Stepwise Contrastive Reasoning for Retrieval-Augmented Generation over Knowledge Graphs

Chenxiao Lin[1], Ye Luo[2*], KunHong Liu[1,4], Qingqiang Wu[1,2,3,4,5*]
[1]School of Film, Xiamen University, Xiamen, China
[2]School of Informatics, Xiamen University, Xiamen, China
[3]Institute of Artificial Intelligence, Xiamen University, Xiamen, China
[4]Xiamen Key Laboratory of Intelligent Storage and Computing, Xiamen University, Xiamen, China
[5]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan, Ministry of Culture and Tourism, Xiamen University, Xiamen, China
lcx064@gmail.com, {wuqq, lkhqz, luoye}@xmu.edu.cn

Abstract

Retrieval-augmented generation (RAG) enhances the reasoning capabilities of large language models (LLMs) by incorporating external knowledge. Among available sources, knowledge graphs (KGs) offer a structured and reliable foundation for factual information, making them increasingly popular in efforts to improve reasoning faithfulness in RAG. Most existing KG-based RAG methods rely on LLMs to extract knowledge from KGs. However, these approaches often require costly fine-tuning and struggle to navigate deep graph structures, limiting their effectiveness in multi-hop reasoning tasks. To address these challenges, we propose Stepwise Contrastive Reasoning (SCR), a lightweight framework that integrates graph structure and textual context for efficient and interpretable RAG over KGs. SCR combines relational message passing layers to encode KG entities with a Transformer encoder for processing question text. It decomposes reasoning into a series of alignment steps. At each step, SCR compares the current topic entity and its neighbors with the question representation, selecting the most relevant entity as the next topic entity. The question is then updated with this entity's textual description. This process continues until the selected entity no longer changes, indicating that the answer entity has been reached. Through stepwise alignment, SCR enables compact models to perform faithful and interpretable reasoning over large-scale KGs. Extensive experiments on several widely used KGQA benchmarks demonstrate that SCR not only achieves state-of-the-art performance but also effectively boosts the capabilities of smaller language models to match those of LLMs.

Code — https://github.com/ado-cs/SCR

Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across natural language processing (NLP) tasks (Brown et al. 2020), bringing machines closer to human-level understanding. However, their reasoning often suffers from knowledge gaps and hallucinations (Huang et al. 2024), which limit their reliability in real-world applications.

*Corresponding authors.

Retrieval-augmented generation (RAG) (Lewis et al. 2020) addresses this limitation by enriching LLM inputs with external, up-to-date knowledge. Knowledge graphs (KGs), which store structured and verifiable factual information, are particularly well-suited for this purpose (Pan et al. 2024). Their reliability and ease of updating have led to growing interest in using KGs within RAG frameworks (Luo et al. 2024a; Mavromatis and Karypis 2025).

In the knowledge graph question answering (KGQA) task, the goal is to answer natural language questions by grounding the reasoning process in facts stored in a KG. The effectiveness of RAG in this setting largely depends on retrieving relevant KG facts. Since KGs encode information as complex graph structures, identifying the useful information for reasoning is non-trivial. Ineffective retrieval may result in irrelevant or misleading contexts that confuse the LLM (He et al. 2024). Existing LLM-based retrieval methods often represent KG facts as fixed-format text, such as "`Austria` → `country.capital` → `Vienna`," allowing LLMs to extract knowledge in natural language form. However, these methods often fall short in multi-hop reasoning tasks, as they rely solely on textual representations and fail to capture the underlying graph structure (Baek, Aji, and Saffari 2023). To address this, some methods treat LLMs as agents that iteratively interact with the KG to perform deep traversal (Sun et al. 2024). Despite improved performance, these approaches are computationally intensive and incur high latency.

In real-world applications, KGs are frequently updated, and fine-tuning LLMs to accommodate these changes is computationally expensive (Luo et al. 2025). As a more efficient alternative, recent studies have explored the use of graph neural networks (GNNs) (Velickovic et al. 2018) to process KG structures more effectively (He et al. 2024; Luo et al. 2025). GNNs can model multi-hop dependencies and improve retrieval quality for KGQA (Mavromatis and Karypis 2025). However, they lack the ability to understand natural language at the level that LLMs do, limiting their effectiveness when semantic alignment between the question and entities is critical. Some methods attempt to overcome this by jointly training GNNs and LLMs and combining their predictions (Mavromatis and Karypis 2025), but this intro-

duces further computational overhead. Moreover, simply expanding the number of candidates not only increases token usage in the LLM input but also raises the proportion of irrelevant information, leading to the "lost in the middle" phenomenon (Liu et al. 2024b).

In this work, we propose Stepwise Contrastive Reasoning (SCR), a lightweight and interpretable framework that integrates graph structure and textual context for effective and efficient RAG over KGs. SCR trains a relational GNN and a Transformer encoder (Vaswani et al. 2017) via contrastive learning, achieving strong performance in both text understanding and complex graph processing. Starting from a topic entity identified in the question, SCR incrementally navigates the KG to reach the answer entity, producing faithful and interpretable reasoning paths.

To initiate the reasoning process, SCR employs a pretrained language model (PLM) to extract features from the textual descriptions of entities and relations. These features are then passed to the GNN to produce aggregated semantic representations for each entity. After that, the reasoning proceeds in steps: (1) the encoder processes the question to generate its embedding; (2) this embedding is compared with the representations of the current topic entity and its neighbors to identify the most semantically aligned entity; and (3) the selected entity becomes the new topic entity, and its textual description is appended to the question context. This process repeats until the topic entity remains unchanged, indicating that the answer has been found. Reasoning in a stepwise manner not only achieves better performance but also provides more interpretable results than single-step reasoning. Additionally, it can be paired with LLM generation strategies, such as beam search (Federico et al. 1995), to explore multiple reasoning paths in parallel, further enhancing retrieval robustness. Our main contributions are as follows:

- We propose Stepwise Contrastive Reasoning (SCR), which decomposes KG retrieval into multiple steps of aligning textual and graph embeddings, effectively handling reasoning tasks that are semantically sensitive or require navigating complex graph structures.
- We introduce a novel framework with a lightweight and flexible design that effectively learns representations from both textual and structural information in KGs, using only 112M trainable parameters.
- We are the first to integrate LLM generation strategies into GNN-based RAG, bridging a gap in the literature while significantly enhancing KG retrieval performance.
- We validate SCR through extensive experiments on multiple KGQA benchmarks, demonstrating both its effectiveness and efficiency.

Related Work

LLM Reasoning. Recent research has made significant progress in enhancing the reasoning capabilities of LLMs, particularly through prompt engineering. Chain-of-Thought (CoT) prompting (Wei et al. 2022) encourages models to generate intermediate reasoning steps when answering a question, while Tree-of-Thought (ToT) (Yao et al. 2023) extends this by exploring multiple reasoning paths in a tree-like structure to identify optimal solutions. Graph-of-Thought (GoT) (Besta et al. 2024) further generalizes this idea by organizing generated information into graph structures, where nodes represent reasoning units and edges denote dependencies. Beyond prompting, other approaches focus on fine-tuning LLMs on reasoning tasks to improve their deductive performance.

KG-Integrated LLM Reasoning. An emerging direction explores combining LLMs with KGs to enhance reasoning grounded in structured information (Pan et al. 2024). KD-CoT (Wang et al. 2023a) retrieves factual knowledge from external KGs to guide the CoT reasoning process. ToG (Sun et al. 2024) treats LLMs as agents that actively interact with KGs to navigate reasoning paths. RoG (Luo et al. 2024a) introduces a planning–retrieval–reasoning framework that retrieves reasoning paths from KGs to support more faithful LLM reasoning. GCR (Luo et al. 2024b) strengthens KG-grounded reasoning by integrating KG structure directly into the LLM decoding process via KG-Trie, a trie-based index representing valid reasoning paths. To overcome the limitations of LLM-based approaches in capturing graph structure, recent methods (He et al. 2024; Mavromatis and Karypis 2025; Luo et al. 2025) incorporate GNNs to aggregate the information within KGs, thereby improving performance on multi-hop reasoning tasks.

Preliminary

KGQA. A KG $\mathcal{G}$ stores factual information as a set of triples in the form (e, r, e'), where e and e' are entities, and r is the relation connecting them. Formally, $\mathcal{G} = (e, r, e') \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, with $\mathcal{E}$ and $\mathcal{R}$ representing the sets of entities and relations, respectively. Given a natural language question q, the goal of KGQA is to identify the correct set of entities $a \subseteq \mathcal{E}$ that answer q, using the information contained in $\mathcal{G}$.

Reasoning over KGs. Given the large size of KGs, it is computationally impractical to use the entire graph for each question. Instead, a smaller, relevant subgraph $\mathcal{G}_q \subseteq \mathcal{G}$ is first retrieved using techniques such as entity linking and neighbor extraction. This subgraph, along with the question q, serves as input to a reasoning model that predicts the answer entities. During this process, the model may uncover reasoning paths such as $p = e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \dots \xrightarrow{r_l} e_l$, where e_0 is the topic entity identified from q, e_l is the predicted answer, and $\forall(e_{i-1}, r_i, e_i) \in \mathcal{G}_q$.

KG-based RAG. To enable LLMs to reason over KG data, the retrieved reasoning paths are translated into natural language. The resulting input to the LLM includes both the reformulated reasoning paths and the original question, typically formatted as a prompt. For example, the input might read: "`Reasoning paths: Austria` $\rightarrow$ `country.capital` $\rightarrow$ `Vienna \n Question: What is the Capital of Austria? \n Based on the reasoning paths, please answer the given question.`". This setup allows the LLM to leverage structured KG information in a natural language context to generate accurate answers.

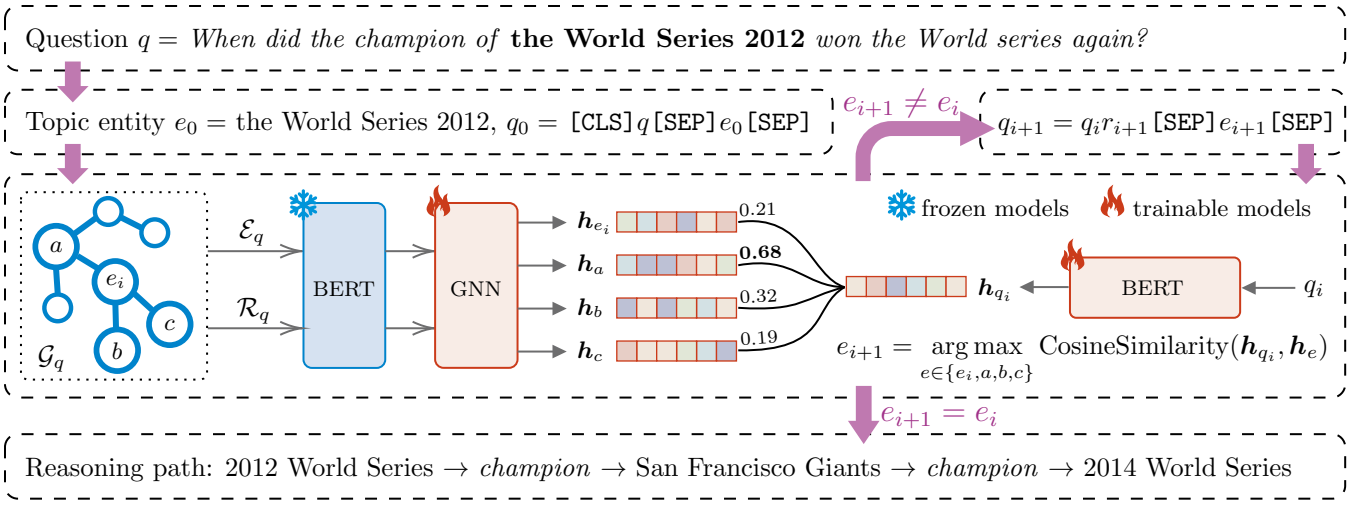

Figure 1: Illustration of the proposed SCR framework. The process operates as an iterative loop that updates both the question and the topic entity based on the similarity between the embedding of current question and the embeddings of candidate entities (i.e., the current topic entity and its neighbors). The iteration continues until the topic entity stabilizes, at which point the model outputs the sequence of traversed entities and their connecting relations as the reasoning path for RAG.

Methodology

Our proposed SCR framework adopts a stepwise embedding alignment pipeline for KG-based RAG, as illustrated in Figure 1. While prior works have applied GNNs for reasoning over KGs (He et al. 2024; Luo et al. 2024a, 2025), our approach is novel in addressing three key challenges: First, we aggregate the textual information of entities and relations along the graph structure and align the resulting features with the question representation, effectively bridging semantic content and structural context through a lightweight network. Second, instead of inferring an answer from the KG in a single step, SCR incrementally traverses the graph based on learned probabilities, resulting in well-grounded and interpretable reasoning paths. Third, SCR supports the integration of LLM generation strategies into stepwise reasoning, significantly improving the robustness of KG retrieval.

Stepwise Contrastive Rasoning

Given a question q, our SCR framework begins by selecting a topic entity e_0 from q to serve as the root node. Since questions may contain multiple topic entities, we append the textual description of the selected entity to the question to clarify the context. Using special tokens commonly applied in BERT (Devlin et al. 2019) models, we form the initial question as:

$$q_0 = \texttt{[CLS]}q\texttt{[SEP]}e_0\texttt{[SEP]}. \tag{1}$$

SCR then proceeds with iterative retrieval over the KG to identify the answer entity. The following paragraphs describe the key processes and components involved in each reasoning step.

Encoding Entities and Relations. We use a pre-trained BERT model to encode all entities and relations within the KG. For a given textual input x (representing either an entity or a relation), we obtain its embedding as:

$$\boldsymbol{v}_x = \text{BERT}(x) \in \mathbb{R}^d, \tag{2}$$

where d is the output embedding dimension. Since the encoder remains frozen during training, we precompute and store these embeddings for efficient reasoning. In practice, such a process is commonly used in KG index construction to facilitate topic entity recognition.

Relational Message Passing. To update entity representations using neighborhood information, we design a relational message passing layer based on the attention mechanism in GAT (Velickovic et al. 2018). Specifically, let $\boldsymbol{v}_e$ and $\boldsymbol{v}_{e'}$ denote the embeddings of entities e and e', respectively, and let $\boldsymbol{v}_r$ represent the embedding of the relation r connecting them. We first apply linear transformations to entity and relation embeddings using learnable weight matrices $\mathbf{W}_\mathcal{E} \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_\mathcal{R} \in \mathbb{R}^{d\times d}$, respectively. Given attention vectors $\mathbf{a}_\mathcal{E} \in \mathbb{R}^{2d}$ for entities and $\mathbf{a}_\mathcal{R} \in \mathbb{R}^d$ for relations, the attention coefficient between e and e' is computed as:

$$\beta_{ee'} = \mathbf{a}_\mathcal{E}^T[\mathbf{W}_\mathcal{E}\boldsymbol{v}_e \parallel \mathbf{W}_\mathcal{E}\boldsymbol{v}_{e'}] + \mathbf{a}_\mathcal{R}^T(\mathbf{W}_\mathcal{R}\boldsymbol{v}_r), \tag{3}$$

where $\cdot^T$ denotes transposition and $\parallel$ is the concatenation operation. The coefficient is then passed through a LeakyReLU activation and normalized across all neighbors of e via the softmax function:

$$\alpha_{ee'} = \frac{\exp(\text{LeakyReLU}(\beta_{ee'}))}{\sum_{n\in\mathcal{N}_e}\exp(\text{LeakyReLU}(\beta_{en}))}, \tag{4}$$

where $\mathcal{N}_e$ is the set of neighbors of e.

The final updated embedding for e incorporates a residual connection and is expressed as:

$$\hat{\boldsymbol{v}}_e = \sigma\left(\mathbf{W}_\mathcal{E}\boldsymbol{v}_e + \sum_{n\in\mathcal{N}_e}\alpha_{en}\mathbf{W}_\mathcal{E}\boldsymbol{v}_n\right) \in \mathbb{R}^d, \tag{5}$$

where σ is a non-linear activation function.

This message passing layer can be stacked to construct a multi-layer GNN, where each layer updates entity features using those from the previous layer. For a query-specific subgraph $\mathcal{G}_q$ with entity set $\mathcal{E}_q$ and $\mathcal{R}_q$, the GNN outputs the aggregated entity features:

$$\mathbf{h}_q = \text{GNN}(\mathcal{G}_q, \mathbf{v}_{\mathcal{E}_q}, \mathbf{v}_{\mathcal{R}_q}), \tag{6}$$

where $\mathbf{v}_{\mathcal{E}_q} = \{\boldsymbol{v}_x | x \in \mathcal{E}_q\}$ and $\mathbf{v}_{\mathcal{R}_q} = \{\boldsymbol{v}_x | x \in \mathcal{R}_q\}$. Each row of $\mathbf{h}_q \in \mathbb{R}^{|\mathcal{E}_q| \times d}$ represents the feature vector of a corresponding entity in $\mathcal{E}_q$. We denote the feature vector of entity e as $\boldsymbol{h}_e \in \mathbb{R}^d$. In theory, a GNN with L message passing layers can aggregate information from neighbors up to L hops away in a single forward pass.

Encoding the Question. The BERT model used to encode the question shares initial weights with the encoder for entities and relations, but unlike the latter, it remains trainable. We denote the representation of the question q as $\boldsymbol{h}_q \in \mathbb{R}^d$.

Selecting the Next Topic Entity. At each reasoning step i, let q_i be the current question and e_i the current topic entity. We retrieve the GNN features of e_i and its neighbors and compute the cosine similarity between each and the encoded question $\boldsymbol{h}_{q_i}$. The next topic entity is chosen as:

$$e_{i+1} = \underset{e \in \{e_i\} \bigcup \mathcal{N}_{e_i}}{\arg\max} \text{CosineSimilarity}(\boldsymbol{h}_{q_i}, \boldsymbol{h}_e). \tag{7}$$

If $e_{i+1} \neq e_i$, we continue reasoning with:

$$q_{i+1} = q_i r_{i+1} \texttt{[SEP]} e_{i+1} \texttt{[SEP]}, \tag{8}$$

where r_{i+1} is the relation connecting e_i and e_{i+1}. If $e_{i+1} = e_i$, reasoning halts, and e_i is treated as the final answer.

This process yields an interpretable reasoning path from e_0 to e_i:

$$p = e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \ldots \xrightarrow{r_i} e_i, \tag{9}$$

which can be fed into a general LLM as reliable external knowledge to enhance answer generation. To prevent revisiting nodes, we record all previously traversed entities during reasoning.

Optimization Objective

The optimization objective of SCR is to maximize the likelihood of selecting the correct next topic entity at each reasoning step. Since the selection is based on the cosine similarity between the question representation and the features of candidate entities, we adopt a contrastive learning approach to increase the margin between positive and negative entities. Specifically, given the question q_i and the topic entity e_i at reasoning step i, we minimize the InfoNCE (He et al. 2020) loss:

$$\mathcal{L} = -\log \frac{\phi(\boldsymbol{h}_{q_i}, \boldsymbol{h}_{e_{i+1}})}{\sum_{n \in \mathcal{M}_{e_i}} \phi(\boldsymbol{h}_{q_i}, \boldsymbol{h}_n)}, \tag{10}$$

$$\phi(\boldsymbol{h}_q, \boldsymbol{h}_e) = \exp(\text{CosineSimilarity}(\boldsymbol{h}_q, \boldsymbol{h}_e)/\tau). \tag{11}$$

Here, τ is a temperature parameter that scales the similarity scores. The set of samples $\mathcal{M}_{e_i} = \{e_i\} \cup \mathcal{N}_{e_i}$ consists of the current topic entity and its neighbors.

Reasoning with Beam Search

At each step of SCR, multiple reasoning paths can be explored in parallel to leverage GPU-based computation. In this case, the next topic entity selection defined in Equation 7 is extended to a top-K formulation:

$$e_{i+1} = \underset{e \in \{e_i\} \bigcup \mathcal{N}_{e_i}}{\arg\text{top-}K} \text{CosineSimilarity}(\boldsymbol{h}_{q_i}, \boldsymbol{h}_e). \tag{12}$$

Given a question q and its relevant subgraph $\mathcal{G}_q$, the probability of an entity a being the answer is computed as:

$$P(a|q, \mathcal{G}_q) = \sum_p P_\theta(a|q, p) P_\theta(p|q, \mathcal{G}_q), \tag{13}$$

where $P_\theta(p|q, \mathcal{G}_q)$ denotes the probability of discovering a reasoning path p within $\mathcal{G}_q$, given the question q, using a model parameterized by θ.

To enhance the interpretability of answers, each path is explicitly represented as a sequence of visited entities. The joint probability of reaching an answer a via path p is then defined as:

$$P(a, p|q) = P_\theta(a|q, p) \prod_{i=1}^{|p|} P_\theta(e_i|q_{i-1}, e_{i-1}). \tag{14}$$

As a result, the top-K hypothesis answers and their corresponding reasoning paths under beam search (Federico et al. 1995) are obtained by:

$$\mathcal{B}_K = \{a^k, p^k\}_{k=1}^K = \underset{p}{\arg\text{top-}K}\, P(a, p|q). \tag{15}$$

In the final stage of RAG, the set $\mathcal{B}_K$ is formatted and injected into the input context of a general LLM to provide reliable KG-derived knowledge for answer generation.

Experiments

Experimental Setup

Datasets. To evaluate the reasoning capabilities of SCR, we conduct experiments on two widely used KGQA benchmarks: WebQuestionsSP (WebQSP) (Yih et al. 2016) and ComplexWebQuestions (CWQ) (Talmor and Berant 2018). Both datasets adopt Freebase (Bollacker et al. 2008) as the underlying KG and require multi-hop reasoning.

Baselines. We compare SCR with baselines across three categories. (1) LLM-based reasoning methods include the Qwen3 (Yang et al. 2025), DeepSeek (Liu et al. 2024a), and GPT (OpenAI 2022) series, as well as three ChatGPT-based prompting strategies: Few-shot (Brown et al. 2020), Chain-of-Thought (CoT) (Wei et al. 2022), and Self-Consistency (Wang et al. 2023b). (2) Graph-based reasoning methods cover GraftNet (Sun et al. 2018), NSM (He et al. 2021), SR+NSM (Zhang et al. 2022), ReaRev (Mavromatis and Karypis 2022), and UniKGQA (Jiang et al. 2023). (3) KG-integrated LLM reasoning methods include KD-CoT (Wang et al. 2023a), EWEK-QA (Dehghan et al. 2024), ToG (Sun et al. 2024), EffiQA (Dong et al. 2025), RoG (Luo et al. 2024a), and GCR (Luo et al. 2024b), which leverage

LLM Input Prompt

```
============== Prompt Input ==============
# Reasoning Paths:
<Path 1><Hypothesis Answer 1>
...
<Path K><Hypothesis Answer K>

# Question:
<Question>

Based on the reasoning paths, answer the given question as simply as possible. Please return only the answers, with each answer on a new line.

============== LLM Output ==============
<Answer 1>
<Answer 2>
...
```

Figure 2: LLM prompt template for KG-based RAG.

LLMs to extract KG facts, and GNN-based methods such as G-Retriever (He et al. 2024), GNN-RAG (Mavromatis and Karypis 2025), and SubgraphRAG (Li, Miao, and Li 2025). The published GCR results are based on training over the combined dataset of WebQSP and CWQ. For a fair comparison, we reimplement GCR using its publicly available code under the same configuration, except that we train it on each dataset separately. Performance for the remaining baselines is taken from their original publications.

Evaluation Metrics. We assess both effectiveness and efficiency. For effectiveness, we report Hit and F1 scores. Hit checks whether any correct answer appears in the generated response. F1 balances precision and recall by evaluating the overlap between predicted and true answers. All scores are scaled from 0 to 100, with higher values indicating stronger factual accuracy. To measure efficiency, we record wall-clock time (in seconds) on an NVIDIA RTX 4090 GPU (24 GB). KG query time is excluded to isolate model computational cost. For text embedding employed by SCR and most baselines, we only include the time needed for question embedding, since entity and relation embeddings can be precomputed.

Implementation Details. Our framework uses *bert-base-uncased*[1] for BERT initialization. The GNN consists of three relational message-passing layers. In total, SCR contains approximately 112 million trainable parameters. During training, we extract the shortest paths between topic and answer entities as supervision signals. To support LLM generation, we incorporate the top-10 reasoning paths and corresponding hypothesis answers into the query prompt (as shown in Figure 2). Additionally, we set the probability threshold (as defined in Equation 14) to 0.02 to filter out results with low confidence. For consistency across runs, we fix both the temperature and random seed to zero for all LLM reasoners.

[1]https://huggingface.co/bert-base-uncased

Methods	WebQSP		CWQ	
	Hit	F1	Hit	F1
LLM Reasoning				
Qwen3-0.6B	24.9	10.2	12.4	6.1
Qwen3-8B	47.6	29.8	25.9	21.1
Qwen3-232B	62.0	39.2	37.8	27.6
DeepSeek-V3	64.4	42.5	42.6	32.2
GPT-4o-mini	63.8	40.5	63.8	40.5
ChatGPT	59.3	43.5	34.7	30.2
ChatGPT+Few-shot	68.5	38.1	38.5	28.0
ChatGPT+CoT	73.5	38.5	47.5	31.0
ChatGPT+Self-Consistency	83.5	63.4	56.0	48.1
Graph Reasoning				
GraftNet	66.7	62.4	36.8	32.7
NSM	68.7	62.8	47.6	42.4
SR+NSM	68.9	64.1	50.2	47.1
ReaRev	76.4	70.9	52.9	47.8
UniKGQA	77.2	72.2	51.2	49.1
KG-Integrated LLM Reasoning				
KD-CoT	68.6	52.5	55.7	-
EWEK-QA	71.3	-	52.5	-
ToG (ChatGPT)	76.2	-	57.6	-
ToG (GPT-4)	82.6	-	68.5	-
EffiQA	82.9	-	69.5	-
RoG (Llama-2-7B)	85.7	70.8	62.6	56.2
GCR (GPT-4o-mini)	76.4	50.1	54.5	44.1
G-Retriever	73.5	53.4	-	-
GNN-RAG	85.7	71.3	66.8	59.4
SubgraphRAG (ChatGPT)	83.1	69.2	56.3	49.1
SubgraphRAG (GPT-4o-mini)	90.1	**77.5**	62.0	54.1
SCR (Qwen3-0.6B)	86.5	51.0	63.2	37.2
SCR (Qwen3-8B)	89.1	68.6	64.5	52.2
SCR (Qwen3-232B)	88.8	72.0	65.7	55.5
SCR (DeepSeek-V3)	**91.6**	71.7	**71.7**	57.3
SCR (ChatGPT)	90.9	73.4	70.4	**60.2**
SCR (GPT-4o-mini)	90.8	74.8	68.4	57.3

Table 1: Performance on KGQA benchmarks. The best results are shown in bold.

Main Results

KGQA Performance. As shown in Table 1, SCR achieves the best overall performance across most metrics on both datasets, outperforming all baselines. It ranks second in F1 score on WebQSP (74.8), slightly behind the top score of 77.5 achieved by SubgraphRAG. Among the baselines, SubgraphRAG and GNN-RAG demonstrate the strongest performance on WebQSP and CWQ, respectively. Both methods leverage the structural properties of the KG to enhance retrieval. However, they face limitations in capturing the semantic connections between questions and descriptive KG text, leading to inconsistent performance across datasets. In contrast, SCR addresses this issue by aligning textual and graph-based representations within a shared embedding space using contrastive learning. This design

Methods	WebQSP		CWQ		
	1 hop	2 hop	1 hop	2 hop	≥3 hop
RoG	73.4	63.3	**50.4**	60.7	40.0
GNN-RAG	72.0	69.8	47.4	69.4	51.8
SCR	**77.2**	**71.5**	**50.4**	**70.1**	**64.2**

Table 2: Breakdown of F1 evaluation by reasoning hops.

Methods	WebQSP		CWQ	
	Time(s)	#Tokens	Time(s)	#Tokens
RoG	948.0	521	2327.0	622
GCR	2.5	360	3.1	422
G-Retriever	672.0	-	1530.0	-
GNN-RAG	68.0	414	160.0	499
SubgraphRAG	6.0	1104	12.0	1229
SCR	**1.2**	**240**	**1.6**	**263**

Table 3: Evaluation of average retrieval time and token usage.

enables more robust generalization across datasets. Notably, among LLMs, the two smallest models (Qwen3-0.6B and Qwen3-8B) achieve performance comparable to that of much larger models when integrated with our SCR. On WebQSP, they even rank among the top three in Hit score across all baselines. This finding underscores the significant contribution of SCR to enhancing the reasoning capabilities of smaller LLMs and highlights its strong potential for resource-efficient applications.

Multi-Hop Reasoning. We compare the F1 performance of RoG (an LLM-based method), GNN-RAG (a GNN-based method), and our proposed SCR, broken down by reasoning hops. As shown in Table 2, RoG outperforms GNN-RAG in 1-hop reasoning, benefiting from the strong natural language understanding capabilities of LLMs. However, it underperforms in multi-hop reasoning, where its limited ability to capture graph structure becomes a bottleneck. GNN-RAG, by contrast, handles multi-hop reasoning more effectively due to its structural modeling, but lacks semantic depth in simple questions. SCR demonstrates superior performance in both 1-hop and multi-hop scenarios, which can be attributed to its effective integration of semantic and structural signals, along with the novel stepwise reasoning strategy introduced in this work.

Efficiency Analysis. Table 3 compares the retrieval efficiency of all methods. SCR not only achieves the lowest average retrieval time but also requires the fewest input tokens for the LLM. GCR demonstrates competitive efficiency; however, its overall performance on KGQA tasks falls short compared to that of other methods. Moreover, GCR relies on a significantly larger model compared to GNN-based methods. Under the same experimental environment, training GCR takes approximately 5 hours on WebQSP and 41 hours on CWQ, whereas training our SCR requires only 1 hour on WebQSP and 9 hours on CWQ. Another lightweight model, SubgraphRAG, achieves strong KGQA performance but consumes significantly more tokens due to its use of in-context learning (ICL) (Brown et al. 2020), which involves carefully crafted prompt templates with explanatory demonstrations to guide LLM reasoning. Overall, SCR achieves a strong balance between effectiveness and efficiency in KG-based RAG.

Methods	WebQSP		CWQ	
	Hit	F1	Hit	F1
SCR	**90.8**	**74.8**	**70.4**	**60.2**
SCR w/o multi-step	70.1	45.4	56.5	30.5
SCR w/o beam search	74.7	49.1	63.5	38.2
SCR w/o customized GNN	83.9	66.8	68.3	57.0

Table 4: Ablation studies on two KGQA benchmarks.

Ablation Studies

The strong performance of SCR is largely attributed to its well-designed relational message passing, the novel stepwise reasoning strategy, and the use of beam search guided by this strategy. To validate their contributions, we conduct ablation studies under three settings. In the first setting, where the stepwise reasoning strategy is removed (w/o multi-step), the model is trained by solely aligning the embeddings of questions and their corresponding answers, and the shortest path between the topic and answer entities is used as the reasoning path. Beam search is disabled in this case. In the second setting, where beam search is removed (w/o beam search) but the stepwise reasoning strategy is retained, the model selects the next entity at each step based on the highest similarity to the current question representation. In the third setting, where our customized GNN is replaced with R-GCN (Schlichtkrull et al. 2018) (w/o customized GNN), a model known for its strong capability in handling KGs and other multi-relational graph data.

As shown in Table 4, removing the stepwise reasoning strategy leads to a significant performance drop—20.7 in Hit and 29.4 in F1 on WebQSP, and 13.9 in Hit and 29.7 in F1 on CWQ. Disabling beam search also causes notable degradation, with decreases of 16.1 in Hit and 25.7 in F1 on WebQSP, and 6.9 in Hit and 22.0 in F1 on CWQ. Analysis of different reasoning pipelines reveals that the stepwise reasoning strategy enables more effective KG retrieval by leveraging richer semantic information, as it incorporates the embeddings of all visited entities and their neighbors. Besides, it facilitates the integration of generation strategies such as beam search, whose effectiveness in enhancing reasoning performance on KGs has been empirically verified.

Furthermore, replacing our customized GNN with R-GCN results in performance drops of 6.9 in Hit and 8.0 in F1 on WebQSP, and 2.1 in Hit and 3.2 in F1 on CWQ. R-GCN assigns dedicated learnable parameters to different relations but overlooks their semantic features. In contrast, our GNN incorporates textual embeddings of relations to enrich entity representations, resulting in improved performance.

Question	What north American country where some people speak **Portuguese** shares the **Central Time Zone**?
Answer	Canada
Reasoning Paths	Path #1: Portuguese Language → *human_language.countries_spoken_in* → Canada Path #2: Central Time Zone → *common.topic.image* → Timezoneswest → *appears_in_topic_gallery* → Samoa Time Zone → *time_zone.locations_in_this_time_zone* → United States of America Segment #1: → *primarily_containedby* → Detroit River → *partially_containedby* → Canada Segment #2: → *partially_contains* → North America → *countries_within* → Canada . . . SCR: → *country.languages_spoken* → American English → *country.countries_spoken_in* → Canada
ChatGPT	Mexico
w/ Shortest Paths	United States of America
w/ SCR	Canada

Table 5: An example from CWQ. **Bold** text indicates the topic entities mentioned in the question. Underlined entities highlight key factors influencing LLM reasoning. *Italicized* text denotes the relation between entities. Red marks the differing remaining paths of Path #2.

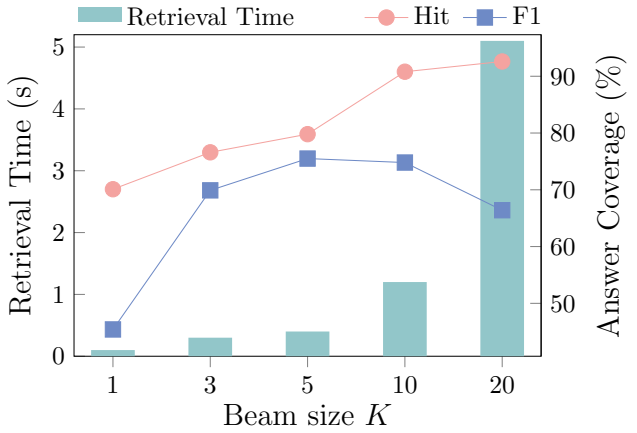

Figure 3: Efficiency and performance comparison across beam sizes on WebQSP.

Impact of Varying Beam Sizes

Our work introduces beam search to explore multiple possible paths at each reasoning step. While this increases the likelihood of identifying correct entities, it also introduces additional computational overhead. To examine the impact of different beam sizes K (as referenced in Equation 15), we conduct experiments on WebQSP with $K \in \{1, 3, 5, 10, 20\}$. As shown in Figure 3, increasing K leads to longer generation times and higher Hit scores. However, the F1 score peaks at $K = 5$ and then declines due to the inclusion of more irrelevant predictions than informative ones. Overall, SCR achieves a good balance between reasoning performance and efficiency when using $K = 10$.

Case Study

We provide a case study to further demonstrate the effectiveness of our stepwise reasoning strategy. As shown in Table 5, the question involves two topic entities: *Portuguese* and *Central Time Zone*. Path #1 and Path #2 represent their respective reasoning paths. The first path correctly identifies Canada as a Portuguese-speaking country in North America. For the second path, there are 74 shortest paths between *Central Time Zone* and *Canada*, which share the same prefix subpath but diverge after the entity *United States of America*. We use ChatGPT as the LLM reasoner to answer the question. Without reasoning paths as supporting facts, ChatGPT incorrectly selects Mexico as the answer, reasoning that Portuguese is spoken by some immigrants there. This rationale is weak, as it applies to multiple North American countries with immigrant populations. We then compare two reasoning strategies. The first, commonly used in GNN-based methods, retrieves the answer in a single step and generates the shortest paths between the topic and answer entities. The second is our proposed stepwise reasoning strategy, which constructs specific reasoning paths dynamically during retrieval. When provided with all possible shortest paths, ChatGPT produces the incorrect answer *United States of America*. In contrast, when guided by SCR's reasoning paths, ChatGPT correctly answers *Canada*. To investigate further, we present the shortest paths one by one. The results show that ChatGPT can infer the correct answer only when given the path that explicitly states the language spoken in the United States. This information serves as a critical clue. However, in single-step reasoning, such useful signals are often buried among many irrelevant ones, causing the model to suffer from the "lost in the middle" phenomenon (Liu et al. 2024b). In contrast, SCR effectively filters and prioritizes question-relevant information during retrieval, resulting in more accurate and grounded LLM generation.

Conclusion

We propose SCR, a KG-based RAG framework that performs interpretable KG retrieval by stepwise incorporating textual and structural information via a lightweight network design. SCR achieves strong effectiveness and efficiency in KGQA tasks. In future work, we aim to develop algorithms for filtering erroneous reasoning paths from candidate results, with the goal of further improving retrieval precision.

Acknowledgments

This work is supported by the Solfeggio ear training intelligent robot and cloud platform research and development project for music education (No.2024CXY0102), the 3D visualization digital twin integrated control system (No.2023CXY0111), the public technology service platform project of Xiamen City (No.3502Z20231043) and Fujian Provincial Science and Technology Major Project (No.2024HZ022003)

References

Baek, J.; Aji, A. F.; and Saffari, A. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. arXiv:2306.04136.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; and Hoefler, T. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Proceedings of the AAAI Conference on Artificial Intelligence*, 17682–17690. Vancouver, Canada: AAAI Press.

Bollacker, K. D.; Evans, C.; Paritosh, P. K.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Wang, J. T., ed., *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1247–1250. Vancouver, BC, Canada: ACM.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 1877–1901. Curran Associates, Inc.

Dehghan, M.; Alomrani, M. A.; Bagga, S.; Alfonso-Hermelo, D.; Bibi, K.; Ghaddar, A.; Zhang, Y.; Li, X.; Hao, J.; Liu, Q.; Lin, J.; Chen, B.; Parthasarathi, P.; Biparva, M.; and Rezagholizadeh, M. 2024. EWEK-QA : Enhanced Web and Efficient Knowledge Graph Retrieval for Citation-based Question Answering Systems. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14169–14187. Bangkok, Thailand: Association for Computational Linguistics.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *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)*, 4171–4186. Minneapolis, MN, USA: Association for Computational Linguistics.

Dong, Z.; Peng, B.; Wang, Y.; Fu, J.; Wang, X.; Zhou, X.; Shan, Y.; Zhu, K.; and Chen, W. 2025. EffiQA: Efficient Question-Answering with Strategic Multi-Model Collaboration on Knowledge Graphs. In Rambow, O.; Wanner, L.; Apidianaki, M.; Al-Khalifa, H.; Eugenio, B. D.; and Schockaert, S., eds., *Proceedings of the 31st International Conference on Computational Linguistics*, 7180–7194. Abu Dhabi, UAE: Association for Computational Linguistics.

Federico, M.; Cettolo, M.; Brugnara, F.; and Antoniol, G. 1995. Language modelling for efficient beam-search. *Comput. Speech Lang.*, 9(4): 353–379.

He, G.; Lan, Y.; Jiang, J.; Zhao, W. X.; and Wen, J. 2021. Improving Multi-hop Knowledge Base Question Answering by Learning Intermediate Supervision Signals. In Lewin-Eytan, L.; Carmel, D.; Yom-Tov, E.; Agichtein, E.; and Gabrilovich, E., eds., *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 553–561. Virtual Event, Israel: ACM.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9726–9735. Seattle, WA, USA: Computer Vision Foundation / IEEE.

He, X.; Tian, Y.; Sun, Y.; Chawla, N. V.; Laurent, T.; LeCun, Y.; Bresson, X.; and Hooi, B. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, 132876–132907. Vancouver, BC, Canada: Curran Associates, Inc.

Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.; Song, X.; and Zhou, D. 2024. Large Language Models Cannot Self-Correct Reasoning Yet. In *The Twelfth International Conference on Learning Representations*. Vienna, Austria: OpenReview.net.

Jiang, J.; Zhou, K.; Zhao, X.; and Wen, J. 2023. UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph. In *The Eleventh International Conference on Learning Representations*. Kigali, Rwanda: OpenReview.net.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, 9459–9474. Curran Associates, Inc.

Li, M.; Miao, S.; and Li, P. 2025. Simple is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation. In *The Thirteenth International Conference on Learning Representations*. Singapore: OpenReview.net.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; et al. 2024a. DeepSeek-V3 Technical Report. arXiv:2412.19437.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024b. Lost in the Middle:

How Language Models Use Long Contexts. *Trans. Assoc. Comput. Linguistics*, 12: 157–173.

Luo, L.; Li, Y.; Haffari, G.; and Pan, S. 2024a. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *The Twelfth International Conference on Learning Representations*. Vienna, Austria: OpenReview.net.

Luo, L.; Zhao, Z.; Gong, C.; Haffari, G.; and Pan, S. 2024b. Graph-constrained Reasoning: Faithful Reasoning on Knowledge Graphs with Large Language Models. In *Forty-second International Conference on Machine Learning*. Vancouver, BC, Canada: PMLR.

Luo, L.; Zhao, Z.; Haffari, G.; Phung, D. Q.; Gong, C.; and Pan, S. 2025. GFM-RAG: Graph Foundation Model for Retrieval Augmented Generation. arXiv:2502.01113.

Mavromatis, C.; and Karypis, G. 2022. ReaRev: Adaptive Reasoning for Question Answering over Knowledge Graphs. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2447–2458. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Mavromatis, C.; and Karypis, G. 2025. GNN-RAG: Graph Neural Retrieval for Efficient Large Language Model Reasoning on Knowledge Graphs. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 16682–16699. Vienna, Austria: Association for Computational Linguistics.

OpenAI. 2022. ChatGPT Introduction. https://openai.com/index/chatgpt. Accessed: 2025-07-01.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Trans. Knowl. Data Eng.*, 36(7): 3580–3599.

Schlichtkrull, M. S.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In Gangemi, A.; Navigli, R.; Vidal, M.; Hitzler, P.; Troncy, R.; Hollink, L.; Tordai, A.; and Alam, M., eds., *The Semantic Web - 15th International Conference, ESWC*, volume 10843 of *Lecture Notes in Computer Science*, 593–607. Heraklion, Crete, Greece: Springer.

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. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4231–4242. Brussels, Belgium: Association for Computational Linguistics.

Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L. M.; Shum, H.; and Guo, J. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *The Twelfth International Conference on Learning Representations*. Vienna, Austria: OpenReview.net.

Talmor, A.; and Berant, J. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 641–651. New Orleans, Louisiana, USA: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, 5998–6008. Long Beach, CA, USA: Curran Associates, Inc.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations*. Vancouver, BC, Canada: OpenReview.net.

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:2308.13259.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023b. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*. Kigali, Rwanda: OpenReview.net.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, 24824–24837. New Orleans, LA, USA: Curran Associates, Inc.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; et al. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, 11809–11822. New Orleans, LA, USA: Curran Associates, Inc.

Yih, W.; Richardson, M.; Meek, C.; Chang, M.; 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)*. Berlin, Germany: The Association for Computer Linguistics.

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. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5773–5784. Dublin, Ireland: Association for Computational Linguistics.