

# Dynamically Adaptive Reasoning via LLM-Guided MCTS for Efficient and Context-Aware KGQA

Yingxu Wang<sup>1</sup>, Shiqi Fan<sup>2</sup>, Mengzhu Wang<sup>3</sup>, Siwei Liu<sup>4</sup>

<sup>1</sup>Mohamed bin Zayed University of Artificial Intelligence    <sup>2</sup>The Hong Kong Polytechnic University

<sup>3</sup>Hebei University of Technology    <sup>4</sup>University of Aberdeen  
yingxu.wang@gmail.com, fsq@mail.nwpu.edu.cn, dreamkily@gmail.com, siwei.liu@abdn.ac.uk

## Abstract

Knowledge Graph Question Answering (KGQA) aims to interpret natural language queries and perform structured reasoning over knowledge graphs by leveraging their relational and semantic structures to retrieve accurate answers. Recent KGQA methods primarily follow either retrieve-then-reason paradigm, relying on GNNs or heuristic rules for static paths extraction, or dynamic path generation strategies that use large language models (LLMs) with prompting to jointly perform retrieval and reasoning. However, the former suffers from limited adaptability due to static path extraction and lack of contextual refinement, while the latter incurs high computational costs and struggles with accurate path evaluation due to reliance on fixed scoring functions and extensive LLM calls. To address these issues, this paper proposes **Dynamically Adaptive MCTS-based Reasoning (DAMR)**, a novel framework that integrates symbolic search with adaptive path evaluation for efficient and context-aware KGQA. DAMR employs a Monte Carlo Tree Search (MCTS) backbone guided by an LLM-based planner, which selects top- $k$  relevant relations at each step to reduce search space. To improve path evaluation accuracy, we introduce a lightweight Transformer-based scorer that performs context-aware plausibility estimation by jointly encoding the question and relation sequence through cross-attention, enabling the model to capture fine-grained semantic shifts during multi-hop reasoning. Furthermore, to alleviate the scarcity of high-quality supervision, DAMR incorporates a dynamic pseudo-path refinement mechanism that periodically generates training signals from partial paths explored during search, allowing the scorer to continuously adapt to the evolving distribution of reasoning trajectories. Extensive experiments on multiple KGQA benchmarks show that DAMR significantly outperforms state-of-the-art methods.

## Introduction

Large Language Models (LLMs) have demonstrated impressive reasoning capabilities across diverse tasks, including mathematical problem solving (Pei et al. 2025; Didolkar et al. 2024), commonsense inference (Wang et al. 2023b; Toroghi et al. 2024), and open-domain question answering (Zhao et al. 2023). Despite their generalization ability, LLMs often struggle in domain-specific scenarios due to the lack of grounded external knowledge, resulting in factual hallucinations and high inference costs (Huang et al. 2025b; Wang et al. 2024c). To address these limitations,

recent efforts have explored integrating domain knowledge into LLM reasoning. A promising direction to overcome these limitations is Knowledge Graph Question Answering (KGQA) (Dammu, Naidu, and Shah 2025; Saxena, Tripathi, and Talukdar 2020; Choi et al. 2023; Yin et al. 2024b), which integrates symbolic relational structures into the reasoning process to provide factual grounding and structural interpretability. By combining the expressiveness of natural language with the precision of knowledge graphs, KGQA offers a scalable solution to improve factual consistency, reasoning transparency, and answer reliability (Liu et al. 2025; Yao et al. 2025).

Existing KGQA approaches can be broadly categorized into two main paradigms based on how they construct reasoning paths: retrieve-then-reason methods and dynamic path generation strategies. The first category adopts a retrieve-then-reason paradigm, where candidate reasoning paths are extracted using either Graph Neural Networks (GNNs) (Ma et al. 2025a; Yao et al. 2025; Yin et al. 2024c; Wang et al. 2024d,b) or rule-based heuristics (Chen et al. 2023; Fang et al. 2024; Yin et al. 2022) prior to answer prediction. However, these methods lack adaptability, as GNNs fail to incorporate question-specific semantics at inference time, while heuristic rules are inherently inflexible to support dynamic reasoning refinement (Liu et al. 2025; Yao et al. 2025). In contrast, dynamic path generation strategies unify retrieval and reasoning by constructing reasoning paths dynamically during question processing. These methods either prompt LLMs to iteratively generate paths via in-context learning or Chain-of-Thought (CoT) prompting (Sui et al. 2024; Li et al. 2024; Yin et al. 2024a), or employ guided search techniques such as Monte Carlo Tree Search (MCTS) to incrementally expand paths with the aid of a path scorer (Ma et al. 2025b; Shen et al. 2025). Despite their flexibility, these approaches incur substantial computational overhead due to repeated LLM invocation and exhibit limited evaluation accuracy, as static scorers fail to capture the evolving semantics of reasoning paths (Chang et al. 2024; Shen et al. 2025).

This paper investigates the design of an adaptive KGQA framework to address the challenges of computational inefficiency and limited path evaluation accuracy in dynamic reasoning. However, developing such a framework presents several key challenges: (1) *How to modularize reasoning*

*to reduce LLM overuse during search?* A major source of computational inefficiency in dynamic KGQA lies in repeatedly invoking LLMs for both relation retrieval and reasoning during multi-hop path construction (Shen et al. 2025; Long et al. 2025). While methods such as CoT and MCTS provide flexible exploration, they tightly couple LLMs with each decision step, resulting in high inference costs and limited scalability. The key challenge is to design a modular reasoning framework that utilizes LLMs efficiently, guiding the search process without requiring direct involvement in every reasoning step. (2) *How to accurately evaluate evolving reasoning paths?* As multi-hop reasoning paths are incrementally constructed, their semantics evolve with each newly added relation and contextual information. However, existing methods typically rely on static scoring functions or shallow similarity metrics, which fail to capture the nuanced semantic shifts that occur throughout the reasoning process (Xu et al. 2024; Sui et al. 2024). This raises a key challenge: how to design a path evaluation model that adaptively captures fine-grained semantic changes conditioned on both the question and the evolving relation sequence. (3) *How to train a reliable path evaluation model with limited supervision?* Accurate path ranking in KGQA hinges on a well-calibrated evaluation model. However, dynamic reasoning methods typically produce a large number of incomplete or irrelevant paths, with only a small subset corresponding to valid reasoning trajectories. This results in highly imbalanced and noisy supervision, especially for multi-hop questions where successful paths are extremely sparse. Although reinforcement learning has been explored to mitigate this issue (Ma et al. 2024; Zhai et al. 2024), it frequently suffers from sparse rewards and unstable optimization. Therefore, the key challenge is how to construct meaningful learning signals from limited or implicit supervision to enable adaptive training of the path scorer.

To address the above challenges, we propose **Dynamically Adaptive MCTS-based Reasoning (DAMR)**, an efficient and adaptive reasoning framework that integrates symbolic search with context-aware semantic modeling to enable accurate and LLM-efficient multi-hop reasoning for KGQA. DAMR is built on an MCTS backbone, where an LLM-based planner dynamically guides path expansion by proposing semantically relevant relations at each step, significantly reducing search space and improving answer identification efficiency. To enable accurate and context-sensitive path evaluation, we introduce a lightweight Transformer-based scorer that estimates path plausibility by jointly encoding the question and relation sequence via cross-attention, effectively capturing evolving semantics during multi-hop reasoning. To address supervision scarcity, DAMR incorporates a dynamic pseudo-path mechanism that continuously adapts the scorer during search. Partial paths sampled from MCTS rollouts are ranked by predicted plausibility and converted into pseudo-path supervision pairs, amplifying learning signals from promising trajectories while suppressing noise from suboptimal ones. Extensive experiments on benchmark KGQA datasets demonstrate that DAMR significantly outperforms state-of-the-art baselines. Our contributions are summarized as follows:

- We study adaptive path reasoning in KGQA, where the key challenges lie in capturing the evolving semantics of multi-hop reasoning paths and ensuring computational efficiency during search, motivating the need for dynamic and context-aware reasoning strategies.
- We propose DAMR, a novel framework that integrates MCTS with a dynamically adapted path evaluation model, enhancing evaluation accuracy while maintaining computational efficiency.
- We conduct extensive experiments across multiple KGQA benchmarks, demonstrating that DAMR consistently outperforms state-of-the-art methods.

## Related work

**Knowledge Graph Question Answering (KGQA).** KGQA aims to enhance reasoning capabilities by incorporating external knowledge graphs to answer natural language questions (Wang et al. 2022a; Choi et al. 2023; Xu et al. 2025). Existing KGQA approaches can be broadly classified into two categories: retrieve-then-reason and dynamic path generation. The first category extracts candidate reasoning paths using Graph Neural Networks (GNNs) (Yao et al. 2023; Wang et al. 2024a; Ma et al. 2025a; Yao et al. 2025) or rule-based heuristics (Fang et al. 2024), followed by LLM-based answer generation. While GNNs learn embeddings to identify relevant paths and rule-based methods apply predefined patterns (Zhao et al. 2023; Liu et al. 2025; Wang et al. 2025), these approaches lack the flexibility to adapt dynamically to question-specific context during inference. In contrast, dynamic path generation methods, such as CoT prompting (Sui et al. 2024; Li et al. 2024) and MCTS (Ma et al. 2025b; Shen et al. 2025), unify retrieval and reasoning for more flexible exploration. However, they suffer from high computational overhead due to repeated LLM calls, and static scorers often fail to adapt to evolving path semantics (Long et al. 2025; Shen et al. 2025). To address these challenges, we propose an adaptive framework that integrates symbolic search with a fine-tuned evaluation model, aiming to improve both computational efficiency and reasoning accuracy in KGQA.

**Adaptive and Self-Improving Reasoning Models.** A promising approach to developing adaptive reasoning models is to frame the process within a reinforcement learning (RL) paradigm, where an agent learns a policy to navigate a state space. Early methods such as DeepPath (Xiong, Hoang, and Wang 2017) and MINERVA (Das et al. 2018) used RL to discover reasoning paths by rewarding the agent only when a correct answer is reached. However, this leads to the sparse rewards problem—positive feedback arrives only after long action sequences, resulting in weak learning signals and poor exploration efficiency (Zhai et al. 2024; Chang et al. 2023). To address this challenge, an alternative is self-training via pseudo-labeling, where the model learns from its own high-confidence predictions (Lee et al. 2013; Xie et al. 2020). While commonly used in semi-supervised learning, pseudo-labeling proves especially effective in reasoning tasks with limited supervision (Wang et al. 2022b; Huang et al. 2025a). Instead of relying on sparse terminal

rewards, we leverage intermediate search paths as dynamic pseudo-paths, offering dense and adaptive supervision. This facilitates continual refinement of the path evaluator to better capture the evolving semantics of reasoning.

## Preliminary

### Problem Formulation

We define Knowledge Graph Question Answering (KGQA) as the task of answering a natural language question  $q$  by reasoning over a knowledge graph  $\mathcal{K}$ . The knowledge graph is typically represented as a set of triples  $\mathcal{K} = \{(e_s, r, e_o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ , where  $\mathcal{E}$  and  $\mathcal{R}$  denote the sets of entities and binary relations. The goal of KGQA is to find a set of answers  $\mathcal{A}_q \subseteq \{(e_1, r_1, e_2), (e_2, r_2, e_3) \dots\}$  for question  $q$ , such that a semantic reasoning path through the graph leads from a topic entity to the correct answer. Formally, this is often framed as mapping  $q$  to an executable query program  $p_q$ , where  $\text{LLM}(p_q|\mathcal{K}) = \mathcal{A}_q$ .

### Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) (Kocsis and Szepesvári 2006) is a heuristic search algorithm designed for optimal decision-making in structured search spaces. It incrementally builds a search tree through stochastic sampling and consists of four key stages:

**Selection.** Starting from the root, recursively select child nodes with the highest value according to the Upper Confidence Bound for Trees (UCT) criterion:

$$UCT = \frac{w_i}{n_i} + C \sqrt{\frac{\ln N}{n_i}}, \quad (1)$$

where  $w_i$  is the accumulated reward of node  $i$ ,  $n_i$  is the visit count of node  $i$ ,  $N$  is the visit count of its parent, and  $C$  balances exploration and exploitation.

**Expansion.** Upon reaching a non-terminal leaf node, add one or more unexplored child nodes to the tree.

**Simulation.** From the newly added node, perform a random rollout (i.e., simulated trajectory) to a terminal state.

**Backpropagation.** Propagate the simulation outcome back up the tree, updating the statistics (e.g., visit count and reward) of each node along the path.

This iterative process incrementally refines the search tree, guiding the exploration toward high-reward paths.

## Methodology

### Overview of Framework

In this paper, we propose a dynamically adaptive reasoning framework DAMR for KGQA, as shown in Fig. 1. DAMR comprises three components: (1) **LLM Guided Expansion.** DAMR employs MCTS to incrementally expand reasoning paths, guided by an LLM-based planner that proposes relevant relations. This significantly reduces computational overhead and enhances efficiency in knowledge graph exploration; (2) **Context-Aware Path Evaluation.** To capture the evolving semantics of reasoning paths, DAMR employs a lightweight Transformer-based scorer with cross-attention to jointly encode the question and path embeddings. This

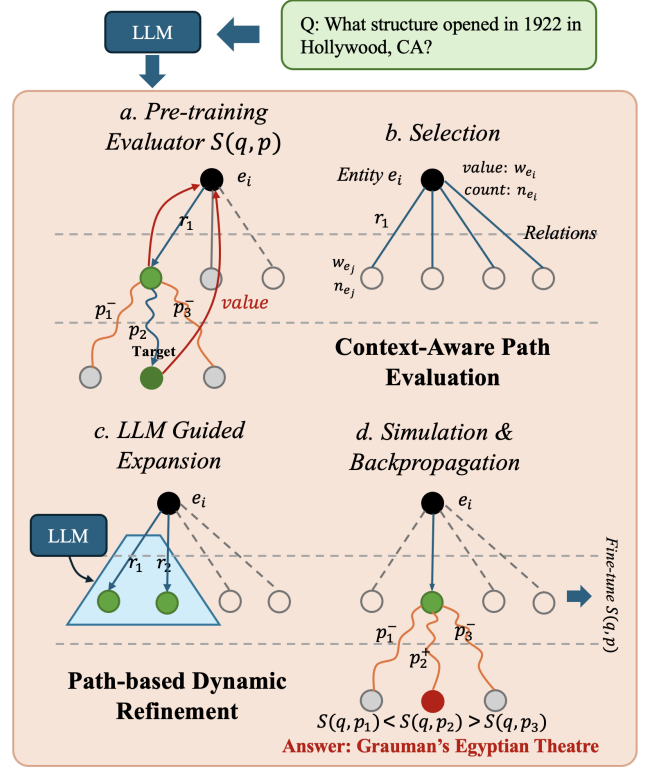


Figure 1: Overview of DAMR. The reasoning process begins with an MCTS guided by an LLM-based planner, which selects top- $k$  semantically relevant relations at each expansion step. A context-aware path evaluator scores each candidate path during simulation. To enable continual adaptation, high-confidence pseudo-paths generated during search are used to dynamically fine-tune the evaluator.

enables context-sensitive evaluation and enhances the accuracy and relevance of multi-hop reasoning; (3) **Path-based Dynamic Refinement.** DAMR uses intermediate paths from MCTS as dynamic pseudo-paths to iteratively fine-tune the path evaluator, enhancing its ability to capture question-specific semantics and improving reasoning accuracy.

### LLM Guided Expansion

A key challenge in KGQA is efficiently exploring the vast search space of multi-hop reasoning paths, especially under weak or no supervision. Existing methods often struggle to balance search efficiency and semantic relevance, resulting in either redundant exploration or missed correct paths. To address this, the LLM-Guided Expansion module employs MCTS (Kocsis and Szepesvári 2006) as the backbone for symbolic path expansion. At each step, an LLM proposes semantically relevant relations, narrowing the search space and improving path quality, while MCTS ensures a balanced trade-off between exploration and exploitation.

Specifically, each node in the MCTS represents a reasoning state anchored at a specific entity in the KG. Given the current state, possible actions correspond to selecting an out-

going relation to extend the reasoning path. During the **Selection** phase, nodes are scored using the UCT in Eq. (1), guiding the search to balance exploration and exploitation.

In the **Expansion** phase, we employ an LLM guided strategy to prioritize semantically meaningful path extensions. Given a specific entity  $e_i$  in KG, we retrieve its associated outgoing relations  $\mathcal{R}_{e_i} = \{r_1, r_2, \dots, r_n\}$ . To focus the search on meaningful directions, we prompt an LLM with the question  $q$  and the candidate relations  $\mathcal{R}_{e_i}$ , selecting the top- $k$  relations most aligned with the question:

$$\mathcal{R}_{top-k} = \text{LLM}(q, \mathcal{R}_{e_i}). \quad (2)$$

These selected relations are then used to expand the current node. This LLM-guided expansion significantly reduces unnecessary branching and ensures that the search remains semantically focused and computationally efficient.

### Context-Aware Path Evaluation

While LLM-Guided Expansion effectively narrows the search space by selecting semantically relevant relations, it does not guarantee that all expanded paths are correct or meaningful in the broader reasoning context. As the search progresses, path semantics evolve dynamically, and early promising paths may later become irrelevant or misleading. To address this, Context-Aware Path Evaluation integrates a lightweight Transformer-based path scorer into the simulation phase of MCTS. This scorer leverages cross-attention to jointly encode the question and the current reasoning path, allowing for adaptive evaluation that captures evolving semantics. By assigning scores to simulated paths based on question-path alignment, this module enables more accurate path ranking throughout the search process.

**Context-Aware Path Evaluator.** Specifically, in the **Simulation** phase, we evaluate the quality of each candidate path constructed during MCTS rollouts. Given a question  $q$  and a candidate relation path  $p_r = (r_1, r_2, \dots, r_l)$ , where  $p_r$  is formed by sequentially selecting relations during the expansion steps, we first encode both the question and the path using a pre-trained LLM. Let  $\mathbf{z}_q \in \mathbb{R}^d$  denote the embedding of the question and  $\mathbf{z}_{r_i} \in \mathbb{R}^d$  denote the embedding of relation  $r_i$ . To capture the sequential structure of relation paths, we incorporate a learnable position encoding  $\mathbf{e}_i^{\text{pos}}$  for each relation  $r_i$ . The final input sequence is constructed by combining each relation embedding  $\mathbf{z}_{r_i}$  with positional encoding and feeding it into a Transformer encoder:

$$\mathbf{E}_{p_r} = \text{Transformer}([\mathbf{z}_{r_1} + \mathbf{e}_1^{\text{pos}}, \dots, \mathbf{z}_{r_l} + \mathbf{e}_l^{\text{pos}}]),$$

where  $\mathbf{e}_i^{\text{pos}} = \mathbf{E}^{\text{pos}}[i]$  denotes the relative position encoding for the  $i$ -th hop in the path, drawn from a trainable embedding matrix  $\mathbf{E}^{\text{pos}} \in \mathbb{R}^{L \times d}$ , with  $L$  as the maximum path length and  $d$  as the embedding dimension. To further incorporate question-specific information, we apply a cross-attention mechanism, allowing the encoded path representation  $\mathbf{E}_{p_r}$  to attend to the question embedding  $\mathbf{z}_q$ :

$$\mathbf{H} = \mathbf{E}_{p_r} + \text{CrossAttn}(\mathbf{E}_{p_r}, \mathbf{z}_q),$$

with  $\text{CrossAttn}(\mathbf{E}_{p_r}, \mathbf{z}_q) = \text{softmax}(\mathbf{E}_{p_r} \cdot \mathbf{z}_q^T / \sqrt{d_k}) \cdot \mathbf{z}_q$ .

We then employ attention pooling over relation representation  $\mathbf{H}$  to obtain the hidden states of relation path:

$$\mathbf{s}_{p_r} = \sum_{i=1}^l \alpha_i \mathbf{h}_i, \quad \alpha = \text{Softmax}(\text{MLP}(\mathbf{H})),$$

where  $\mathbf{h}_i$  denotes the hidden state of the  $i$ -th relation and  $\alpha_i$  is its learned attention weight. This pooling mechanism enables the model to selectively emphasize informative steps along the reasoning path. Finally, the pooled path representation  $\mathbf{s}_{p_r}$  is concatenated with the question embedding  $\mathbf{z}_q$ , and the combined vector is fed into a multi-layer perceptron to compute the plausibility score of the question-path pair:

$$S(q, p_r) = \text{MLP}([\mathbf{s}_{p_r}; \mathbf{z}_q]). \quad (3)$$

This context-aware evaluation model dynamically scores partial reasoning paths by jointly considering the question and the relation sequence, offering accurate and context-sensitive guidance to the MCTS search process.

**Pre-training of Evaluator.** To train the context-aware path evaluation model, we construct supervision signals by generating positive and negative relation paths from local subgraphs. A path is labeled positive if it connects the head entity to a correct answer entity within a predefined hop limit. Negative paths are drawn from two sources: hard negatives that end near but do not reach the answer, and random negatives obtained via walks that avoid answer entities entirely. Each training instance is a triplet  $(q, p^+, p^-)$ , and sequences are zero-padded with attention masks for efficient batch training.

The model computes a plausibility score  $S(q, p)$  for each question-path pair and is optimized using the Pair-wise Ranking loss to encourage higher scores for positive paths:

$$\mathcal{L}_{\text{PR}} = -\frac{1}{M} \sum_{i=1}^M \log \sigma(S(q, p_i^+) - S(q, p_i^-)), \quad (4)$$

where  $\sigma(\cdot)$  is the sigmoid function. This training strategy equips the evaluator with the ability to distinguish plausible reasoning paths, thereby improving the guidance signal during MCTS-based inference.

### Path-based Dynamic Refinement

While LLM-guided expansion and semantic scoring improve path exploration, the static evaluator may fail to generalize to the evolving search space. To address this, we introduce a dynamic refinement mechanism that leverages high-confidence paths from MCTS rollouts as pseudo-paths. These pseudo-paths serve as supervision signals, enabling continual adaptation of the evaluator to new reasoning contexts without requiring additional labeled data.

Specifically, during **Backpropagation** phase, the plausibility score estimated by the context-aware path evaluator is propagated along the visited nodes in the MCTS tree after each simulation. For every entity  $e_i$  on the simulated path, we update its visit count and aggregated value as follows:

$$n_{e_i} = n_{e_i} + 1, \quad w_{e_i} = \frac{\sum_j n_{e_j} \cdot w_{e_j}}{\sum_j n_{e_j}}, \quad (5)$$

where  $n_{e_i}$  is the visit count and  $w_{e_i}$  is the aggregated value of entity  $e_i$ . The value is computed as a weighted average over its child nodes  $\{e_j\}$ , and reflects the plausibility scores  $w_{e_j}$  assigned during simulation. These updates refine the UCT estimates used in future selection steps, progressively biasing the search toward high-quality reasoning paths.

To construct supervision signals for fine-tuning, we dynamically sample pseudo-path pairs  $(\hat{p}_i, \hat{p}_j)$  from the set of explored paths during MCTS. Instead of relying on the evaluator’s predictions, we assign pseudo-labels based on empirically grounded values derived from the search process. Specifically, for entity  $e_i$  along a reasoning path  $p_r$ , we define its search value as:  $w_{e_i} = \frac{w_{p_r}}{n_{e_i}}$ , where  $w_{p_r}$  is the cumulative reward from all rollouts passing through  $p_r$ , and  $n_{e_i}$  is the visit count of entity  $e_i$ . Given a pair of paths, we assign pseudo-labels based on their relative values:

$$(\hat{p}^+, \hat{p}^-) = \begin{cases} (p'_i, p'_j), & \text{if } w_{e_i} > w_{e_j}, \\ (p'_j, p'_i), & \text{otherwise.} \end{cases} \quad (6)$$

The path evaluator is then fine-tuned using the PR loss in Eq. (4), encouraging higher scores for more promising paths.

## Reasoning Process

The overall reasoning process is summarized in Appendix A. The framework begins by initializing the path evaluation model to distinguish between plausible and implausible reasoning paths derived from the knowledge graph, establishing a strong foundation for downstream search. During the dynamic MCTS process, the algorithm iteratively performs selection, expansion, simulation, and backpropagation. In the expansion step, an LLM-based planner adaptively selects the top- $k$  relations most relevant to the question, effectively steering the search toward semantically meaningful paths. The path evaluation model informs the simulation phase by prioritizing trajectories that are more likely to yield correct answers. To enable continual adaptation, pseudo-path pairs obtained during search are periodically used to refine the evaluator. Finally, entities reached by high-scoring reasoning paths are aggregated to construct the answer set.

## Experiments

### Experimental Settings

**Datasets.** To evaluate the effectiveness of DAMR, we conduct experiments on two widely used KGQA benchmarks: WebQSP (Talmor and Berant 2018) and CWQ (Yih et al. 2016). Following prior work (Sun et al. 2023; Liu et al. 2025), we uniformly sample 1,000 questions from the test sets of both datasets to evaluate the performance. More details about datasets are provided in Appendix C.

**Baselines.** We compare DAMR with a comprehensive set of baselines. These baselines include: the semantic parsing methods, e.g., KV-Mem (Miller et al. 2016), Embed-KGQA (Saxena, Tripathi, and Talukdar 2020), QGG (Lan and Jiang 2020), NSM (He et al. 2021), TransferNet (Shi et al. 2021), and KGT5 (Saxena, Kochsiek, and Gemulla 2022); the retrieval-based methods, e.g., GraftNet (Sun et al. 2018), PullNet (Sun, Bedrax-Weiss, and Cohen 2019),

Table 1: Performance comparison (%) on WebQSP and CWQ datasets. **Bold** results indicate the best performance.

Type	Methods	WebQSP		CWQ	
		Hits@1	F1	Hits@1	F1
Semantic Parsing	KV-Mem	46.7	34.5	18.4	15.7
	EmbedKGQA	66.6	-	45.9	-
	QGG	73.0	73.8	36.9	37.4
	NSM	68.7	62.8	47.6	42.4
	TransferNet	71.4	-	48.6	-
	KGT5	56.1	-	36.5	-
Retrieval	GraftNet	66.4	60.4	36.8	32.7
	PullNet	68.1	-	45.9	-
	SR+NSM	68.9	64.1	50.2	47.1
	SR+NSM+E2E	69.5	64.1	49.3	46.3
LLMs	Flan-T5-xl	31.0	-	14.7	-
	Alpaca-7B	51.8	-	27.4	-
	Llama3-8B	30.3	25.7	30.5	27.8
	Qwen2.5-7B	28.4	23.7	25.9	24.1
	ChatGPT	66.8	-	39.9	-
	ChatGPT+CoT	75.6	-	48.9	-
LLMs+KGs	UniKGQA	77.2	72.2	51.2	49.0
	DECAF	82.1	78.8	70.4	-
	KD-CoT	68.6	52.5	55.7	-
	Nutrea	77.4	72.7	53.6	49.5
	ToG	81.9	76.0	68.5	60.2
	RoG	80.8	70.8	57.8	56.2
	KAPING	72.4	65.1	53.4	50.3
	ReasoningLM	78.5	71.0	69.0	64.0
	FiDeLis	84.3	78.3	71.5	64.3
	GNN-RAG	80.8	70.8	57.8	56.2
	DoG	65.4	55.6	41.0	46.4
	DualR	81.5	71.6	65.3	62.1
	DP	87.5	81.4	75.8	69.4
	RwT	87.0	79.7	72.4	66.7
	<b>DAMR</b>	<b>94.0</b>	<b>81.7</b>	<b>78.0</b>	<b>75.1</b>

SR+NSM (Zhang et al. 2022), and SR+NSM+E2E (Zhang et al. 2022); the general LLMs, including Flan-T5-xl (Chung et al. 2024), Alpaca-7B (Taori et al. 2023), Llama3-8B (Dubey et al. 2024), Qwen2.5-7B (Team 2024), ChatGPT (Schulman et al. 2022), and ChatGPT+CoT (Wei et al. 2022); and recent LLMs with KG methods, including UniKGQA (Jiang et al. 2022), DECAF (Yu et al. 2022), KD-CoT (Wang et al. 2023a), Nutrea (Choi et al. 2023), ToG (Sun et al. 2023), RoG (Luo et al. 2023), KAPING (Baek, Aji, and Saffari 2023), ReasoningLM (Jiang et al. 2023), FiDeLis (Sui et al. 2024), GNN-RAG (Mavromatis and Karypis 2024), DoG (Ma et al. 2025a), DualR (Liu et al. 2025), DP (Ma et al. 2025b), and RwT (Shen et al. 2025). More introductions are provided in Appendix D.

**Implementation Details.** We implement the DAMR framework using PyTorch, and all experiments are conducted on NVIDIA A100 GPUs. The LLM-based planner is implemented with GPT-4.1 (Liu et al. 2023), while question and relation embeddings are generated from Qwen3-8B (Yang et al. 2025) with an embedding dimension of 1024. For the path evaluation module, we use a 128-dimensional embed-

Table 2: Statistics of average number of LLM calls and token consumption per question on WebQSP and CWQ datasets.

Method	WebQSP		CWQ	
	#Tokens	#Calls	#Tokens	#Calls
DoG	22,538	30.9	37,741	58.1
ToG	16,372	23.2	26,183	41.9
RwT	10,680	15.1	17,885	28.6
DAMR	3,931	7.1	9,266	16.8

ding and employ the Adam optimizer with a learning rate of  $1 \times 10^{-4}$  during pretraining and  $1 \times 10^{-5}$  during fine-tuning. The model consists of two Transformer layers and is trained for 15 epochs in the pretraining stage and 10 epochs in the fine-tuning stage. Following (Luo et al. 2023; Yao et al. 2025; Ma et al. 2025b), we evaluate DAMR using Hits@1 and F1 score, assessing answer correctness and overall accuracy for questions with potentially multiple correct answers.

### Performance Comparison

We report the experimental results of DAMR in Table 1, benchmarking its performance against state-of-the-art baselines across KGQA datasets. From the results, we find that: (1) Semantic parsing and retrieval-based methods serve as early foundations for KGQA by extracting subgraphs and capturing structural semantics. However, embedding-based models struggle with complex relational patterns, while retrieval-based methods rely on rigid pipelines that limit generalization. In contrast, LLM with KG approaches combine the language understanding of LLMs with structured reasoning over KGs, enabling more flexible path exploration and improved adaptability to diverse, multi-hop queries. (2) General-purpose LLMs, such as ChatGPT and Alpaca-7B, show basic reasoning ability but often perform worse than methods that combine LLMs with KGs in KGQA tasks. This is mainly because they are not grounded in domain-specific knowledge, making them more likely to produce incorrect or made-up answers. (3) DAMR consistently outperforms all baselines across both datasets, showcasing its strong reasoning capability. This superior performance is driven by its integration of an LLM-based planner, which selectively retrieves relevant relations to reduce noise and guide the search toward high-quality reasoning paths, and a path evaluation model that is dynamically fine-tuned during search to capture semantic differences among candidate paths and accurately rank those most likely to yield correct answers.

### Efficiency Analysis

As shown in Table 2, DAMR achieves substantial improvements in computational efficiency. It reduces the average number of LLM calls to 7.1 on WebQSP and 16.8 on CWQ, with corresponding token usage of 3,931 and 9,266. These correspond to reductions of over 50% in LLM calls and 75% in token consumption relative to the strongest baseline. This efficiency is achieved by invoking the LLM only during the expansion phase of MCTS to select the top- $k$  semantically relevant relations, which effectively narrows

Table 3: The results of ablation studies on the WebQSP and CWQ datasets. **Bold** results indicate the best performance.

Method	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
DAMR w/o PE	91.2	78.2	74.3	72.1
DAMR w/o FT	91.9	80.1	75.1	73.0
DAMR w/ GPT 4.1	92.5	79.8	74.9	72.4
DAMR	<b>94.0</b>	<b>81.7</b>	<b>78.0</b>	<b>75.1</b>

the search space and avoids redundant reasoning steps that lead to unnecessary computational overhead. During simulation, the context-aware path evaluator efficiently assesses candidate paths based on question-path alignment without requiring any further LLM interaction or model inference. These design choices reduce both the frequency and verbosity of LLM usage while maintaining strong reasoning performance, making DAMR more efficient, scalable, and practically deployable than previous work.

### Ablation Study

We conduct ablation studies to examine the key components in DAMR: (1) DAMR w/o PE: It removes the path evaluation module; (2) DAMR w/o FT: disables the fine-tuning mechanism for the path evaluation module; (3) DAMR w/ GPT 4.1: replaces the context-aware path evaluation module with a general LLM.

Experimental results are summarized in Table 3. From the results, we find that: (1) Removing the path evaluation module (DAMR w/o PE) leads to a noticeable performance drop on both datasets, highlighting its critical role in guiding the search process. Without this component, the model cannot effectively assess or rank candidate paths, leading to sub-optimal reasoning and degraded answer accuracy. (2) Compared to DAMR w/o FT, the proposed DAMR consistently achieves superior results on both datasets, highlighting the importance of the finetuning mechanism in the path evaluation module. This mechanism enables the model to adapt to the evolving distribution of explored paths, improving its ability to distinguish between plausible and implausible reasoning trajectories. (3) Replacing the context-aware path evaluation module with general LLMs leads to degraded performance, confirming the advantage of our fine-tuned path scorer. By capturing fine-grained semantic distinctions among candidate paths, it provides more accurate evaluation signals, thereby enhancing the overall search effectiveness.

### Sensitivity Analysis

We conduct a sensitivity analysis to assess the impact of two key hyperparameters in DAMR: the number of selected relations  $k$  and the maximum reasoning path length  $L$ . The parameter  $k$  controls how many relations are proposed by the LLM-based planner at each step, while  $L$  determines the number of reasoning hops allowed during path construction.

Figure 2 illustrates how  $k$  and  $L$  affect the performance of DAMR on the WebQSP and CWQ datasets. We vary  $k$  and

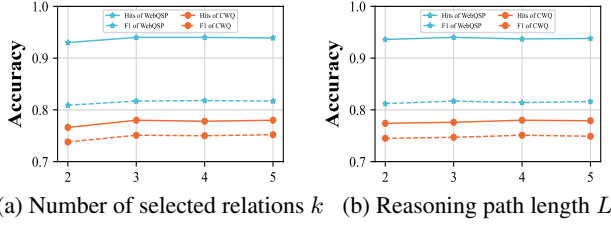


Figure 2: Sensitivity analysis of hyperparameter on the WebQSP and CWQ datasets.

Table 4: Performance of DAMR using different LLM-based planners as backbones on the WebQSP and CWQ datasets. **Bold** values denote the best results.

Method	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
DAMR (Llama2-13B)	91.0	76.7	73.9	69.5
DAMR (Qwen3-14B)	91.5	77.8	74.4	70.1
DAMR (GPT 4.1-mini)	93.1	80.6	76.1	72.7
DAMR (GPT 4.1)	<b>94.0</b>	<b>81.7</b>	<b>78.0</b>	<b>75.1</b>

Table 5: Case study of DAMR. We highlight the correct answers in **Bold** and the wrong answers in underline.

Question	What structure opened in 1922 in Hollywood, CA?
Answer	<b>Grauman’s Egyptian Theatre</b>
Llama-2-13B	<u>The Grauman’s Chinese Theatre</u> , a historic movie palace, opened on November 18, 1922, in Hollywood, CA. It was built by Sid Grauman and Charles E. Toberman and has since become a famous landmark and tourist attraction.
Qwen-3-14B	The TCL Chinese Theatre, originally known as the Chinese Grand Theatre, opened in 1922 in Hollywood, CA. It is a historic movie theater renowned for its unique architecture and celebrity handprints and footprints.
GPT 4.1	The Hollywood Bowl, an iconic amphitheater in Hollywood, California, opened in 1922. It is renowned for its distinctive band shell and has hosted numerous concerts and events, becoming a significant cultural landmark in the area.
GPT 4.1-mini	The Hollywood Bowl, an iconic amphitheater in Hollywood, California, opened in 1922 and has since been a renowned venue for music performances and cultural events.
DAMR	Path 1: Entity (id: 83076) → location.location.events → time.event.locations → travel.travel_destination.tourist_attractions → <b>Grauman’s Egyptian Theatre</b> . Path 2: Entity (id: 83076) → travel.travel_destination.tourist_attractions → <b>Grauman’s Egyptian Theatre</b> .

$L$  within the range of  $\{2, 3, 4, 5\}$ . From the results, we observe that: (1) As shown in Figure 2(a), increasing  $k$  initially leads to performance gains, which then stabilize before experiencing a slight decline. While larger  $k$  values encourage broader relational exploration, they may also introduce irrelevant candidates and increased computational cost. Conversely, smaller  $k$  restrict the diversity of the search. To balance these trade-offs, we select a moderate  $k = 3$  as the default setting. (2) As shown in Figure 2(b), on the WebQSP dataset, performance improves from  $L = 2$  to 3, then fluctuates between  $L = 3$  and 5, suggesting limited gains beyond three hops. In contrast, performance on the CWQ dataset steadily increases up to  $L = 4$  before slightly declining at  $L = 5$ , reflecting its need for deeper reasoning due to more complex questions. Balancing effectiveness and efficiency across both datasets, we set  $L = 4$  as the default path length in all experiments. More results are provided in Appendix E.

### Impact of Different LLMs

To evaluate the impact of different LLM-based planners within the DAMR framework, we compare several backbones including Llama2 13B (Roque 2025), Qwen3 14B (Team 2024), GPT 4.1 mini, and GPT 4.1, as shown in Table 4. Across both datasets, stronger LLMs consistently yield higher F1 and Hits scores, with GPT 4.1 achieving the best performance on all metrics. This highlights the critical role of advanced LLMs in guiding relation selection and

reasoning path expansion. The results confirm that improved language modeling and semantic understanding capabilities directly enhance KGQA accuracy. Overall, these findings emphasize the importance of backbone selection and further validate the design of DAMR, which leverages powerful LLMs for robust and effective multi-hop reasoning.

### Case study

Table 5 presents a case study comparing the reasoning process of DAMR with four general LLMs: Llama-2-13B, Qwen-3-14B, GPT 4.1-mini, and GPT 4.1. While all baseline LLMs fail to identify the correct structure that opened in Hollywood in 1922, the proposed DAMR accurately finds **Grauman’s Egyptian Theatre** by explicitly traversing relation paths in the knowledge graph from two different reasoning paths. This example demonstrates that, although LLMs appear capable of answering the question, their responses can still be factually incorrect due to a lack of grounded knowledge. In contrast, DAMR consistently produces accurate and faithful answers by grounding its reasoning in KG and explicitly modeling reasoning paths. More studies can be found in Appendix E.

### Conclusion

In this work, we present DAMR, a dynamically adaptive MCTS-based reasoning framework for complex KGQA tasks. DAMR incorporates an LLM-based planner to guide



top- $k$  relation expansion, a context-aware path evaluator to assess reasoning paths without further LLM queries, and a dynamic refinement module that continually adapts the evaluator using pseudo-path supervision from MCTS rollouts. This modular design enables efficient yet accurate multi-hop reasoning by narrowing the search space, reducing redundant computation, and enhancing evaluation quality. Extensive experiments on WebQSP and CWQ confirm the effectiveness and efficiency of DAMR, making it a practical and scalable solution for real-world KGQA deployment.

## References

- Baek, J.; Aji, A. F.; and Saffari, A. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.
- Chang, J. D.; Brantley, K.; Ramamurthy, R.; Misra, D.; and Sun, W. 2023. Learning to generate better than your llm. *arXiv preprint arXiv:2306.11816*.
- Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3): 1–45.
- Chen, X.; Wang, Y.; Fang, J.; Meng, Z.; and Liang, S. 2023. Heterogeneous graph contrastive learning with metapath-based augmentations. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(1): 1003–1014.
- Choi, H. K.; Lee, S.; Chu, J.; and Kim, H. J. 2023. Nutrea: Neural tree search for context-guided multi-hop kgqa. *Proceedings of the Conference on Neural Information Processing Systems*, 36: 35954–35965.
- Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.
- Dammu, P. P. S.; Naidu, H.; and Shah, C. 2025. Dynamic-kgqa: A scalable framework for generating adaptive question answering datasets. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3498–3508.
- Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; and McCallum, A. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.
- Didolkar, A.; Goyal, A.; Ke, N. R.; Guo, S.; Valko, M.; Lillicrap, T.; Jimenez Rezende, D.; Bengio, Y.; Mozer, M. C.; and Arora, S. 2024. Metacognitive capabilities of llms: An exploration in mathematical problem solving. *Proceedings of the Conference on Neural Information Processing Systems*, 37: 19783–19812.
- Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv:2407.
- Fang, S.; Ma, K.; Zheng, T.; Du, X.; Lu, N.; Zhang, G.; and Tang, Q. 2024. KARPA: A Training-free Method of Adapting Knowledge Graph as References for Large Language Model’s Reasoning Path Aggregation. *arXiv preprint arXiv:2412.20995*.
- He, G.; Lan, Y.; Jiang, J.; Zhao, W. X.; and Wen, J.-R. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the International ACM Conference on Web Search & Data Mining*, 553–561.
- Huang, J.; Chen, R.; Li, Z.; Gao, Z.; He, X.; Guo, Y.; Gong, M.; and Liu, T. 2025a. MLLM-For3D: Adapting Multimodal Large Language Model for 3D Reasoning Segmentation. *arXiv preprint arXiv:2503.18135*.
- Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2025b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55.
- Jiang, J.; Zhou, K.; Zhao, W. X.; Li, Y.; and Wen, J.-R. 2023. Reasoninglm: Enabling structural subgraph reasoning in pre-trained language models for question answering over knowledge graph. *arXiv preprint arXiv:2401.00158*.
- Jiang, J.; Zhou, K.; Zhao, W. X.; and Wen, J.-R. 2022. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. *arXiv preprint arXiv:2212.00959*.
- Kocsis, L.; and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, 282–293. Springer.
- Lan, Y.; and Jiang, J. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
- Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2, 896. Atlanta.
- Li, Y.; Song, D.; Zhou, C.; Tian, Y.; Wang, H.; Yang, Z.; and Zhang, S. 2024. A Framework of Knowledge Graph-Enhanced Large Language Model Based on Question Decomposition and Atomic Retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 11472–11485.
- Liu, G.; Zhang, Y.; Li, Y.; and Yao, Q. 2025. Dual reasoning: A gnn-llm collaborative framework for knowledge graph question answering. In *The Second Conference on Parsimony and Learning (Proceedings Track)*.
- Liu, H.; Ning, R.; Teng, Z.; Liu, J.; Zhou, Q.; and Zhang, Y. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.
- Long, X.; Zhuang, L.; Shen, C.; Yan, S.; Li, Y.; and Wang, S. 2025. Enhancing Large Language Models with Reward-guided Tree Search for Knowledge Graph Question and Answering. *arXiv preprint arXiv:2505.12476*.



- Luo, L.; Li, Y.-F.; Haffari, G.; and Pan, S. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.
- Ma, H.; Hu, T.; Pu, Z.; Boyin, L.; Ai, X.; Liang, Y.; and Chen, M. 2024. Coevolving with the other you: Fine-tuning llm with sequential cooperative multi-agent reinforcement learning. *Proceedings of the Conference on Neural Information Processing Systems*, 15497–15525.
- Ma, J.; Gao, Z.; Chai, Q.; Sun, W.; Wang, P.; Pei, H.; Tao, J.; Song, L.; Liu, J.; Zhang, C.; et al. 2025a. Debate on graph: a flexible and reliable reasoning framework for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 23, 24768–24776.
- Ma, J.; Qu, N.; Gao, Z.; Xing, R.; Liu, J.; Pei, H.; Xie, J.; Song, L.; Wang, P.; Tao, J.; et al. 2025b. Deliberation on Priors: Trustworthy Reasoning of Large Language Models on Knowledge Graphs. *arXiv preprint arXiv:2505.15210*.
- Mavromatis, C.; and Karypis, G. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.
- Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.
- Pei, Q.; Wu, L.; Pan, Z.; Li, Y.; Lin, H.; Ming, C.; Gao, X.; He, C.; and Yan, R. 2025. MathFusion: Enhancing Mathematical Problem-solving of LLM through Instruction Fusion. *arXiv preprint arXiv:2503.16212*.
- Roque, L. 2025. The Evolution of Llama: From Llama 1 to Llama 3.1.
- Saxena, A.; Kochsiek, A.; and Gemulla, R. 2022. Sequence-to-sequence knowledge graph completion and question answering. *arXiv preprint arXiv:2203.10321*.
- Saxena, A.; Tripathi, A.; and Talukdar, P. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 4498–4507.
- Schulman, J.; Zoph, B.; Kim, C.; Hilton, J.; Menick, J.; Weng, J.; Uribe, J. F. C.; Fedus, L.; Metz, L.; Pokorny, M.; et al. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2(4).
- Shen, T.; Wang, J.; Zhang, X.; and Cambria, E. 2025. Reasoning with Trees: Faithful Question Answering over Knowledge Graph. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 3138–3157.
- Shi, J.; Cao, S.; Hou, L.; Li, J.; and Zhang, H. 2021. Transfernet: An effective and transparent framework for multi-hop question answering over relation graph. *arXiv preprint arXiv:2104.07302*.
- Sui, Y.; He, Y.; Liu, N.; He, X.; Wang, K.; and Hooi, B. 2024. Fidelis: Faithful reasoning in large language model for knowledge graph question answering. *arXiv preprint arXiv:2405.13873*.
- Sun, H.; Bedrax-Weiss, T.; and Cohen, W. W. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. *arXiv preprint arXiv:1904.09537*.
- Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; Salakhutdinov, R.; and Cohen, W. W. 2018. Open domain question answering using early fusion of knowledge bases and text. *arXiv preprint arXiv:1809.00782*.
- Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L. M.; Shum, H.-Y.; and Guo, J. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.
- Talmor, A.; and Berant, J. 2018. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*.
- Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford alpaca: An instruction-following llama model.
- Team, Q. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Toroghi, A.; Guo, W.; Pesaranhader, A.; and Sanner, S. 2024. Verifiable, Debuggable, and Repairable Commonsense Logical Reasoning via LLM-based Theory Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Wang, F.; Wang, Y.; Li, D.; Gu, H.; Lu, T.; Zhang, P.; and Gu, N. 2022a. Enhancing CTR prediction with context-aware feature representation learning. In *Proceedings of the International ACM SIGIR Conference on Research & Development in Information Retrieval*, 343–352.
- Wang, K.; Duan, F.; Wang, S.; Li, P.; Xian, Y.; Yin, C.; Rong, W.; and Xiong, Z. 2023a. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*.
- Wang, M.; Su, H.; Wang, S.; Wang, S.; Yin, N.; Shen, L.; Lan, L.; Yang, L.; and Cao, X. 2025. Graph Convolutional Mixture-of-Experts Learner Network for Long-Tailed Domain Generalization. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Wang, Y.; Chen, X.; Fang, J.; Meng, Z.; and Liang, S. 2023b. Enhancing conversational recommendation systems with representation fusion. *ACM Transactions on the Web*, 17(1): 1–34.
- Wang, Y.; Liang, V.; Yin, N.; Liu, S.; and Segal, E. 2024a. SGAC: A Graph Neural Network Framework for Imbalanced and Structure-Aware AMP Classification. *arXiv preprint arXiv:2412.16276*.
- Wang, Y.; Liu, S.; Wang, M.; Liang, S.; and Yin, N. 2024b. Degree distribution based spiking graph networks for domain adaptation. *arXiv e-prints*, arXiv–2410.
- Wang, Y.; Wang, H.; Shen, Y.; Fei, J.; Li, W.; Jin, G.; Wu, L.; Zhao, R.; and Le, X. 2022b. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4248–4257.
- Wang, Y.; Wang, M.; Manzoor, M. A.; Liu, F.; Georgiev, G.; Das, R. J.; and Nakov, P. 2024c. Factuality of large language models: A survey. *arXiv preprint arXiv:2402.02420*.
- Wang, Y.; Yin, N.; Xiao, M.; Yi, X.; Liu, S.; and Liang, S. 2024d. Dusego: Dual second-order equivariant graph ordinary differential equation. *arXiv preprint arXiv:2411.10000*.

- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Proceedings of the Conference on Neural Information Processing Systems*, 35: 24824–24837.
- Xie, Q.; Luong, M.-T.; Hovy, E.; and Le, Q. V. 2020. Self-training with noisy student improves imagenet classification. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10687–10698.
- Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 564–573.
- Xu, M.; Chen, K.; Bai, X.; Yang, M.; Zhao, T.; and Zhang, M. 2024. Llm-based discriminative reasoning for knowledge graph question answering. *arXiv preprint arXiv:2412.12643*.
- Xu, M.; Liang, G.; Chen, K.; Wang, W.; Zhou, X.; Yang, M.; Zhao, T.; and Zhang, M. 2025. Memory-augmented query reconstruction for llm-based knowledge graph reasoning. *arXiv preprint arXiv:2503.05193*.
- Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Yao, T.; Li, H.; Shen, Z.; Li, P.; Liu, T.; and Zhang, K. 2025. Learning Efficient and Generalizable Graph Retriever for Knowledge-Graph Question Answering. *arXiv preprint arXiv:2506.09645*.
- Yao, T.; Wang, Y.; Zhang, K.; and Liang, S. 2023. Improving the expressiveness of k-hop message-passing gnns by injecting contextualized substructure information. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3070–3081.
- Yih, W.-t.; Richardson, M.; Meek, C.; Chang, M.-W.; and Suh, J. 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)*, 201–206.
- Yin, N.; Feng, F.; Luo, Z.; Zhang, X.; Wang, W.; Luo, X.; Chen, C.; and Hua, X.-S. 2022. Dynamic hypergraph convolutional network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 1621–1634. IEEE.
- Yin, N.; Wan, M.; Shen, L.; Patel, H. L.; Li, B.; Gu, B.; and Xiong, H. 2024a. Continuous spiking graph neural networks. *arXiv preprint arXiv:2404.01897*.
- Yin, N.; Wang, M.; Chen, Z.; De Masi, G.; Xiong, H.; and Gu, B. 2024b. Dynamic spiking graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16495–16503.
- Yin, N.; Wang, M.; Chen, Z.; Shen, L.; Xiong, H.; Gu, B.; and Luo, X. 2024c. DREAM: Dual structured exploration with mixup for open-set graph domain adaption. In *Proceedings of the International Conference on Learning Representations*.
- Yu, D.; Zhang, S.; Ng, P.; Zhu, H.; Li, A. H.; Wang, J.; Hu, Y.; Wang, W.; Wang, Z.; and Xiang, B. 2022. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. *arXiv preprint arXiv:2210.00063*.
- Zhai, S.; Bai, H.; Lin, Z.; Pan, J.; Tong, P.; Zhou, Y.; Suhr, A.; Xie, S.; LeCun, Y.; Ma, Y.; et al. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37: 110935–110971.
- Zhang, J.; Zhang, X.; Yu, J.; Tang, J.; Tang, J.; Li, C.; and Chen, H. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. *arXiv preprint arXiv:2202.13296*.
- Zhao, W.; Liu, Y.; Niu, T.; Wan, Y.; Yu, P. S.; Joty, S.; Zhou, Y.; and Yavuz, S. 2023. DIVKNOWQA: assessing the reasoning ability of llms via open-domain question answering over knowledge base and text. *arXiv preprint arXiv:2310.20170*.

## A. Algorithm

Algorithm 1: Dynamic MCTS-based KGQA with Path Model Pretraining and Online Refinement

---

**Input:** Question  $q$ , knowledge graph  $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ , number of selected relations  $k$ , MCTS iterations  $N$ , length of reasoning path  $L$

**Output:** Answer set  $\mathcal{A}$

```

1: / Stage 1: Path Evaluation Model Pre-training /
2: Construct reasoning path pairs  $(q, p^+, p^-)$  from  $\mathcal{G}$ 
3: Initialize path evaluation model  $S(q, \cdot; \Theta)$ 
4: for each batch in pretraining data do
5:   Update  $S(q, \cdot; \Theta)$  by minimizing the Pair-wise Ranking loss in Eq.(4)
6: end for
7: / Stage 2: Dynamic MCTS Reasoning /
8: for  $i = 1$  to  $N$  do
9:   Selection: Traverse the tree from root to a leaf node by selecting child nodes according to the UCT criterion in Eq.(1)
10:  Expansion:
11:    i. At the selected node, enumerate all candidate relations from current entities and use the LLM-based planner to select the top- $k$  most relevant relations
12:    ii. Expand a new child node for each selected relation
13:  Simulation: For each expanded node, perform a rollout by sequentially selecting relations (guided by the path evaluation model) up to  $L$  hops or until a correct answer is reached
14:  Backpropagation: Update the value ( $w_i$ ) and visit ( $n_i$ ) statistics along the traversed path from the leaf node back to the root using the score from the simulation, as per Eq.(6)
15:  Path Evaluation Model Fine-tuning: Generate the explored pseudo-path pairs  $(\hat{p}^+, \hat{p}^-)$  via Eq.(5) and fine-tune the path evaluation model  $S(q, \cdot; \Theta)$  based on Pair-wise Ranking loss in Eq.(4)
16: end for
17: / Stage 3: Answer Extraction /
18: Collect entities reached by high-scoring reasoning paths as  $\mathcal{A}$ 
19: return  $\mathcal{A}$ 

```

---

## B. Complexity Analysis

The overall time complexity of the proposed DAMR framework is governed by its two primary online stages: the LLM-Guided MCTS Search and the interleaved Path-based Dynamic Refinement. In the MCTS Search phase, executed over  $N$  iterations, the LLM-guided relation expansion incurs a total complexity of  $\mathcal{O}(N \cdot T_{\text{LLM}}(k))$ , where  $T_{\text{LLM}}(k)$  denotes the inference time of the LLM when provided with up to  $k$  candidate relations from the Knowledge Graph (KG). The context-aware path evaluation performed during the simulation step introduces an additional cost of  $\mathcal{O}(N \cdot k \cdot L^3 \cdot d)$ , where  $L$  is the maximum path length and  $d$  is the embedding dimension. In the Path-based Dynamic Refinement stage, the evaluator is fine-tuned for  $N_{\text{FT}}$  steps, with a total complexity of  $\mathcal{O}(N_{\text{FT}} \cdot L^2 \cdot d)$ . Consequently, the overall time complexity of DAMR is:  $\mathcal{O}(N \cdot (T_{\text{LLM}}(k) + k \cdot L^3 \cdot d) + N_{\text{FT}} \cdot L^2 \cdot d)$ .

## C. Datasets

### Dataset Description

Table 6: Statistics of KGQA datasets.

Datasets	#Train	#Valid	#Test
WebQSP	2,848	250	1,639
CWQ	27,639	3,519	3,531

We conduct extensive experiments on two widely used multi-hop Knowledge Graph Question Answering (KGQA) benchmarks: WebQSP (Talmor and Berant 2018) and CWQ (Yih et al. 2016). The statistics of these two benchmarks can be found in Table 6, and their details are shown as follows:

- The WebQuestionsSP (WebQSP) dataset is a widely adopted benchmark for evaluating single-hop and simple multi-hop KGQA (Yih et al. 2016). It consists of 4,837 natural language questions annotated with corresponding SPARQL queries over the Freebase knowledge graph. The dataset is partitioned into 2,848 training, 250 validation, and 1,639 test instances.

- The ComplexWebQuestions (CWQ) dataset is a challenging benchmark designed for multi-hop KGQA (Talmor and Berant 2018). It comprises 34,689 questions derived from WebQuestionsSP, reformulated to include more complex and compositional queries. Each question typically requires multi-step reasoning over the Freebase knowledge graph, often involving conjunctions, comparatives, or nested logical structures. The dataset is divided into 27,639 training, 3,519 validation, and 3,531 test examples.

## Data Processing

Following prior work (Shen et al. 2025; Long et al. 2025), we preprocess the datasets by constructing localized subgraphs centered around each question entity to reduce the size of the search space. Specifically, for each question in WebQSP (Yih et al. 2016) and CWQ (Talmor and Berant 2018), we extract a subgraph from the Freebase knowledge graph by including all triples within a predefined number of hops from the topic entity. This approach preserves the essential context required for multi-hop reasoning while significantly improving computational efficiency.

## D. Baselines

In this part, we introduce the details of the compared baselines as follows:

- **Semantic Parsing Methods.** We compare our DAMR with six semantic parsing methods:
  - **KV-Mem:** KV-Mem (Miller et al. 2016) introduce a neural architecture that stores facts as key-value pairs and enables question answering by attending over memory slots, directly retrieving relevant information to infer answers.
  - **EmbedKGQA:** EmbedKGQA (Saxena, Tripathi, and Talukdar 2020) enhances multi-hop question answering over knowledge graphs by leveraging pretrained knowledge base embeddings, enabling the model to reason over entity and relation representations without explicit path enumeration during answer prediction.
  - **QGG:** QGG (Lan and Jiang 2020) generates query graphs to answer multi-hop complex questions over knowledge bases, formulating question answering as query graph prediction and enabling structured reasoning through graph matching and path ranking mechanisms.
  - **NSM:** NSM (He et al. 2021) enhances multi-hop KBQA by leveraging intermediate supervision signals, decomposing questions into reasoning steps, and training a neural state machine to sequentially predict relations and entities for accurate path-based reasoning.
  - **TransferNet:** TransferNet (Shi et al. 2021) proposes a transparent framework for multi-hop QA over relational graphs by transferring question semantics to relation paths through interpretable path ranking and structured reasoning, enabling effective and explainable answer prediction.
  - **KGT5:** KGT5 (Saxena, Kochsiek, and Gemulla 2022) formulates knowledge graph completion and question answering as unified sequence-to-sequence tasks, leveraging pre-trained language models to jointly encode input queries and generate answer entities or triples in a flexible and end-to-end manner.
- **Retrieval-Based Methods.** We compare our DAMR with four retrieval-based methods:
  - **GraftNet:** GraftNet (Sun et al. 2018) proposes an early fusion framework that jointly encodes knowledge base facts and supporting text by constructing a heterogeneous graph, enabling effective reasoning through graph convolutional networks for open-domain question answering.
  - **PullNet:** PullNet (Sun, Bedrax-Weiss, and Cohen 2019) introduces an iterative retrieval mechanism that expands a query-specific subgraph by pulling relevant facts from both knowledge bases and text, enabling joint reasoning over heterogeneous evidence for open-domain question answering.
  - **SR+NSM:** SR+NSM (Zhang et al. 2022) enhances multi-hop KBQA by first retrieving a question-relevant subgraph and then performing symbolic reasoning over it using Neural Symbolic Machines, improving efficiency and accuracy through constrained and focused logical inference.
  - **SR+NSM+E2E:** SR+NSM+E2E (Zhang et al. 2022) extends SR+NSM by enabling end-to-end training that jointly optimizes subgraph retrieval and reasoning. This integration enhances model coherence and allows better alignment between retrieved subgraphs and final answer prediction.
- **General Large Language Models (LLMs).** We compare our DAMR with six general LLMs:
  - **Flan-T5-xl:** Flan-T5-xl (Chung et al. 2024) is an instruction-finetuned variant of the T5 model, trained on a diverse collection of tasks with natural language instructions. By leveraging large-scale instruction tuning, it improves zero-shot and few-shot performance across diverse NLP benchmarks.
  - **Alpaca-7B:** Alpaca-7B (Taori et al. 2023) is an instruction-following language model fine-tuned from LLaMA-7B using self-instruct techniques. It demonstrates strong zero-shot and few-shot performance by aligning with human instructions across various NLP tasks.

- **Llama3-8B**: Llama3-8B (Dubey et al. 2024) is part of the LLaMA 3 family of models, designed for improved instruction following, reasoning, and code generation. Pretrained on a high-quality corpus and fine-tuned with supervised signals, it achieves strong performance across diverse benchmarks.
- **Qwen2.5-7B**: Qwen2.5-7B (Team 2024) is a 7B-parameter instruction-tuned language model developed by Alibaba, optimized for tasks such as reasoning, code generation, and dialogue. It supports multi-turn conversation and demonstrates competitive performance on standard benchmarks.
- **ChatGPT**: ChatGPT (Schulman et al. 2022) is a conversational AI developed by OpenAI, based on the GPT architecture. It is designed to understand natural language, engage in dialogue, answer questions, and assist with a wide range of tasks across domains.
- **ChatGPT+CoT**: ChatGPT with Chain-of-Thought (CoT) (Wei et al. 2022) prompting enhances the model’s reasoning capabilities by encouraging it to generate intermediate reasoning steps before arriving at a final answer, improving performance on complex, multi-step problems.
- **LLMs with KG**. We compare our DAMR with fourteen LLMs with KG methods:
  - **UniKGQA**: UniKGQA (Jiang et al. 2022) is a unified framework that integrates retrieval and reasoning for multi-hop question answering over knowledge graphs, combining subgraph retrieval, query decomposition, and neural reasoning in an end-to-end manner.
  - **DECAF**: DECAF (Yu et al. 2022) is a joint framework for question answering over knowledge bases that simultaneously decodes logical forms and answers. By leveraging dual supervision, it enhances both symbolic reasoning accuracy and direct answer prediction in a unified architecture.
  - **KD-CoT**: KD-CoT (Wang et al. 2023a) is a framework that enhances the faithfulness of large language models by guiding Chain-of-Thought reasoning with external knowledge, improving accuracy in knowledge-intensive question answering tasks.
  - **Nutrea**: Nutrea (Choi et al. 2023) proposes a neural tree search framework for context-guided multi-hop KGQA. It incrementally constructs reasoning trees by integrating question semantics and graph context, enabling efficient exploration of multi-hop paths for accurate answer prediction.
  - **ToG**: ToG (Sun et al. 2023) is a framework that enables large language models to perform deep and responsible reasoning over knowledge graphs by combining structured graph information with iterative thinking and verification mechanisms for reliable multi-hop QA.
  - **RoG**: RoG (Luo et al. 2023) is a framework that enhances the faithfulness and interpretability of large language model reasoning by grounding multi-hop question answering on knowledge graphs, integrating symbolic path tracking with natural language generation.
  - **KAPING**: KAPING (Baek, Aji, and Saffari 2023) introduces knowledge-augmented prompting by integrating structured triples into Chain-of-Thought (CoT) reasoning. It guides large language models to generate intermediate reasoning steps, enabling zero-shot multi-hop KGQA without task-specific fine-tuning.
  - **ReasoningLM**: ReasoningLM (Jiang et al. 2023) enhances pre-trained language models for KGQA by injecting subgraph structures into the input representation. It enables structural reasoning over retrieved subgraphs through a reasoning-aware encoder, improving performance on complex multi-hop queries.
  - **FiDeLis**: FiDeLis (Sui et al. 2024) proposes a faithfulness-aware KGQA framework that enhances reasoning consistency in LLMs by aligning generated logical forms with answer predictions. It introduces fidelity constraints to reduce hallucinations and improve answer correctness.
  - **GNN-RAG**: GNN-RAG (Mavromatis and Karypis 2024) integrates graph neural networks with retrieval-augmented generation by encoding knowledge subgraphs into LLMs’ context. It enables structural reasoning over retrieved subgraphs, improving answer accuracy in KGQA through explicit graph-aware representations.
  - **DoG**: DoG (Ma et al. 2025a) is a flexible and reliable reasoning framework that enables large language models to generate and evaluate multiple reasoning paths over knowledge graphs through a debate-style process, enhancing robustness and answer faithfulness.
  - **DuarL**: DuarL (Liu et al. 2025) is a collaborative framework that integrates GNNs and LLMs for KGQA, where GNNs capture structural semantics and LLMs perform adaptive reasoning, enabling accurate and interpretable multi-hop QA.
  - **DP**: DP (Ma et al. 2025b) is a trustworthy reasoning framework that guides large language models using prior knowledge from knowledge graphs. It iteratively verifies and refines reasoning paths to enhance faithfulness, robustness, and answer accuracy in KGQA.
  - **RwT**: RwT (Shen et al. 2025) is a faithful KGQA framework that models multi-hop reasoning as tree-structured exploration over knowledge graphs, enabling large language models to generate interpretable reasoning paths and improve answer consistency and accuracy.

<b>Question</b>	Where does greek language come from?
<b>Answer</b>	<b>Attic Group</b>
<b>Llama-2-13B</b>	The Greek language originates from the Proto-Indo-European language family, which spread to Greece around 2000 BCE. It evolved into the ancient Greek language, influenced by Mycenaean and Aeolic dialects, and eventually developed into Modern Greek.
<b>Qwen-3-14B</b>	The Greek language originates from the Proto-Hellenic branch of the Indo-European language family. It evolved from the Mycenaean Greek of the second millennium BCE and became the foundation for Classical Greek, spoken in ancient Greece.
<b>GPT 4.1</b>	The Greek language originates from the Indo-European language family. It developed in the southern Balkans around the 3rd millennium BCE and is the <u>oldest recorded living language</u> , with written records dating back to the 15th century BCE (Linear B script).
<b>GPT 4.1-mini</b>	The Greek language originates from the Hellenic branch of the Indo-European language family. It developed in the region of Greece and the Aegean around the 3rd millennium BCE, evolving through ancient, medieval, and modern stages.
<b>DAMR</b>	Path 1: Entity (id: 120026) → base.rosetta.languoid.parent → <b>Attic Group</b> .

Table 7: Case study of DAMR. We highlight the correct answers in **Bold** and the wrong answers in underline.

Table 8: Hyperparameter sensitivity analysis of the number of selected relations  $k$  on the WebQSP and CWQ datasets.

Method	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
$k = 2$	93.0	80.9	76.6	73.8
$k = 3$	94.0	81.7	78.0	75.1
$k = 4$	94.0	81.8	77.8	75.0
$k = 5$	93.9	81.7	78.0	75.2

## E. More experimental results

### More Sensitivity Analysis

To more thoroughly illustrate the impact of hyperparameter variations on model performance, we report detailed numerical results showing how performance fluctuates under different hyperparameter settings. As presented in Table 8 and Table 9, these results provide a comprehensive understanding of the model’s sensitivity and stability across a range of configurations.

### More case study

Table 7 presents a case study comparing the answer accuracy of DAMR with general LLMs: Llama-2-13B, Qwen-3-14B, GPT 4.1-mini, and GPT 4.1. When asked about the origin of the Greek language, all baseline models generate fluent and seemingly plausible responses grounded in general linguistic knowledge, such as “Proto-Indo-European” or “Proto-Hellenic”, but fail to identify the correct answer: Attic Group. In contrast, DAMR accurately predicts the correct entity by explicitly traversing the relation path *base.rosetta.languoid.parent* within the knowledge graph. This example illustrates a key advantage of DAMR: rather than relying solely on learned linguistic patterns, it performs structured reasoning over the knowledge graph, enabling precise and faithful answers to ontology-specific queries that often elude general-purpose LLMs.

## F. Prompt Template

We provide the prompt templates used by the LLM-based planner to select the top- $k$  most relevant relations from the candidate set at each step of path expansion in Fig. 3, as part of the LLM Guided Path Expansion module.

Table 9: Hyperparameter sensitivity analysis of the reasoning path length  $L$  on the WebQSP and CWQ datasets.

Method	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
$L = 2$	93.6	81.2	77.4	74.5
$L = 3$	94.0	81.7	77.6	74.7
$L = 4$	93.7	81.4	78.0	75.1
$L = 5$	93.8	81.6	77.9	74.9

#### Prompt Template for LLM-Guided Path Expansion

##### Role

You are an expert assistant for Knowledge Graph Question Answering (KGQA). Your core capability is to deeply understand natural language questions and the semantics of knowledge graph relations to find the most relevant reasoning paths.

##### Task

Your task is to act as a “**Relation Retriever**.” Given a natural language question and a list of candidate relations, you must analyze the semantics of the question and each relation to select up to  $k$  relations that are most likely to lead to the correct answer.

##### Rules and Constraints

- **Fidelity to Candidates:** Your selection of relations **MUST** come strictly from the provided `Candidate Relations` list. Do not invent or modify relations.
- **Quantity Limit:** Return no more than  $k$  relations. If multiple relations are highly relevant, order them from most to least relevant. If there are fewer than  $k$  relevant relations, return only those.
- **Output Format:** Your response **MUST** be a list of strings, containing the names of the relations you have selected.

##### Example

###### • Input:

- Question: "who was the president after jfk died"
- Candidate Relations: {"government.president", "government.president.successor", "location.location.containedby", "people.person.place\_of\_birth"}
- $K$ : 2

###### • Output:

```
["government.president", "government.president.successor"]
```

##### Your Task

- Question: {question}
- Candidate Relations: {relations\_list}
- $K$ : {k}

##### Output:

```
[]
```

Figure 3: Prompt template used in the LLM-based planner to select top- $k$  relations during reasoning.