# DRAoG: A Dynamic Reasoning Adaptive on Graphs framework for Large Language Models

Zili Zhou[1,*], Nuo Zhuang[1], Chengyuan Xue[1]
[1]School of Cyber Science and Engineering, Qufu Normal University, Qufu, China
*Corresponding author: zzl@qfnu.edu.cn

***Abstract*—Large Language Models (LLMs) have shown strong capabilities in knowledge-intensive reasoning tasks, but still face challenges in large-scale Knowledge Graph (KG) scenarios, including hallucinations, inefficient multi-hop reasoning, and noise sensitivity. These issues mainly arise from fixed-hop search, discrete decision boundaries, and subgraph linearization, which introduce contextual noise and computational overhead.**

**To address these issues, we propose DRAoG, a training-free adaptive dynamic graph reasoning framework. It reduces irrelevant information through relation-level subgraph filtering, dynamically controls reasoning depth via a state-aware mechanism, and ensures answer–evidence consistency through semantic matching-based verification, thereby achieving a balance between efficiency and reliability.**

**Experimental results on four KGQA benchmark datasets show that DRAoG improves accuracy by 3.0%–8.4% on multi-hop tasks while reducing LLM API calls by 15%, demonstrating its effectiveness in enhancing performance and lowering computational cost. Our code and data are publicly available at https://github.com/nuo0214/DRAoG .**

***Index Terms*—KGs, LLMs, Multi-Hop Question Answering, KGQA**

## I. Introduction

In recent years, Large Language Models (LLMs) have demonstrated remarkable capabilities in language understanding and generation through large-scale pre-training, achieving significant success in tasks such as question answering [1] [2]. However, LLMs face challenges in knowledge-intensive tasks, including outdated information and hallucinations caused by parameterized knowledge, as well as a lack of transparency and susceptibility to errors in complex reasoning processes [3].

To address these issues, recent studies [4] [5] have introduced Knowledge Graphs (KGs) as external knowledge sources for LLMs. The structured relationships in KGs can support explicit reasoning paths and enhance factual reliability [5] [6].

Currently, the mainstream paradigms for integrating LLMs with KGs can be divided into two categories. The first category injects KG knowledge into model parameters during the pre-training [7] or fine-tuning stages [8]. However, such static approaches fail to address inherent limitations during deployment. Methods like K-BERT [9] and KEPLER [7] can improve factual coverage, but they suffer from high computational costs, inflexible knowledge updates, and a tendency toward catastrophic forgetting.

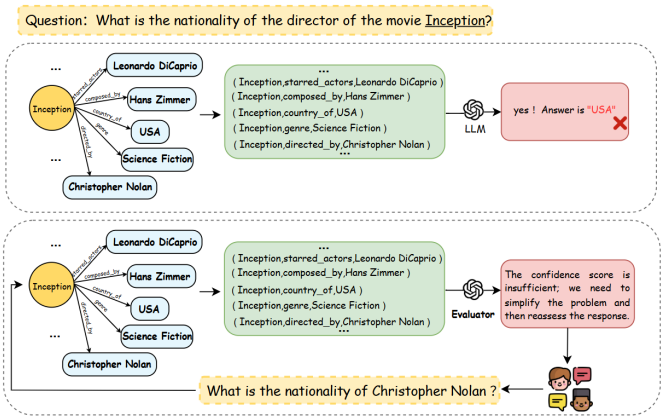


Fig. 1. Illustration of challenge and our solution

The second category dynamically retrieves KG information as context during the inference stage. Early approaches, such as KG-BART [10], linearize subgraphs as input to the model. More recently, DoG [11] enables LLMs to perform iterative "thought–query" reasoning over KGs, representing an important advancement in interactive reasoning. However, when handling highly connected entities, it linearizes all related relations into the input, leading to context explosion and noise interference.

Despite the progress made by these two paradigms, significant limitations remain. First, during the evidence retrieval stage, the signal-to-noise ratio is often low, as irrelevant relations lead to context bloat. Second, fixed-hop reasoning strategies cannot dynamically adjust the number of reasoning steps, resulting in either wasted resources or insufficient reasoning (as shown in Fig. 1). Moreover, there is a lack of verification mechanisms to ensure semantic consistency between the generated answers and the evidence subgraphs.

To address the aforementioned challenges, this work proposes DRAoG—a Dynamic Reasoning Adaptive on Graphs framework for multi-hop question answering with large language models. It is designed to explicitly balance reasoning accuracy and efficiency while achieving end-to-end optimization for Knowledge Graph Question Answering (KGQA) tasks. Our primary contributions are summarized as follows:

- We propose a retrieval-to-generation closed-loop framework that integrates relation-level subgraph filtering, state-aware decision-making, and semantic matching-based verification, improving reasoning accuracy and

efficiency without requiring additional training.

- An adaptive termination mechanism with state awareness dynamically adjusts reasoning depth based on evidence sufficiency, reducing redundant traversals and unnecessary LLM calls while lowering overall reasoning cost.
- A semantic matching-based grounding verification mechanism aligns generated answers with evidence subgraphs in a shared embedding space, suppressing unsupported generation and enhancing factual consistency.

Extensive experiments conducted on four public KGQA benchmark datasets demonstrate that DRAoG excels in both accuracy and reasoning efficiency. Results show improvements of 3.0% to 8.4% on multi-hop datasets, while reducing LLM API calls by 15%.

## II. RELATED WORK

### A. Dense Retrieval and Subgraph Selection

In Knowledge Graph Question Answering (KGQA) tasks, precisely locating evidence subgraphs from massive graph data serves as the foundation for subsequent reasoning. Traditional symbolic matching methods mainly rely on entity linking and fuzzy string matching, which are highly susceptible to entity variants and lexical diversity, resulting in low recall.

To address this issue, recent research has shifted towards embedding-based and Large Language Model (LLM)-augmented paradigms. For instance, G-Retriever [12] innovatively utilizes Graph Neural Networks (GNNs) and soft prompt techniques to map graph structural information into the latent space of the LLM, achieving a deep fusion of graph-text modalities. Similarly, GraphRAG [13] proposes a local-to-global retrieval strategy that enhances the LLM's understanding of complex queries through hierarchical graph summaries.

However, existing frameworks such as ToG [3] and RoG [1] often face challenges related to context noise while pursuing reasoning breadth. These methods typically retain a large number of low-relevance neighbor nodes, causing the limited context window of the LLM to be filled with redundant information. This not only leads to "context explosion" but also increases the risk of model hallucinations.

### B. Adaptive Reasoning Strategies

In exploring the reasoning capabilities of Large Language Models (LLMs), research has evolved from Chain-of-Thought (CoT) [15] to more structured approaches such as Tree of Thoughts (ToT) [16] and Graph of Thoughts (GoT) [17], often combined with self-correction mechanisms to improve reasoning accuracy [14]. However, LLMs still suffer from issues such as outdated knowledge, hallucinations, and lack of transparency in decision-making, motivating efforts to integrate them with Knowledge Graphs (KGs) for more reliable reasoning [18].

In KG reasoning, the key challenge lies in path planning. Existing methods such as StructGPT [4] and ToG [3] mainly adopt fixed-hop strategies, which struggle to balance efficiency and robustness: simple queries may introduce noise and error propagation, while complex queries are limited by insufficient search depth. To address this, recent work has explored dynamic mechanisms. For example, HippoRAG [19] improves retrieval but still lacks effective termination decisions, while Adaptive-RAG [20] demonstrates the effectiveness of adapting strategies based on query complexity, highlighting the importance of adaptive reasoning mechanisms.

## III. METHODOLOGY

To address the limitations of existing methods in complex reasoning, we propose DRAoG, a dynamic graph-grounded reasoning framework following a "retrieval-decision-verification" pipeline. Given a query and a topic entity, the framework first retrieves an evidence subgraph, then dynamically determines whether further reasoning is needed, and verifies the consistency between the generated answer and the evidence. This design enables DRAoG to balance retrieval precision, reasoning efficiency, and factual faithfulness in Knowledge Graph Question Answering (KGQA). An overview of the framework is shown in Fig. 2.

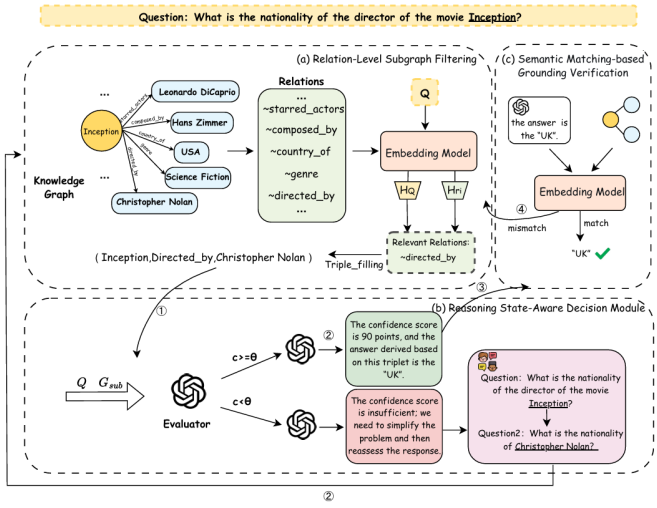


Fig. 2. The framework overview of DRAoG, which includes three key components: Relation-Level Subgraph Filtering, Reasoning State-Aware Decision Module, Semantic Matching-based Grounding Verification.

### A. Relation-Level Subgraph Filtering

To mitigate the context noise introduced by highly connected entities, we perform relation-level filtering prior to subgraph expansion. Specifically, we employ an instruction-tuned encoder to project both the input query and candidate relation descriptions into a shared semantic embedding space. We then calculate the cosine similarity to identify and select the top-K relations that are most relevant to the current reasoning target, allowing us to construct a compact and focused evidence subgraph.

This targeted filtering strategy preserves the essential local graph structure while removing irrelevant relational branches, thereby improving the signal-to-noise ratio in the downstream reasoning process.

1) Entity Anchoring and Adjacent Relations Extraction

Given a natural language query $Q$, the system first anchors the topic entity $e_h$. From the knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, we extract its one-hop relational neighborhood $\mathcal{N}_r(e_h)$. Relation types are the core elements determining the direction and semantics of reasoning paths. Therefore, converting the complete subgraph retrieval task into the filtering of starting relation edges essentially constructs a high-quality set of path starting points for multi-step reasoning.

2) Semantic Projection: Vectorization of Relations and Queries

To achieve projection from the discrete graph to the semantic space and break through the evidence bottleneck, we employ an instruction-tuned dense encoder as the semantic projection operator. For the query $Q$, we construct an instruction-augmented sequence $X_Q = [\texttt{CLS}] \oplus I \oplus Q \oplus [\texttt{SEP}]$. After passing through the encoder, we obtain the deep representation $H_Q$. We take the vector corresponding to the $[\texttt{CLS}]$ token, $z_Q = H_Q^{(0)}$, and normalize it via $L_2$ normalization to obtain the query embedding $e_Q$:

$$e_Q = \frac{z_Q}{\|z_Q\|_2} \in \mathbb{R}^d \tag{1}$$

Similarly, for a candidate relation $r_i \in \mathcal{N}_r(e_h)$, its textual description is fed into the same encoder to obtain $H_{r_i}$, and its normalized relation embedding vector $e_{r_i}$ is acquired in a similar manner. In the projected semantic space, the evidence score of relation $r_i$ relative to query $Q$ is defined as its cosine similarity with the query embedding:

$$\text{score}(r_i) = e_Q^\top e_{r_i} \tag{2}$$

This score quantifies the potential utility of the relation as supporting evidence. Based on these scores, a graph pruning operator is implemented: the top-$K$(K=15) relations with the highest evidence scores are selected from $\mathcal{N}_r(e_h)$ to form the refined relation subset $\hat{\mathcal{R}}$. This process is formalized as an optimization problem with cardinality constraints, directly compressing the evidence search space.

3) Evidence Subgraph Construction

This layer performs back-projection to re-instantiate the semantic relation subset $\hat{\mathcal{R}}$ into concrete graph structures, generating a high-purity evidence subgraph $G_{sub}$ for subsequent reasoning:

$$\begin{aligned} G_{\text{sub}} = &\{(e_h, r, t) \in \mathcal{G} \mid r \in \hat{\mathcal{R}}\} \\ &\cup \{(t, r, e_h) \in \mathcal{G} \mid r \in \hat{\mathcal{R}}\} \end{aligned} \tag{3}$$

By aggregating bidirectional neighborhood information, this set effectively constructs a local reasoning subgraph with a high signal-to-noise ratio, which is then directly injected into the reasoning module to drive decision-making.

### *B. Reasoning State-Aware Decision Module*

Traditional iterative reasoning frameworks typically rely on a fixed number of hops. This often leads to redundant computation for simple questions and premature termination for complex ones. To address this issue, this study designs a Reasoning State-Aware Decision Module. This module upgrades the termination mechanism from static rules to a dynamic decision-making process that explicitly balances evidence sufficiency against reasoning costs.

1) Reasoning State-Aware Dynamic Judgment Mechanism

At each reasoning step $t$, DRAoG aims to estimate whether the currently retrieved evidence is sufficient to support answer generation. The system takes the current query $Q_t$ and the retrieved evidence subgraph $G_{sub}$ as inputs. By comprehensively evaluating three dimensions—query relevance, evidence coverage, and logical coherence—it calculates a structured evidence sufficiency score $c_t \in [0, 100]$.This score $c_t$ can be interpreted as a scalar estimation of the degree to which the evidence subgraph $G_{sub}$ sufficiently supports a correct answer to the query $Q_t$. Formally, the following correlation can be established:

$$c_t \propto S(Q_t, G_{\text{sub}}) \tag{4}$$

Where $S(\cdot)$ represents the evidence sufficiency measurement function.

We discard the fixed-hop strategy and introduce a step-aware dynamic decision threshold. As the reasoning depth increases, this threshold gradually relaxes the termination criteria, reflecting the trade-off between additional retrieval costs and marginal reasoning gains. We define a dynamic decision boundary function, $\mathcal{T}(t)$, which evolves with the reasoning state and adopts an efficient linear decay strategy:

$$\mathcal{T}(t) = \max(\tau_{\text{base}} - \delta \cdot t, \tau_{\text{min}}) \tag{5}$$

Here, $\mathcal{T}(t)$ serves as the decision boundary function, representing a cost-sensitive dynamic threshold. Based on this, the system's decision logic is formalized as follows:

$$\text{Decision}(t) = \begin{cases} \text{Terminate}, & \text{if } c_t \geq \mathcal{T}(t) \\ \text{Continue}, & \text{otherwise} \end{cases} \tag{6}$$

This mechanism simulates a near-optimal decision-making process under resource constraints: it applies strict standards in the early stages of reasoning, which is equivalent to pursuing high precision when computational costs are low. As the number of reasoning steps increases and cumulative costs rise, the system appropriately relaxes precision requirements when the expected marginal returns begin to diminish, thereby achieving optimal global efficiency.

2) Collaborative Query Refinement

When the decision outcome is "Continue," it indicates that the current evidence is insufficient to support an answer. In this case, the module triggers a chained query optimization process consisting of three key steps: complexity decomposition, logical validation, and semantic reconstruction. This process transforms the current complex query $Q_t$ into a more focused sub-query $Q_{t+1}$ to guide the next "retrieve-and-evaluate" cycle. Essentially, this iteratively decomposes high-hop problems into low-hop sub-problems, ensuring that

subsequent retrieval steps have a clear and directed focus.

3) Decision Scheduling Logic

The system executes a strict hierarchical scheduling strategy based on the evaluation results:

- High-Confidence Early Stopping: if $c_t \geq \mathcal{T}(t)$, the system determines that the current evidence has passed the dynamic sufficiency check. It immediately terminates the retrieval process and proceeds to the answer generation phase.
- Iterative Deepening: Otherwise, the system continues the reasoning process using the refined query $Q_{t+1}$.
- Parametric Knowledge Fallback: If the termination criteria are still not met upon reaching the maximum number of steps, the system smoothly falls back to the LLM's internal parameterized knowledge to generate an answer, ensuring the final robustness of the system.

### C. Semantic Matching-based Grounding Verification

Traditional generative models often suffer from "hallucinations" because they tend to ignore the retrieved context. To ensure the factual faithfulness of the generated content, this module introduces a grounding verification step at the end of the reasoning process. We project both the generated candidate answer $A_{gen}$ and the retrieved core evidence subgraph $G_{sub}$ into the same dense vector space.Specifically, we utilize the encoder described in Section III.A to calculate a semantic entailment score, $Score_{consistency}$, between $A_{gen}$ and $G_{sub}$. Based on this score, a threshold $\theta$ is set to determine whether the generated content deviates from the graph evidence. The decision logic is formalized as:

$$\text{IsGrounded} = \mathbb{I}\left(\text{Sim}(E(A_{\text{gen}}), E(G_{\text{sub}})) \geq \theta\right) \tag{7}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, and $E(\cdot)$ represents the embedding function. If IsGrounded is false, a fallback mechanism is triggered, suppressing the output of non-factual content at its source and significantly improving system robustness. As a post-generation filter, this lightweight verification mechanism enhances factual consistency without introducing additional training overhead.

## IV. EXPERIMENTS

### A. Datasets and Knowledge Graph

To evaluate our framework, we utilized four public datasets: WebQSP [21], CWQ [22], WebQuestions [23], and GrailQA [24]. WebQSP consists of questions derived from the WebQuestions dataset, while CWQ is specifically designed for complex questions that require reasoning across multiple web fragments. GrailQA tests three levels of generalization: independent (id), compositional, and zero-shot learning. The sample counts for the training and test sets across these four datasets are summarized in Table I .

Following the methodology of previous works such as DoG [11]and ToG [3], we leverage the high-order topic entities provided by the benchmarks to ensure a fair comparison.

TABLE I
STATISTICAL INFORMATION OF FOUR KGQA DATASETS.

| Dataset | Test | Train |
|---|---|---|
| CWQ [21] | 1000 | 27734 |
| WebQSP [22] | 1639 | 3098 |
| GrailQA [23] | 1000 | 14894 |
| WebQuestion [24] | 2032 | 3778 |

All four datasets are based on the Freebase KG [25]. For all experiments, we employed the version of Freebase preprocessed following the methods of Lan [26] and Jiang [27]. This preprocessing strategy extracts only triples containing entities with Freebase IDs, English text, or numeric formats from the complete Freebase.

### B. Baselines

For the baselines, considering the performance variations of different methods across datasets, we compare our framework against representative approaches from both Supervised Learning and In-Context Learning paradigms to comprehensively validate its effectiveness and superiority.Supervised Learning: KV-Mem [28], GraftNet [29], PullNet [30], EmbedKGQA [31], NSM [32], TransferNet [33], and UniKGQA [34].In-Context Learning: StructGPT [4], KG-GPT [35], KB-BINDER [36], ToG [3], and DoG [11].

### C. Experimental Setups

This study established an experimental environment based on the Freebase and deployed a local Virtuoso service to facilitate efficient querying. Regarding model configuration, we employed GPT-3.5-turbo as the backbone Large Language Model (LLM). The key parameters of the framework are set as follows: the dynamic decision threshold follows a linear decay function, with the parameters assigned as $\tau_{base} = 80$, $\delta = 10$, and $\tau_{min} = 50$.We utilized the same test samples as DoG [12] to enhance computational efficiency and adopted the standard Exact Match (Hits@1) as the evaluation metric. All experiments were conducted using the PyTorch framework on a single NVIDIA RTX A5000 GPU.

### D. Main Results

To comprehensively evaluate the effectiveness of the proposed framework, we conducted experiments on four widely used Knowledge Base Question Answering benchmark datasets: WebQSP, ComplexWebQuestions (CWQ), WebQuestions (WebQ), and GrailQA.Table II presents the performance comparison between our method and the baselines. The experimental results show that DRAoG achieves performance improvements across all datasets.

The detailed analysis is as follows:

**Noise Resistance in Complex Multi-hop Scenarios.** On the complex multi-hop dataset CWQ, the model improves accuracy by 8.4% (from 58.2% to 66.6%). This gain is mainly attributed to relation-level subgraph filtering, which effectively reduces context noise introduced by "hub nodes" in Freebase,

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON KGQA DATASETS (HIT@1)

| Method | Class | LM | CWQ | WebQSP | GrailQA | WebQ |
|---|---|---|---|---|---|---|
| KV-Mem [28] | SL | - | 18.4 | 46.7 | - | - |
| GraftNet [29] | | - | 36.8 | 66.4 | - | - |
| PullNet [30] | | - | 45.9 | 45.9 | - | - |
| EmbedKGQA [31] | | RoBERTa | - | 66.6 | - | - |
| NSM [32] | | - | 47.6 | 68.7 | - | - |
| TransferNet [33] | | BERT | 48.6 | 71.4 | - | - |
| UniKGQA [34] | | RoBERTa | 77.2 | 77.2 | - | - |
| StructGPT [4] | ICL | GPT-3.5-Turbo | - | 72.6 | - | - |
| KG-GPT [35] | | GPT-3.5-Turbo | - | - | - | - |
| KB-BIINDER [36] | | Codex | - | 74.4 | - | - |
| ToG [3] | | GPT-3.5-Turbo | 57.1 | 76.2 | 68.7 | 54.5 |
| DOG [11] | ICL | GPT-3.5-Turbo | 58.2 | 88.6 | 77.8 | 78.2 |
| DRAoG | | GPT-3.5-Turbo | **66.6** | **93.6** | **85.6** | **81.2** |

and the reasoning state-aware mechanism that dynamically adjusts search depth based on confidence, alleviating truncation issues in complex reasoning chains.

**Generalization Performance in Zero-shot Scenarios.** On the GrailQA dataset under zero-shot settings, the method achieves a 7.8% improvement. The semantic matching-based grounding verification mechanism bridges the lexical gap between generated answers and graph evidence, significantly enhancing generalization to unseen entities.

**Efficiency Balance on Simple Questions.** On relatively simpler datasets such as WebQSP and WebQ, improvements of 5.0% and 3.0% are observed, with WebQSP reaching an accuracy of 93.6%. These results indicate that the adaptive termination mechanism effectively enables early stopping based on question difficulty, reducing redundant computation and noise while achieving a strong balance between reasoning accuracy and computational efficiency.

### *E. Ablation Studies*

To systematically analyze the individual contributions and synergistic effects of the components within DRAoG, we conducted ablation studies across four multi-hop question-answering datasets. Our investigation focused on the effectiveness of three core modules: Relation-level Subgraph Filtering (Filter), Reasoning State-Aware Decision-making (Decision), and Semantic Matching-based Grounding Verification (Verify).

The experiment defines four progressive model configurations: Base is the baseline model without any integrated optimization modules; w/o (Decision and Verify) introduces only the relation-level subgraph filtering module (Filter) on top of the baseline; w/o Verify further incorporates the reasoning state-aware decision module (Decision); and DRAoG is the complete framework integrating all three modules (Filter, Decision, and Verify) to verify the robustness of Grounding Verification in the final generation stage.The ablation study results are shown in Table III.

Relation-level subgraph filtering (Filter) significantly improves system performance, increasing accuracy on CWQ from 58.2% to 61.3% (+3.1%), demonstrating that it narrows the candidate fact space and provides more precise knowledge for subsequent reasoning. Building on this, the reasoning state-aware decision module (Decision) further enhances performance, improving accuracy from 91.0% to 93.2% (+2.2%) on WebQSP and from 61.3% to 65.6% (+4.3%) on CWQ. This indicates that the cost-aware early stopping mechanism reduces redundant search for simple queries and mitigates noise introduced by over-reasoning, achieving a balance between accuracy and efficiency. With the addition of the semantic matching-based grounding verification module (Verify), the full DRAoG framework reaches 93.6% on WebQSP and 66.6% on CWQ. By aligning generated answers with evidence subgraphs in a shared semantic space, this module effectively suppresses hallucinations and improves consistency between outputs and retrieved evidence (as shown in Fig. 3).

TABLE III
MODEL ABLATION STUDY OF OUR DRAOG FRAMEWORK.

| Method | CWQ | WebQSP | GrailQA | WebQuestion |
|---|---|---|---|---|
| Base | 58.2 | 88.6 | 77.8 | 78.2 |
| w/o Decision and Verify | 61.3 | 91.0 | 80.2 | 79.7 |
| w/o Verify | 65.6 | 93.2 | 83.3 | 80.4 |
| **DRAoG** | **66.6** | **93.6** | **85.6** | **81.2** |

### *F. Decision Mechanism Analysis*

To evaluate the effectiveness of the proposed reasoning state-aware decision module, we randomly sampled 200 instances from the WebQSP test set and analyzed the relationship between the evidence sufficiency score $c_t$, system behavior, and answer correctness. As shown in Table IV, we observed the following:

1) When the confidence score is high, the system generates answers primarily based on the retrieved knowledge graph evidence and maintains a high level of accuracy.

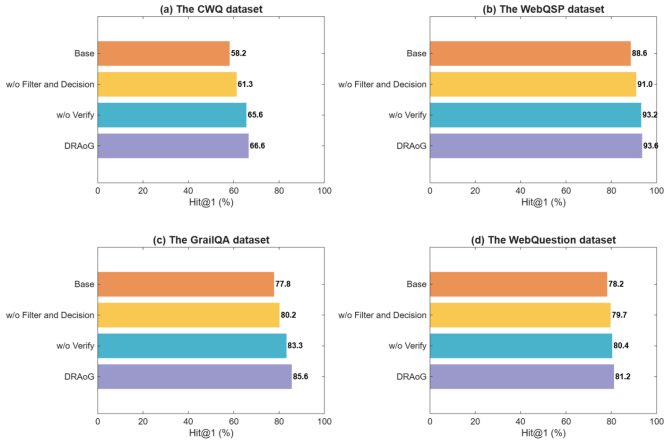


Fig. 3. Visualization of the Ablation Studies analysis on four knowledge-based question-answering datasets.

2) When the confidence score is low, it indicates that the system considers the knowledge graph evidence insufficient even after completing the maximum number of iterations; in these cases, the final answer is generated using the large language model's internal parametric knowledge.

Additionally, the early stopping decisions made in the first reasoning round (with a threshold of 80) exhibit high reliability, as the vast majority of queries meeting this condition result in correct answers. This result confirms that the module can use $c_t$ to reliably assess the state of evidence, thereby guiding the reasoning process dynamically and efficiently.

TABLE IV
ANALYSIS OF CONFIDENCE SCORE $c_t$ AND SYSTEM BEHAVIOR

| **Confidence Interval** ($c_t$) | ***N*** | **Accuracy** | **System Behavior** |
|---|---|---|---|
| $c_t \geq 80$ | 130 | 97.1% | Early-Stop |
| $50 \leq c_t < 80$ | 53 | 89.6% | Iterative Reasoning |
| $c_t < 50$ | 17 | 35.0% | Fallback |

*Note: N denotes the number of samples.*

## G. Efficiency Analysis

1) Comparison with Baselines

The computational efficiency of DRAoG was evaluated alongside two key baselines, ToG [3]and DoG [12], in supporting KG-enhanced LLMs. As summarized in Table V, a comparison was conducted across four datasets focusing on the average number of LLM calls, token consumption, and time required per question.

Results demonstrate that DRAoG achieves improvements in efficiency. Compared to ToG and DoG, DRAoG reduces the required LLM calls by approximately 83% and 15%, respectively. Regarding token consumption, DRAoG achieves reductions of 65% (input) / 56% (output) against the first baseline and 85% (input) / 37% (output) against the second. In terms of inference speed, DRAoG delivers a relative acceleration of 77% and 55% compared to ToG and DoG. These improvements stem from the synergistic design of its core components. The relation-level subgraph filtering module compresses the search space by retaining the top 15 most relevant relations, which lowers semantic matching costs. Simultaneously, the reasoning state-aware decision module utilizes a cost-sensitive dynamic boundary to trigger an early stopping mechanism, thereby reducing redundant graph traversals and unnecessary model calls. Together, these mechanisms enable the framework to increase end-to-end efficiency and reduce resource consumption while maintaining competitive accuracy.

TABLE V
EFFICIENCY COMPARISON OF DIFFERENT METHODS

| **Dataset** | **Method** | **LLM Calls** | **Input Tokens** | **Output Tokens** | **Time (s)** |
|---|---|---|---|---|---|
| CWQ | ToG [3] | 22.6 | 8182.9 | 1486.4 | 96.5 |
| | DoG [11] | 3.08 | 6562.4 | 240.8 | 44.39 |
| | **DRAoG** | **2.98** | **2196.7** | **197.3** | **33.5** |
| WebQSP | ToG [3] | 15.9 | 6031.2 | 987.7 | 63.1 |
| | DoG [11] | **2.27** | 3883.9 | 112.9 | 22.32 |
| | **DRAoG** | 2.49 | **1782.3** | **84.84** | **13.0** |
| GrailQA | ToG [3] | 11.1 | 4066 | 774.6 | 50.2 |
| | DoG [11] | 3.1 | 9857.3 | 235.7 | 20.9 |
| | **DRAoG** | **2.80** | **5889.3** | **131.3** | **10.5** |
| WebQuestions | ToG [3] | 14.7 | 7021.45 | 1275.9 | 59.4 |
| | DoG [11] | 2.84 | 6892 | 177.9 | 23.6 |
| | **DRAoG** | **2.68** | **3792** | **110.8** | **16.3** |

2) Impact of Adaptive Termination

To evaluate the reasoning state-aware decision mechanism, this study analyzes the distribution of early termination across different datasets. As shown in Fig. 4, GrailQA achieves the highest early termination rate (89.8%), followed by WebQSP (78.8%), WebQuestions (70.0%), and CWQ (65.2%).

Although WebQuestions mainly consists of single-hop queries, its early termination rate (70.0%) is lower than that of GrailQA (89.8%) and WebQSP (78.8%). This is due to colloquial expressions and semantic ambiguity, which increase uncertainty in entity and relation linking and require more reasoning steps to accumulate evidence. In contrast, while GrailQA involves complex tasks, its more structured queries enable faster accumulation of high-confidence evidence and earlier termination. CWQ, being the most complex multi-hop dataset, requires exploration of more reasoning paths and thus has the lowest early termination rate.

Overall, early termination is influenced not only by the number of reasoning hops but also by the quality and speed of evidence accumulation. Approximately 76% of samples avoid redundant computation through this mechanism, indicating that the module dynamically adjusts the iteration depth based on the reasoning state, achieving a balance between efficiency and effectiveness.

# V. CONCLUSION

To address the limitations of existing KGQA reasoning frameworks in efficiency, robustness, and evaluation standards,

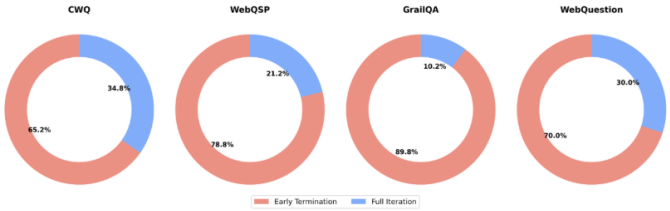


Fig. 4. Distribution of Early Termination vs. Full Iteration Across Different Datasets

this paper proposes DRAoG. The framework reduces computational cost through relation-level subgraph filtering, a reasoning state-aware decision module, and semantic matching-based grounding verification, while balancing reasoning overhead with output quality via dynamic strategies. Experiments show it improves accuracy and reduces resource consumption on complex queries.

## ACKNOWLEDGMENT

This work was supported in part by the Shandong Provincial Natural Science Foundation (No.ZR2023MF085)

## REFERENCES

[1] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on Graphs: Faithful and interpretable large language model reasoning," in ICLR 2024: The Twelfth International Conference on Learning Representations, 2024.

[2] R. Zhao, F. Zhao, L. Wang, X. Wang, and G. Xu, "KG-CoT: Chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering," in Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, 2024, pp. 6642–6650.

[3] J. Sun et al., "Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph," arXiv preprint arXiv:2307.07697, 2023.

[4] J. Jiang et al., "Structgpt: A general framework for large language model to reason over structured data," arXiv preprint arXiv:2305.09645, 2023.

[5] Y. Wu et al., "Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering," arXiv preprint arXiv:2309.11206, 2023.

[6] T. Guo et al., "KnowledgeNavigator: Leveraging large language models for enhanced reasoning over knowledge graph," Complex & Intelligent Systems, vol. 10, no. 5, pp. 7063–7076, 2024.

[7] X. Wang et al., "KEPLER: A unified model for knowledge embedding and pre-trained language representation,"Transactions of the Association for Computational Linguistics, vol. 9, pp. 176–194, 2021.

[8] Z. Zhang et al., "ERNIE: Enhanced language representation with informative entities," arXiv preprint arXiv:1905.07129, 2019.

[9] W. Liu et al., "K-BERT: Enabling language representation with knowledge graph," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 3, pp. 2901–2908, 2020.

[10] Y. Liu et al., "KG-BART: Knowledge graph-augmented bart for generative commonsense reasoning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 7, pp. 6418–6425, 2021.

[11] J. Ma et al., "Debate on graph: A flexible and reliable reasoning framework for large language models," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 23, pp. 24768–24776, 2025.

[12] X. He et al., "G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering," Advances in Neural Information Processing Systems, vol. 37, pp. 132876–132907, 2024.

[13] H. Han et al., "RAG vs. GraphRAG: A systematic evaluation and key insights," arXiv preprint arXiv:2502.11371, 2025.

[14] T. Ao et al., "LightPROF: A lightweight reasoning framework for large language model on knowledge graph," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 22, pp. 23424–23432, 2025.

[15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., "Chain-of-thought prompting elicits reasoning in large language models," Adv. Neural Inf. Process. Syst., vol. 35, pp. 24824–24837, 2022.

[16] S. Yao et al., "Tree of Thoughts: Deliberate problem solving with large language models," Advances in Neural Information Processing Systems, vol. 36, 2024.

[17] M. Besta et al., "Graph of Thoughts: Solving elaborate problems with large language models," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 16, pp. 17682–17690, 2024.

[18] Y. Tan, H. Lv, P. Zhan, S. Wang, and C. Yang, "Graph-oriented instruction tuning of large language models for generic graph mining," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025.

[19] B. Jimenez Gutierrez et al., "HippoRAG: Neurobiologically inspired long-term memory for large language models," Advances in Neural Information Processing Systems, vol. 37, pp. 59532–59569, 2024.

[20] S. Jeong et al., "Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity," arXiv preprint arXiv:2403.14403, 2024.

[21] W.-T. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh, "The value of semantic parse labeling for knowledge base question answering," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 201–206, 2016.

[22] A. Talmor and J. Berant, "The web as a knowledge-base for answering complex questions," arXiv preprint arXiv:1803.06643, 2018.

[23] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1533–1544, 2013.

[24] Y. Gu et al., "Beyond IID: Three levels of generalization for question answering on knowledge bases," in Proceedings of the Web Conference, pp. 3477–3488, 2021.

[25] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," inProceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1247–1250, 2008.

[26] Y. Lan and J. Jiang, "Query graph generation for answering multi-hop complex questions from knowledge bases," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 969–974, 2020.

[27] J. Jiang et al., "UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph," arXiv preprint arXiv:2212.00959, 2022.

[28] A. Miller, "Key-value memory networks for directly reading documents," arXiv preprint arXiv:1606.03126, 2016.

[29] H. Sun et al., "Open domain question answering using early fusion of knowledge bases and text," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 4231–4242, 2018.

[30] H. Sun, T. Bedrax-Weiss, and W. Cohen, "PullNet: Open domain question answering with iterative retrieval on knowledge bases and text," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 2380–2390, 2019.

[31] A. Saxena, A. Tripathi, and P. Talukdar, "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, pp. 4498–4507, 2020.

[32] G. He, Y. Lan, J. Jiang, W. X. Zhao, and J.-R. Wen, "Improving multi-hop knowledge base question answering by learning intermediate supervision signals," in Proceedings of the ACM International Conference on Web Search and Data Mining, pp. 553–561, 2021.

[33] J. Shi et al., "TransferNet: An effective and transparent framework for multi-hop question answering over relation graph," arXiv preprint arXiv:2104.07302, 2021.

[34] J. Jiang et al., "UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph," *arXiv preprint arXiv:2212.00959*, 2022.

[35] J. Kim et al., "KG-GPT: A general framework for reasoning on knowledge graphs using large language models," arXiv preprint arXiv:2310.11220, 2023.

[36] T. Li et al., "Few-shot in-context learning for knowledge base question answering," arXiv preprint arXiv:2305.01750, 2023.