w. \end{aligned} \tag{9}$$

Here, $\text{clip}_\epsilon(\cdot)$ denotes a lower-bounded clipping operator for numerical stability, and $\lambda \in [0,1]$ is the interpolation coefficient. The priority score $\mathbf{s}^*$ is obtained by reliability-weighted summation. Overall, this hierarchical fusion reduces local noise amplification and improves the stability of final reasoning.

3.5 Training Objective

The main loss adopts a weighted mean squared error over the candidate entity space. Let $\mathbf{y}$ denote the answer label vector, $\mathbf{m}$ denote the candidate entity mask, and define the class weight as $\omega_j = 9y_j + 1$. Then,

$$\mathcal{L}_{\text{main}} = \frac{\sum_j m_j \omega_j (e_j^* - y_j)^2}{\sum_j m_j \omega_j}. \tag{10}$$

The final entity score $\mathbf{e}^*$ is produced by the dynamic step integration and path reliability fusion in Section 3.4. Accordingly, $\mathcal{L}_{\text{main}}$ directly constrains the fused global entity prediction and back-propagates through step weighting, path reliability estimation, and earlier relation prediction and propagation.

To strengthen priority ranking capability, we introduce a margin-based auxiliary loss. For the i-th sample, let p_i denote the average fused priority score assigned to positive answer entities and let n_i denote the average fused priority score assigned to negative candidate entities inside the candidate mask:

$$\begin{aligned} p_i &= \frac{\sum_j s_{i,j}^* y_{i,j}}{\sum_j y_{i,j} + \epsilon}, \\ n_i &= \frac{\sum_j s_{i,j}^* \mathbb{I}[m_{i,j} = 1 \wedge y_{i,j} < 0.5]}{\sum_j \mathbb{I}[m_{i,j} = 1 \wedge y_{i,j} < 0.5] + \epsilon}. \end{aligned} \tag{11}$$

The auxiliary loss is defined as

$$\mathcal{L}_{\text{aux}} = \frac{1}{B} \sum_{i=1}^{B} \max(0, \delta - (p_i - n_i)). \tag{12}$$

Both quantities are computed from the fused priority score $\mathbf{s}^*$. Thus, the auxiliary loss directly supervises priority ranking, encouraging positive entities to receive higher overall priority than negative ones.

Here, δ is the margin hyperparameter. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}, \tag{13}$$

where λ_{aux} controls the influence of ranking supervision; during training, this auxiliary term is jointly optimized with the main loss to enhance the model's ability to separate positive and negative candidates in relative ranking. Concretely, $\mathcal{L}_{\text{main}}$ back-propagates through the final entity scores, while $\mathcal{L}_{\text{aux}}$ back-propagates through the fused priority scores, so that the learnable parameters in question encoding, relation scoring, semantic-aware priority modeling, and path fusion are jointly updated end to end.

Type	Method	AccessKB	WebQSP		CWQ	
			Acc.	Hits@10	Acc.	Hits@10
LLM-only	IO prompts (Patel et al., 2023)	No	63.3	-	37.6	-
	CoT prompts (Wei et al., 2022)	No	62.2	-	38.8	-
	SC (Wang et al., 2023a)	No	61.1	-	45.4	-
KBQA	Previous KBQA (Jiang et al., 2023)	Yes	77.2	-	51.2	-
	ToG-R (Sun et al., 2023)	Yes	75.8	-	58.9	-
Graph Reasoning + LLM	KG-CoT (Zhao et al., 2024)	Yes	84.9	91.4	62.3	72.5
	SRG-KC(ours)	Yes	86.18	93.58	69.01	76.88

Table 1: Comparison of end-to-end final answer accuracy (%) on WebQSP and CWQ. Type indicates the broad method family, and AccessKB indicates whether a method explicitly uses external knowledge graph evidence during inference. For methods with available graph reasoning results, WebQSP Hits@10 and CWQ Hits@10 (%) are additionally reported to reflect the ranking quality of intermediate graph reasoning outputs.

4 Experiments

4.1 Research Questions

Our experiments are designed to answer three questions: 1: whether the proposed framework can improve structured evidence quality and final answer performance in complex knowledge graph question answering; 2: whether semantic-aware path priority modeling and path reliability fusion each provide stable performance gains; and 3: whether structured reasoning graphs, when used as knowledge-chain prompts, can improve answer interpretability.

4.2 Datasets and Implementation Details

We evaluate the proposed method on the WebQSP and CompWebQ datasets (Yih et al., 2016; Shi et al., 2021). WebQSP mainly contains one-hop and two-hop questions, whereas CompWebQ involves more complex multi-hop and compositional constraints. In the current project setup, the WebQSP resources used for experimentation contain 3,097 training questions and 1,596 test questions, while the CompWebQ resources loaded by our implementation contain 6,997 training instances and 3,192 test instances. For fair comparison, we follow the public subgraph retrieval results and standard data splits.

The question encoder is based on BERT (Devlin et al., 2019). Training uses grouped learning rates, linear warm-up, gradient clipping, and candidate entity masking. All reported results are based on the final implementation with gated propagation, semantic priority modeling, and path reliability fusion.

4.3 Experimental Settings

Unless otherwise stated, the model uses $W=3$ reasoning paths and $T=4$ reasoning steps. The question encoder is BERT-base-uncased (Devlin et al., 2019). Training uses batch size 8, BERT learning rate 3×10^{-5}, task learning rate 10^{-3}, weight decay 10^{-5}, warm-up proportion 0.1, and 100 training epochs. Reverse relations are included in the reasoning graph, and optimization is performed with AdamW (Loshchilov and Hutter, 2017). For the main paper results, the answer-generation stage uses a locally deployed Qwen2.5-72B (Yang et al., 2024) with deterministic decoding and temperature set to 0. Additional implementation details and supplementary reproduction settings are reported in Appendix A.

The evaluation metrics fall into two categories:

1. Final answer metric: the accuracy of answer outputs generated by the language model, which measures whether the end-to-end prediction matches the gold answer.

2. Graph reasoning metrics: ranking metrics such as Hits@1, Hits@3, and Hits@10, which measure how well the correct answer is retained and ranked near the top during graph reasoning.

4.4 Main Results

Table 1 presents the comparison results of final answer accuracy.

Table 1 summarizes final answer accuracy under the respective reported settings of different methods. When available, each dataset's graph reasoning Hits@10 (%) is listed next to its corresponding final answer accuracy. Under our current experimental setup, SRG-KC reaches 86.18 on WebQSP

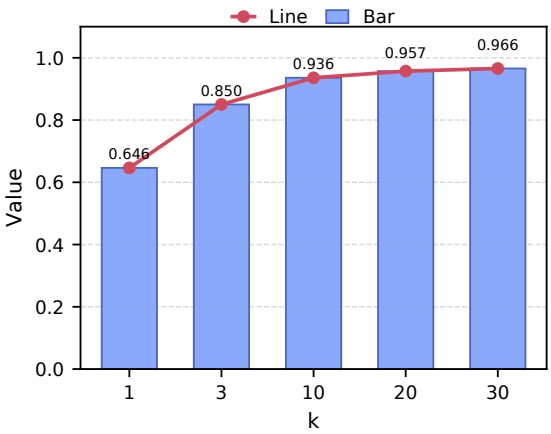

Figure 3: WebQSP Hits@k performance of SRG-KC at the graph reasoning stage.

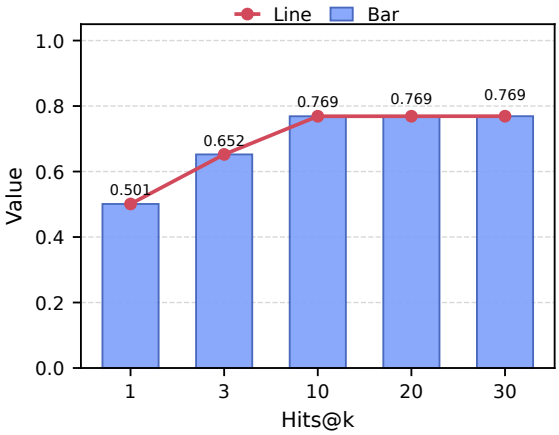

Figure 4: CWQ Hits@k performance of SRG-KC at the graph reasoning stage.

and 69.01 on CWQ, with graph reasoning Hits@10 of 93.58 and 76.88, respectively.

SRG-KC improves over prior graph reasoning methods mainly by introducing semantic-aware path priority modeling and dynamic step integration with path reliability fusion. The former suppresses high-propagation but semantically weak candidates, and the latter reduces error accumulation from fixed-depth reasoning and noisy paths through cross-step integration and cross-path weighting. Thus, the main advantage of SRG-KC lies in finer control and more robust fusion of intermediate graph reasoning states.

Figures 3 and 4 further illustrate the ranking behavior of SRG-KC. On WebQSP, the curve rises steadily from Hits@1 to Hits@10 and then gradually saturates; on CWQ, the lower overall values and flatter tail are consistent with its stronger compositional constraints and larger candidate space.

Setting	Hits@1	Hits@3	Hits@10
Full SRG-KC	0.6957	0.8897	0.9358
w/o Sem. Priority	0.5373	0.6197	0.6360
w/o Step-Path Fusion	0.4568	0.6067	0.6379

Table 2: Ablation results on WebQSP for the full SRG-KC model and two ablated variants, evaluated by graph reasoning Hits@1, Hits@3, and Hits@10.

Setting	Hits@1	Hits@3	Hits@10
Full SRG-KC	0.5009	0.6523	0.7688
w/o Sem. Priority	0.4245	0.5652	0.7174
w/o Step-Path Fusion	0.3947	0.6247	0.7500

Table 3: Ablation results on CWQ for the full SRG-KC model and two ablated variants, evaluated by graph reasoning Hits@1, Hits@3, and Hits@10.

These ranking results are consistent with final answer accuracy: keeping correct or highly relevant candidates near the top provides better structured inputs for later knowledge-chain prompting. This supports the goal of propagating gains from front-end graph reasoning control to downstream language-model answer synthesis.

4.5 Ablation Study

To verify the actual contributions of the proposed key modules to reasoning performance, we conduct ablation studies around two core designs: removing semantic-aware path priority modeling, and removing dynamic step integration with path reliability fusion. Since the ablation study mainly reflects entity ranking and evidence selection capability at the graph reasoning stage, we report Hits@1, Hits@3, and Hits@10 as evaluation metrics.

Table 2 presents the ablation results on the WebQSP dataset.

As shown in Table 2, the full model substantially outperforms both ablated settings on WebQSP. Removing semantic-aware path priority modeling reduces Hits@1/3/10 by 0.1584/0.2700/0.2998, showing that propagation strength alone cannot reliably identify the critical reasoning frontier. Removing dynamic step integration and path reliability fusion causes an even larger degradation, with Hits@1 dropping to 0.4568, indicating that adaptive depth selection and path reliability are important even for relatively short reasoning chains.

Table 3 presents the ablation results on the CWQ dataset.

On the more challenging CWQ dataset, the

full model again performs best. Removing semantic-aware path priority modeling decreases Hits@1/3/10 by 0.0764/0.0871/0.0514, indicating that question-conditioned recalibration improves ranking quality under stronger compositional constraints. Removing dynamic step integration and path reliability fusion decreases Hits@1 to 0.3947, showing that this module is especially important for placing the correct answer at the top rank.

Overall, both modules contribute substantially. Semantic-aware path priority modeling improves semantic consistency in intermediate ranking, while dynamic step integration with path reliability fusion alleviates depth mismatch and noisy path interference.

4.6 From Structured Reasoning Graphs to Knowledge Chain Prompts

To fully specify the downstream answer-generation stage, this subsection describes how graph reasoning outputs are converted into knowledge-chain prompts. During reasoning, the model produces step-wise relation predictions, entity propagation states, and cross-step and cross-path fusion signals. Based on these intermediate signals, the system identifies high-confidence reasoning paths, unfolds relation chains from the topic entity, and serializes the retained paths into structured facts of the form `Path: Topic Entity -> relation_1 -> entity_1 -> relation_2 -> entity_2`.

These structured paths are then combined with the original question and passed to the language model for answer generation. In this way, SRG-KC first locates relatively reliable reasoning trajectories on the graph side and then exposes them as explicit knowledge chains for downstream decoding. Compared with directly linearizing retrieved graph evidence, this conversion better preserves multi-hop dependency, relation directionality, and the internal consistency of the reasoning process, thereby providing more stable structured support for final answer generation.

5 Conclusion

This paper proposes SRG-KC, a structured reasoning framework for complex knowledge graph question answering. Under question-guided semantics, the method jointly models relation selection, entity propagation, and intermediate-state control, and further incorporates semantic-aware path priority modeling together with dynamic step integration and path reliability fusion to improve evidence retention and the stability of cross-path aggregation in multi-hop reasoning; on this basis, it also converts high-confidence structured results into knowledge-chain prompts for language models.

Experiments on WebQSP and CWQ show that SRG-KC improves both final answer accuracy and graph reasoning quality. Ablation studies further confirm that the two proposed modules are the main source of the gains, improving both ranking quality and robustness to noisy paths.

Limitations

SRG-KC still depends on the quality of front-end structured reasoning: when early relation expansion misses key facts, downstream generation has limited ability to recover. For example, for the question "Where did the 'Country Nation World Tour' concert artist go to college?", the retained evidence reaches `Brad Paisley` but also includes an unrelated path ending at `Colorado State University`, while the crucial link to `Belmont University` is not recovered, causing the final answer to drift toward a plausible but unsupported candidate. Moreover, even when the retained path is close to the target answer, the final generation stage may still fail because of answer normalization or output-form errors. Thus, SRG-KC is better at selecting and organizing retrieved evidence than at recovering facts omitted early in reasoning.

Ethics Statement

The experiments use public KGQA benchmarks and do not involve sensitive personal data. However, LLM-based systems may still generate incorrect answers and should not be directly deployed in high-risk scenarios without human supervision.

References

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. https://assets.anthropic.com/m/61e7d27f8c8f5919/original/Claude-3-Model-Card.pdf. Model card.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from

question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Jianyu Cai, Zhanqiu Zhang, Feng Wu, and Jie Wang. 2021. Deep cognitive reasoning network for multi-hop question answering over knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 219–229, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Junnan Dong, Qinggang Zhang, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. 2023. Hierarchy-aware multi-hop question answering over knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, pages 2519–2527, Austin, TX, USA. Association for Computing Machinery.

Haowei Du, Quzhe Huang, Chen Li, Chen Zhang, Yang Li, and Dongyan Zhao. 2023. Relation-aware question answering for heterogeneous knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13582–13592, Singapore. Association for Computational Linguistics.

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 Fourteenth ACM International Conference on Web Search and Data Mining*, pages 553–561.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Complex knowledge base question answering: A survey. *arXiv preprint arXiv:2108.06688*.

Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2024. A survey of graph meets large language model: Progress and future directions. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 8123–8131.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. 2024. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *arXiv preprint arXiv:2410.13080*.

Sayantan Mitra, Roshni Ramnani, and Shubhashis Sengupta. 2022. Constraint-based multi-hop question answering with knowledge graph. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 280–288, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Subhadarshi Panda, Subrata Pani, and Saeedeh Shekarpour. 2024. Holmes: Hyper-relational knowledge graphs for multi-hop question answering using llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13320–13338, Bangkok, Thailand. Association for Computational Linguistics.

Ajay Patel, Bryan Li, Mohammad Sadegh Rasooli, Noah Constant, Colin Raffel, and Chris Callison-Burch. 2023. Bidirectional language models are also few-shot learners. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Zile Qiao, Wei Ye, Tong Zhang, Tong Mo, Weiping Li, and Shikun Zhang. 2022. Exploiting hybrid semantics of relation paths for multi-hop question answering over knowledge graphs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1813–1822, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

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

Yanjie Wang, Lingyong Yu, Jiawen Zhang, Xin Chen, Xiang Zhou, and Yuxia Li. 2023b. Knowgpt: Knowledge graph based prompting for large language models. *arXiv preprint arXiv:2312.06185*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043, Dublin, Ireland. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1311–1321, Beijing, China. Association for Computational Linguistics.

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

Yasaman Zarrinkia, Venkatesh Srinivasan, and Alex Thomo. 2026. The reasoning bottleneck in graph-rag: Structured prompting and context compression for multi-hop qa. *arXiv preprint arXiv:2603.14045*.

Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. 2024. Kg-cot: Chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6642–6650.

A Appendix

A.1 Implementation Details

This subsection supplements Sections 3.2–3.5 and Section 4.3 with implementation details not unfolded in the main text. Questions are tokenized to a maximum length of 300 tokens with fixed-length padding. Training uses batch size 8, 100 epochs, AdamW, learning rate 0.001 for non-encoder parameters, learning rate 3×10^{-5} for the BERT encoder, weight decay 10^{-5}, warm-up proportion 0.1, and random seed 2024. The implementation is based on PyTorch 2.4.1+cu121 and Transformers 4.46.3. The trainable graph reasoning model contains approximately 155.91M parameters, with BERT-base-uncased contributing 109.48M parameters. Training is conducted on a single NVIDIA A100 GPU, with total training time of approximately 26 hours and evaluation time of about 20 minutes. The reported implementation uses one reasoning path and four reasoning steps.

As a supplement to Section 3.2, each path-step pair is assigned an independent question projection:

$$\mathbf{c}_{w,t} = \tanh\left(\mathbf{W}_{w,t}^{\text{enc}}\mathbf{h}_q + \mathbf{b}_{w,t}^{\text{enc}}\right). \tag{14}$$

The resulting step-specific question state is used to form the contextual representation for relation scoring. In implementation, the relation logit for relation r at step t of path w is computed as

$$\ell_{w,t,r} = \mathbf{W}_w^{\text{rel}}\mathbf{z}_{w,t} + \mathbf{b}_w^{\text{rel}}, \tag{15}$$

where $\mathbf{z}_{w,t}$ denotes the attention-pooled question context. To complement the reachability constraint described in Section 3.2, the mask is instantiated as

$$\mathbf{M}_{w,t,r} = \mathbb{I}[\exists(h,r,o) \in \mathcal{G}_c \text{ s.t. } \mathbf{e}_{w,t-1}(h) > 0], \tag{16}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function, $\exists(h,r,o) \in \mathcal{G}_c$ means that there exists a triple (h,r,o) in the candidate subgraph $\mathcal{G}_c$, and *s.t.* abbreviates "such that." The mask takes value 1 if and only if such a triple exists and its head entity h is active at step $t-1$. Unreachable relations are then suppressed by setting their logits to a large negative constant before sigmoid normalization. In the same part of the implementation, token-level attention is multiplied by the input attention mask and then renormalized so that padded positions do not affect the contextual question vector.

As a supplement to Section 3.3, the question-conditioned modulation module is implemented as a two-layer multilayer perceptron with hidden size 64, ReLU activation, and dropout rate 0.1. After the modulation scores are produced, they are clipped to be nonnegative and normalized by the maximum value within each sample.

As a supplement to Section 3.4, path reliability is estimated from the mean step representation of each path:

$$\begin{aligned} \mathbf{u}_w &= \frac{1}{T}\sum_{t=1}^{T}\mathbf{c}_{w,t}, \\ \rho_w &= \text{softmax}\big(\text{MLP}_{\text{path}}(\mathbf{u}_w)\big), \end{aligned} \tag{17}$$

where MLP_{path} is a two-layer perceptron with hidden size 64 and dropout rate 0.1. The final entity score then interpolates weighted geometric aggregation and weighted summation with coefficient 0.7. In addition, whenever propagated entity activations exceed the predefined upper bound, the implementation applies stop-gradient-based rescaling for numerical stability, corresponding to the stabilization step discussed after the propagation update in Section 3.2.

As a supplement to Section 3.5, the reported implementation uses auxiliary-loss coefficient 0.5 and margin 0.2. In back-propagation, $\mathcal{L}_{\text{main}}$ is applied through the fused entity score $\mathbf{e}^*$, so its gradients pass through the step weights, the path reliability weights ρ_w, the path-level entity states $\mathbf{e}_w$, and the earlier propagation states $\mathbf{e}_{w,t}$. By contrast, $\mathcal{L}_{\text{aux}}$ is applied through the fused priority score $\mathbf{s}^*$, so its gradients further act on the aggregated priority signals $\mathbf{s}_w$ and step-level priority states $\mathbf{s}_{w,t}$, thereby refining the question-conditioned modulation parameters and the downstream fusion layers associated with semantic-aware priority modeling. Training is performed with mixed-precision gradient scaling, followed by gradient value clipping at 0.5, gradient norm clipping at 2.0, linear warm-up scheduling, and parameter update under AdamW. Model selection is based on the highest answer accuracy on the evaluation split.

As a supplement to Section 4.3, reasoning is conducted on the retrieved candidate subgraph without imposing an additional manually specified size cap after retrieval. The candidate entity range is defined as the union of all subject and object entities appearing in that subgraph. During training, a sample is removed only when neither the topic entity nor any gold answer appears in this range. For answer generation, the selected structured paths are serialized together with the question into a fixed

prompt that requests the model to return all answer entities directly and to avoid explanations. Generation is performed with a locally deployed Qwen2.5-72B under deterministic decoding, with temperature set to 0, `enable_thinking=False`, timeout set to 300 seconds, and at most 5 retries. For multi-answer questions, the generated response is evaluated after normalization under the same exact-match criterion. The matching procedure removes non-alphanumeric characters from both prediction and gold answer, strips parenthetical suffixes from gold strings, and then regards a prediction as correct only when the normalized output exactly matches a normalized gold answer string. Consequently, the reported final answer accuracy in Table 1 should be interpreted as end-to-end answer accuracy under this fixed prompting and normalization protocol, rather than as another form of graph ranking score.

A.2 Path-to-Answer Procedure

After graph reasoning, the system first selects the reasoning depth with the highest step-level confidence. It then extracts the entity scores and priority scores at that depth and applies the path-to-answer procedure summarized in Tables 4 and 5.

Item	Setting
Reasoning depth	highest step-level confidence
Tracing entities	top 10 by priority score
Fallback pool	top 100 entities
Relation threshold	> 0.5
Relaxed threshold	0.01 if no edge is retained

Table 4: Path extraction settings in Appendix A.2.

The retained paths are then serialized into the answer-generation prompt using the following template:

```
Question
{Path: path_1}
{Path: path_2}
...
Based on the relational facts provided,
you are asked to return all answer
entities directly. Do not provide any
explanations, and the output must not
be empty.
```

If no structured path reaching a retained entity can be recovered, the system introduces a fallback edge of the form `topic_entity` $\rightarrow$ `fallback_candidate` $\rightarrow$ `entity`. Answer generation is performed with a locally deployed Qwen2.5-72B under deterministic decoding. For answer matching, non-alphanumeric characters are removed from both prediction and gold answer, parenthetical suffixes are stripped from gold strings, and normalized exact match is applied to both single-answer and multi-answer evaluation.

Item	Setting
Generator	locally deployed Qwen2.5-72B
Decoding	deterministic
Temperature	0
Enable_thinking	False
Timeout	300 s
Retries	at most 5

Table 5: Answer-generation settings in Appendix A.2.

A representative successful example from the evaluation records is the question: "What type of government is used in the country with Northern District?" The serialized evidence is:

```
Path: North District -> location.admini
strative_division.first_level_division_
of -> Israel
Path: North District -> loca
tion.administrative_division.first
_level_division_of -> Israel ->
location.country.form_of_government ->
Parliamentary system
```

This case illustrates the full pipeline. The topic entity `North District` initializes the entity state. Under the reachability constraint, the first step activates only relations attached to currently active head entities, which allows propagation from `North District` to `Israel`. At the next step, the frontier already contains `Israel`, enabling the relation `location.country.form_of_government` and yielding `Parliamentary system`. Because the terminal entities of these paths remain in the high-priority set at the selected reasoning depth, the paths are retained and serialized into the prompt for answer generation. The resulting prediction is `Parliamentary system`, which matches the gold answer after normalization. This example directly reflects the intended workflow from graph reasoning to entity and relation filtering, to knowledge-chain construction, and finally to language-model answering. When the reasoning frontier does not support such a structured path, the system falls back to `fallback_candidate` serialization rather than introducing unsupported graph edges.