Human Cognition Inspired RAG with Knowledge Graph for Complex Problem Solving

Yao Cheng[1], **Yibo Zhao**[1], **Jiapeng Zhu**[1], **Yao Liu**[1*], **Xing Sun**[2], **Xiang Li**[1*]

[1]School of Data Science and Engineering, East China Normal University, China
[2]Tencent Youtu Lab, China

Abstract

Large Language Models (LLMs) have demonstrated significant potential across various domains. However, they often struggle with integrating external knowledge and performing complex reasoning, leading to hallucinations and unreliable outputs. Retrieval Augmented Generation (RAG) has emerged as a promising paradigm to mitigate these issues by incorporating external knowledge. Yet, conventional RAG approaches, especially those based on vector similarity, fail to effectively capture relational dependencies and support multi-step reasoning. In this work, we propose CogGRAG, a human cognition-inspired, graph-based RAG framework designed for Knowledge Graph Question Answering (KGQA). CogGRAG models the reasoning process as a tree-structured mind map that decomposes the original problem into interrelated subproblems and explicitly encodes their semantic relationships. This structure not only provides a global view to guide subsequent retrieval and reasoning but also enables self-consistent verification across reasoning paths. The framework operates in three stages: (1) top-down problem decomposition via mind map construction, (2) structured retrieval of both local and global knowledge from external Knowledge Graphs (KGs), and (3) bottom-up reasoning with dual-process self-verification. Unlike previous tree-based decomposition methods such as MindMap or Graph-CoT, CogGRAG unifies problem decomposition, knowledge retrieval, and reasoning under a single graph-structured cognitive framework, allowing early integration of relational knowledge and adaptive verification. Extensive experiments demonstrate that CogGRAG achieves superior accuracy and reliability compared to existing methods.

Code — https://github.com/cy623/RAG.git
Extended version — https://arxiv.org/abs/2503.06567

Introduction

As a foundational technology for artificial general intelligence (AGI), large language models (LLMs) have achieved remarkable success in practical applications, demonstrating transformative potential across a wide range of domains (Touvron et al. 2023; AI@Meta 2024; Yang et al. 2024). Their ability to process, generate, and reason with natural language has enabled significant advancements in areas such as machine translation (Zhu et al. 2023), text summarization (Basyal and Sanghvi 2023), and question answering (Pan et al. 2023). Despite their impressive performance, LLMs still face significant limitations in knowledge integration beyond their pre-trained data boundaries. These limitations often lead to the generation of plausible but factually incorrect responses, a phenomenon commonly referred to as *hallucinations*, which undermines the reliability of LLMs in critical applications.

To mitigate hallucinations, Retrieval Augmented Generation (RAG) (Niu et al. 2023; Gao et al. 2023; Edge et al. 2024; Xin et al. 2024; Chu et al. 2024; Cao et al. 2023; Wu et al. 2025; Zhao et al. 2025) has emerged as a promising paradigm, significantly improving the accuracy and reliability of LLM-generated contents through the integration of external knowledge. However, while RAG successfully mitigates certain aspects of hallucination, it still exhibits inherent limitations in processing complex relational information. As shown in Figure 1 (a), the core limitation of standard RAG systems lies in their reliance on vector-based similarity matching, which processes knowledge segments as isolated units without capturing their contextual interdependencies or semantic relationships (Jin et al. 2024b). Consequently, traditional RAG implementations are inadequate for supporting advanced reasoning capabilities in LLMs, particularly in scenarios requiring complex problem-solving, multi-step inference, or sophisticated knowledge integration.

Recently, graph-based RAG (Edge et al. 2024; Ma et al. 2024; Jin et al. 2024b; Mavromatis and Karypis 2024; Xiong, Bao, and Zhao 2024) has been proposed to address the limitations of conventional RAG systems by incorporating deep structural information from external knowledge sources. These approaches typically utilize KGs to model complex relation patterns within external knowledge bases, employing structured triple representations (<entity, relation, entity>) to integrate fragmented information across multiple document segments. While graph-based RAG has shown promising results in mitigating hallucination and improving factual accuracy, several challenges remain unresolved:

• **Lack of holistic reasoning structures.** Complex problems cannot be resolved through simple queries; they typically require multi-step reasoning to derive the final answer. Existing methods often adopt iterative or sequential inference pipelines that are prone to cascading errors due to the absence

*Corresponding author.

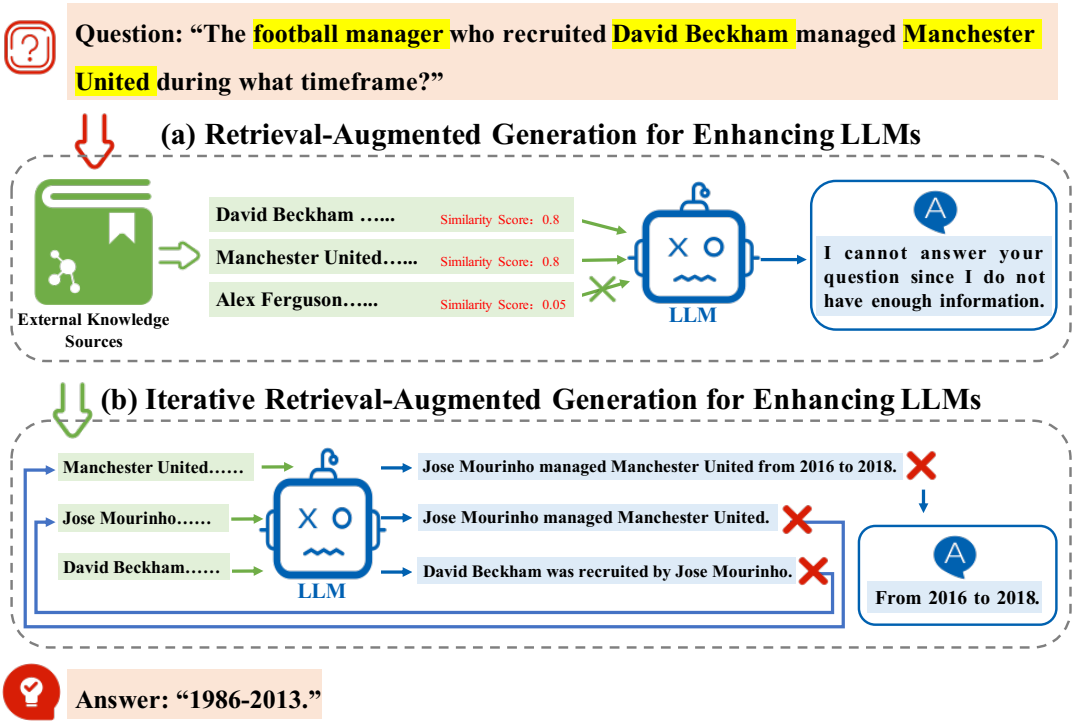

Figure 1: Representative workflow of two Retrieval-Augmented Generation paradigms for enhancing LLMs.

of global reasoning plans (Jin et al. 2024b; Mavromatis and Karypis 2024). As shown in Figure 1 (b), each step in the iterative process relies on the result of the previous step, indicating that errors occurred at previous steps can propagate to subsequent steps.

• **Absence of verification mechanisms.** Despite the integration of external knowledge sources, LLMs remain prone to generating inaccurate or fabricated responses when confronted with retrieval errors or insufficient knowledge coverage. Most approaches do not incorporate explicit self-verification, making them vulnerable to retrieval errors or spurious reasoning paths.

To address these challenges, we propose CogGRAG, a **Cog**nition inspired **G**raph **RAG** framework, designed to enhance the complex problem-solving capabilities of LLMs in Knowledge Graph Question Answering (KGQA) tasks. CogGRAG is motivated by dual-process theories in human cognition, where problem solving involves both structured decomposition and iterative reflection. Our key contributions are as follows:

• **Human-inspired decomposition.** CogGRAG introduces a top-down question decomposition strategy that constructs a tree-structured mind map, explicitly modeling semantic relationships among subproblems. This representation enables the system to capture dependencies, causal links, and reasoning order between subcomponents, providing a global structure that guides subsequent retrieval and reasoning.

• **Hierarchical structured retrieval.** CogGRAG performs both local-level (entity/triple-based) and global-level (subgraph) retrieval based on the mind map. This dual-level retrieval strategy ensures precise and context-rich knowledge grounding.

• **Self-verifying reasoning.** Inspired by cognitive self-reflection, CogGRAG employs a dual-LLM verification mechanism. The reasoning LLM generates answers, which are then reviewed by a separate verifier LLM to detect and revise erroneous outputs. This design reduces hallucinations and improves output faithfulness.

We validate CogGRAG across three general KGQA benchmarks (HotpotQA, CWQ, WebQSP) and one domain-specific KGQA dataset (GRBench). Empirical results demonstrate that CogGRAG consistently outperforms strong baselines in both accuracy and hallucination suppression, under multiple LLM backbones.

Related Work

Reasoning with LLM Prompting. Recent advancements in prompt engineering have demonstrated that state-of-the-art prompting techniques can significantly enhance the reasoning capabilities of LLMs on complex problems (Wei et al. 2022; Yao et al. 2024; Besta et al. 2024). Chain of Thought (CoT (Wei et al. 2022) explores how generating a chain of thought—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. Tree of Thoughts (ToT) (Yao et al. 2024) introduces a new framework for language model inference that generalizes the popular Chain of Thought approach to prompting language models, enabling exploration of coherent units of text (thoughts) as intermediate steps toward problem-solving. The Graph of Thoughts

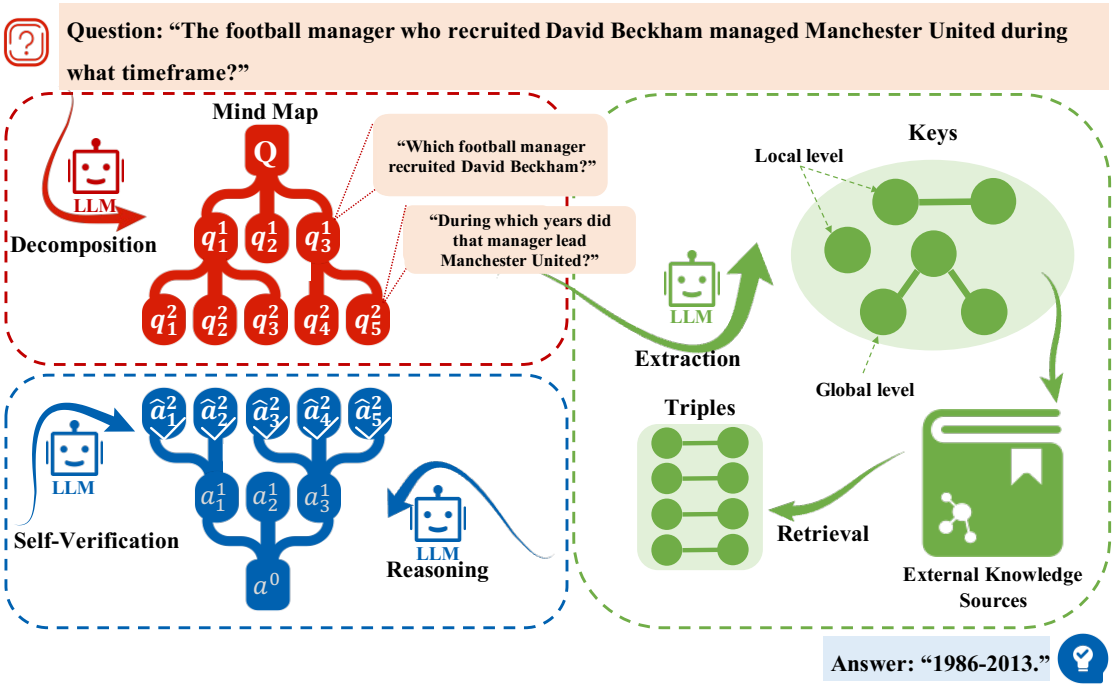

Figure 2: The overall process of CogGRAG. Given a target question Q, CogGRAG first prompts the LLM to decompose it into a hierarchy of sub-problems in a top-down manner, constructing a structured mind map. Subsequently, CogGRAG prompts the LLM to extract both local level (entities and triples) and global level (subgraphs) key information from these questions. Finally, CogGRAG guides the LLM to perform bottom-up reasoning and verification based on the mind map and the retrieved knowledge, until the final answer is derived.

(GoT) (Besta et al. 2024) models the information generated by a LLM as an arbitrary graph, enabling LLM reasoning to more closely resemble human thinking or brain mechanisms. However, these methods remain constrained by the limitations of the model's pre-trained knowledge base and are unable to address hallucination issues stemming from the lack of access to external, up-to-date information.

Knowledge Graph Augmented LLM. KGs (Vrandečić and Krötzsch 2014) offer distinct advantages in enhancing LLMs with structured external knowledge. Early graph-based RAG methods (Edge et al. 2024; He et al. 2024; Hu et al. 2024; Wu et al. 2023; Jin et al. 2024a) demonstrated this potential by retrieving and integrating structured, relevant information from external sources, enabling LLMs to achieve superior performance on knowledge-intensive tasks. However, these methods exhibit notable limitations when applied to complex problems. Recent advancements have introduced methods like chain-of-thought prompting to enhance LLMs' reasoning on complex problems (Sun et al. 2023; Ma et al. 2024; Jin et al. 2024b; Chen et al. 2024; Luo et al. 2025, 2024b; Li, Miao, and Li 2024). Think-on-Graph (Sun et al. 2023) introduces a new approach that enables the LLM to iteratively execute beam search on the KGs, discover the most promising reasoning paths, and return the most likely reasoning results. ToG-2 (Ma et al. 2024) achieves deep and faithful reasoning in LLMs through an iterative knowledge retrieval process that involves collaboration between contexts and the KGs. GRAPH-COT (Jin et al. 2024b) propose a simple and effective framework to augment LLMs with graphs by encouraging LLMs to reason on the graph iteratively. However, iterative reasoning processes are prone to error propagation, as errors cannot be corrected once introduced. This leads to error accumulation and makes it difficult to reason holistically or revise previous steps. In contrast, our CogGRAG framework constructs a complete reasoning plan in advance through top-down decomposition, forming a mind map that globally guides retrieval and reasoning. None of the aforementioned approaches incorporate a self-verification phase. Once an incorrect inference is made, it propagates without correction. CogGRAG addresses this by introducing a dual-LLM self-verification module inspired by dual-process cognitive theory, allowing the system to detect and revise incorrect outputs during the reasoning phase.

Methodology

In this section, we introduce CogGRAG a human cognition-inspired graph-based retrieval-augmented generation framework designed to enhance complex reasoning in KGQA tasks. CogGRAG simulates human cognitive strategies by decomposing complex questions into structured sub-problems, retrieving relevant knowledge in a hierarchical manner, and verifying reasoning results through dual-process reflection. The overall framework consists of three stages: (1) *Decomposition*, (2) *Structured Retrieval*, and (3) *Reasoning with*

Self-Verification. The workflow is illustrated in Figure 2.

Problem Formulation

Given a natural language question Q, the goal is to generate a correct answer A using an LLM p_θ augmented with a KG $\mathcal{G}$. CogGRAG aim to design a graph-based RAG framework with an LLM backbone p_θ to enhance the complex problem-solving capabilities and generate the answer A.

Top-Down Decomposition

Inspired by human cognition in problem-solving, we simulate a top-down analytical process by decomposing the original question into a hierarchy of semantically coherent sub-questions. This decomposition process produces a *reasoning mind map*, which serves as an explicit structure to guide subsequent retrieval and reasoning steps. In this mind map, each node represents a sub-question, and edges capture the logical dependencies among them. Specifically, CogGRAG decomposes the original question Q into a hierarchical structure forming a mind map $\mathcal{M}$. Each node in $\mathcal{M}$ is defined as a tuple $m = (q, t, s)$, where q denotes the sub-question, t indicates its depth level in the tree, and $s \in \{\texttt{Continue}, \texttt{End}\}$ specifies whether further decomposition is required.

The decomposition proceeds recursively using the LLM:

$$\{(q_j^{t+1}, s_j^{t+1})\}_{j=1}^{N} = \texttt{Decompose}(q^t, p_\theta, \texttt{prompt}_{dec}), \tag{1}$$

where N is determined adaptively by LLM. This process continues until all leaves in $\mathcal{M}$ are labeled with $\texttt{End}$, representing atomic questions.

This hierarchical planning mechanism allows CogGRAG to surface latent semantic dependencies that may be overlooked by traditional matching-based RAG methods. For instance, consider the query: "*The football manager who recruited David Beckham managed Manchester United during what timeframe?*" A conventional retrieval pipeline may fail to resolve the intermediate entity "Alex Ferguson," as it is not directly mentioned in the input question. In contrast, CogGRAG first decomposes the query into "Who recruited David Beckham?" and "When did that manager coach Manchester United?", thereby enabling the system to recover critical missing entities and construct a more accurate reasoning path.

By performing top-down question decomposition before any retrieval or reasoning, CogGRAG explicitly separates planning from execution. This design enables holistic reasoning over the entire problem space, mitigates error propagation, and sets the foundation for effective downstream processing.

Structured Knowledge Retrieval

Once the mind map $\mathcal{M}$ has been constructed through top-down decomposition, CogGRAG enters retrieval stage. This phase is responsible for identifying the external knowledge required to support reasoning across all sub-questions. In contrast to prior iterative methods (Jin et al. 2024b) that retrieve relevant content at each reasoning step, CogGRAG performs a **one-pass, globally informed retrieval** guided by the full structure of $\mathcal{M}$. This design not only enhances retrieval completeness and contextual coherence, but also avoids error accumulation caused by sequential retrieval failures.

To retrieve relevant information from the KG $\mathcal{G}$, CogGRAG extracts two levels of key information from $\mathcal{M}$. **Local-level information** captures entities and fine-grained relational facts associated with individual sub-questions. These include entity mentions, entity-relation pairs, and triples. **Global-level information** reflects semantic dependencies across multiple sub-questions and is expressed as interconnected subgraphs. These subgraphs encode logical chains or joint constraints necessary to resolve the overall problem. For example, given the decomposed question "*Which football manager recruited David Beckham?*", we can extract local-level information such as the entity (David Beckham) and the triple (manager, recruited, David Beckham). By additionally considering another decomposed question—such as "*During which years did that manager lead Manchester United?*"—we can derive global level information, which is represented in the form of a subgraph [(manager, recruited, David Beckham), (manager, manage, Manchester United)]. The extraction process is performed using the LLM:

$$\mathcal{K} = \texttt{Extract}(\mathcal{M}, p_\theta, \texttt{prompt}_{ext}), \tag{2}$$

where $\mathcal{K}$ denotes the set of all extracted keys, including entity nodes, triples, and subgraph segments. This design ensures that both isolated facts and relational contexts are captured prior to KGs querying. These keys guide graph-based retrieval. For each entity e in the extracted key set $\mathcal{K}$, we expand to its neighborhood in $\mathcal{G}$, yielding a candidate triple set $\tilde{\mathcal{T}}$. A semantic similarity filter is applied to retain only relevant triples:

$$\mathcal{T} = \{\tau \in \tilde{\mathcal{T}}, k \in \mathcal{K} \mid \texttt{sim}(\tau, k) > \varepsilon\}, \tag{3}$$

where $\texttt{sim}(\cdot)$ is cosine similarity, ε is a threshold, τ is the triple in candidate triple set $\tilde{\mathcal{T}}$ and k is the triple in the extracted key set $\mathcal{K}$. The final retrieved triple set $\mathcal{T}$ serves as the input evidence pool for the reasoning module. By leveraging both local and global context from $\mathcal{M}$, CogGRAG ensures that $\mathcal{T}$ is semantically rich, structurally connected, and targeted toward the entire reasoning path.

Reasoning with Self-Verification

The final stage of CogGRAG emulates human problem-solving behavior, where conclusions are not only derived through reasoning but also reviewed through self-reflection (Frederick 2005). To this end, CogGRAG employs a **dual-process reasoning** architecture, composed of two distinct LLMs:

- $\textbf{LLM}_{\textbf{res}}$: responsible for answer generation via bottom-up reasoning over the mind map $\mathcal{M}$;
- $\textbf{LLM}_{\textbf{ver}}$: responsible for evaluating the validity of generated answers based on context and prior reasoning history.

This dual-agent setup is inspired by dual-process theories in cognitive psychology (Vaisey 2009), where System 1 performs intuitive reasoning and System 2 monitors and corrects errors.

Bottom-Up Reasoning. Given the final retrieved triple set $\mathcal{T}$ and the structured mind map $\mathcal{M}$, CogGRAG performs reasoning in a bottom-up fashion: sub-questions at the lowest level are answered first, and their verified results are recursively used to resolve higher-level nodes.

Let $\mathcal{M}$ denote the full mind map. We define $\hat{\mathcal{M}} \subset \mathcal{M}$ as the subset of sub-questions that have been answered and verified. Each element in $\hat{\mathcal{M}}$ is a tuple $(q, \hat{a})$, where q is a sub-question and $\hat{a}$ is its verified answer. Then the reasoning LLM generates a candidate answer:

$$a^t = \texttt{LLM}_{\text{res}}(\mathcal{T}, q^t, \hat{\mathcal{M}}, \texttt{prompt}_{res}). \tag{4}$$

Self-Verification and Re-thinking. The candidate answer a^t is passed to the verifier LLM along with the current reasoning path. The verifier assesses consistency, factual grounding, and logical coherence. If the answer fails validation, a rethinked response $\hat{a}^t$ is regenerated:

$$\hat{a}^t = \begin{cases} \texttt{LLM}_{\text{res}}(\mathcal{T}, q^t, \hat{\mathcal{M}}, \texttt{prompt}_{rethink}) & \text{if } \delta^t = \texttt{False}, \\ a^t & \text{otherwise.} \end{cases} \tag{5}$$

Here, δ^t is a boolean indicator defined as:

$$\delta^t = \texttt{LLM}_{\text{ver}}(q^t, a^t, \hat{\mathcal{M}}, \texttt{prompt}_{ver}), \tag{6}$$

which returns `True` if the verification module accepts a^t, and `False` otherwise. In addition to verification, CogGRAG incorporates a selective abstention mechanism. When the reasoning LLM cannot confidently produce a grounded answer based on $\mathcal{T}$, it is explicitly prompted to respond with "`I don't know`" rather than hallucinating. The final answer A to the original question Q is obtained by recursively aggregating verified answers $\hat{a}^t$ from the leaf nodes up to the root node q^0, such that $A = \hat{a}^0$.

This bottom-up reasoning pipeline, governed by a globally constructed mind map and protected by verification layers, enables CogGRAG to perform accurate and reliable complex reasoning over structured knowledge. All prompt templates used in CogGRAG can be found in the Appendix.

Experiments

Experimental Settings

Datasets and evaluation metrics. In order to test CogGRAG's complex problem-solving capabilities on KGQA tasks, we evaluate CogGRAG on three widely used complex KGQA benchmarks: (1) **HotpotQA** (Yang et al. 2018), (2) **WebQSP** (Yih et al. 2016), (3) **CWQ** (Talmor and Berant 2018). Following previous work (Li et al. 2023), full Wikidata (Vrandečić and Krötzsch 2014) is used as structured knowledge sources for all of these datasets. Considering that Wikidata is commonly used for training LLMs, there is a need for a domain-specific QA dataset that is not exposed during the pretraining process of LLMs in order to better evaluate the performance. Thus, we also test CogGRAG on a recently released domain-specific dataset **GRBENCH** (Jin et al. 2024b). All methods need to interact with domain-specific graphs containing rich knowledge to solve the problem in this dataset. The statistics and details of these datasets can be found in Appendix. For all datasets, we use three evaluation metrics: (1) **Exact match (EM):** measures whether the predicted answer or result matches the target answer exactly. (2) **Rouge-L (RL):** measures the longest common subsequence of words between the responses and the ground truth answers. (3) **F1 Score (F1):** computes the harmonic mean of precision and recall between the predicted and gold answers, capturing both completeness and exactness of overlapping tokens.

Baselines. In our main results, we compare CogGRAG with three types state-of-the-art methods: (1) **LLM-only** methods without external knowledge, including direct reasoning and CoT (Wei et al. 2022) by LLM. (2) **LLM+KG** methods integrate relevant knowledge retrieved from the KG into the LLM to assist in reasoning, including direct reasoning and CoT by LLM. (3) **Graph-based RAG methods** allow KGs and LLMs to work in tandem, complementing each other's capabilities at each step of graph reasoning, including Mindmap (Wen, Wang, and Sun 2023), Think-on-graph (Sun et al. 2023), Graph-CoT (Jin et al. 2024b), RoG (Luo et al. 2024a), GoG (Xu et al. 2024). Due to the space limitation, we move details on datasets, baselines and experimental setup to Appendix.

Experimental Setup. We conduct experiments with four LLM backbones: LLaMA2-13B (Touvron et al. 2023), LLaMA3-8B (AI@Meta 2024), Qwen2.5-7B (Yang et al. 2024) and Qwen2.5-32B (Hui et al. 2024). For all LLMs, we load the checkpoints from huggingface[1] and use the models directly without fine-tuning. We implemented CogGRAG and conducted the experiments with one A800 GPU. Consistent with the Think-on-graph settings, we set the temperature parameter to 0.4 during exploration and 0 during reasoning. The threshold ϵ in the retrieval step is set to 0.7.

Main Results

We perform experiments to verify the effectiveness of our framework CogGRAG, and report the results in Table 1. We use Rouge-L (RL), Exact match (EM) and F1 Score (F1) as metric for all three datasets. The backbone model for all the methods is LLaMA2-13B. From the table, the following observations can be derived: (1) CogGRAG achieves the best results in most cases except on WebSQP. Since the dataset is widely used, we attribute the reason to be data leakage. (2) Compared to methods that incorporate external knowledge, the LLM-only approach demonstrates significantly inferior performance. This performance gap arises from the lack of necessary knowledge in LLMs for reasoning tasks, highlighting the critical role of external knowledge integration in enhancing the reasoning capabilities of LLMs. (3) Graph-based RAG methods demonstrate superior performance compared to LLM+KG approaches. This performance advantage is particularly evident in complex problems, where not only external knowledge integration but also involving "thinking procedure" is essential. These methods synergize LLMs with KGs to enhance performance by retrieving and reasoning over the KGs, thereby generating the most probable inference outcomes through "thinking" on the KGs.

We attribute CogGRAG 's outstanding effectiveness in most cases primarily to its ability to decompose complex

[1] https://huggingface.co

Type	Method	HotpotQA			CWQ			WebQSP		
		RL	EM	F1	RL	EM	F1	RL	EM	F1
LLM-only	Direct	19.1%	17.3%	18.7%	31.4%	28.8%	31.7%	51.4%	47.9%	53.5%
	CoT	23.3%	20.8%	22.1%	35.1%	32.7%	33.5%	55.2%	51.6%	55.3%
LLM+KG	Direct+KG	27.5%	23.7%	27.6%	39.7%	35.1%	38.3%	52.5%	49.3%	52.1%
	CoT+KG	28.7%	25.4%	26.9%	42.2%	37.6%	40.8%	52.8%	48.1%	50.5%
Graph-based RAG	ToG	29.3%	26.4%	29.6%	49.1%	46.1%	47.7%	54.6%	57.4%	56.1%
	MindMap	27.9%	25.6%	27.8%	46.7%	43.7%	44.1%	56.6%	53.1%	58.3%
	RoG	30.7%	28.1%	30.4%	55.3%	51.8%	54.7%	65.2%	**62.8%**	**67.2%**
	GoG	31.5%	30.1%	31.1%	55.7%	52.4%	54.8%	**65.5%**	59.1%	63.6%
Ours	**CogGRAG**	**34.4%**	**30.7%**	**35.5%**	**56.3%**	**53.4%**	**55.8%**	59.8%	56.1%	58.9%

Table 1: Overall results of our CogGRAG on three KBQA datasets. The best score on each dataset is highlighted.

Type	Method	HotpotQA			CWQ			WebQSP		
		RL	EM	F1	RL	EM	F1	RL	EM	F1
LLM-only	Qwen2.5-7B	15.3%	15.0%	15.8%	25.4%	24.1%	24.5%	46.7%	45.3%	45.5%
	LLaMA3-8B	17.5%	14.9%	16.2%	30.3%	27.5%	29.0%	50.4%	45.1%	48.3%
	LLaMA2-13B	19.1%	17.3%	18.7%	31.4%	28.8%	31.7%	51.4%	47.9%	53.5%
	Qwen2.5-32B	28.7%	27.4%	28.5%	55.1%	50.3%	54.2%	68.4%	60.5%	65.1%
LLM+KG	Qwen2.5-7B+KG	24.2%	15.6%	21.4%	33.8%	32.1%	34.9%	46.7%	45.3%	46.1%
	LLaMA3-8B+KG	25.9%	21.4%	23.6%	40.6%	35.3%	39.1%	53.6%	49.1%	52.3%
	LLaMA2-13B+KG	27.5%	23.7%	27.6%	39.7%	35.1%	38.3%	52.5%	49.3%	52.1%
	Qwen2.5-32B+KG	35.6%	32.8%	34.9%	58.3%	54.7%	57.6%	70.2%	65.1%	68.4%
Graph-Based RAG	CogGRAG w/ Qwen2.5-7B	28.4%	27.1%	28.2%	50.5%	45.7%	48.9%	53.2%	51.6%	55.0%
	CogGRAG w/ LLaMA3-8B	32.1%	27.2%	31.0%	53.5%	48.4%	52.6%	57.2%	55.3%	55.4%
	CogGRAG w/ LLaMA2-13B	34.4%	30.7%	35.5%	56.3%	53.4%	55.8%	59.8%	56.1%	58.9%
	CogGRAG w/ Qwen2.5-32B	40.5%	37.1%	40.2%	66.5%	62.7%	65.4%	74.1%	68.3%	73.0%

Table 2: Overall results of our CogGRAG with different backbone models on three KBQA datasets.

problems and construct a structured mind map prior to retrieval. This approach enables the construction of a comprehensive reasoning pathway, facilitating more precise and targeted retrieval of relevant information. Furthermore, CogGRAG incorporates a self-verification mechanism during the reasoning phase, further enhancing the accuracy and reliability of the final results. Together, these designs collectively enhance LLMs' ability to tackle complex problems.

Performance with different backbone models

We evaluate how different backbone models affect its performance on three datasets HotpotQA, CWQ and WebQSP, and report the results in Table 2. We conduct experiments with three LLM backbones LLaMA2-13B, LLaMA3-8B, Qwen2.5-7B and Qwen2.5-32B. For all LLMs, we load the checkpoints from huggingface and use the models directly without fine-tuning. From the table, we can observe the following key findings: (1) CogGRAG achieves the best results across all backbone models compared to the baseline approaches, demonstrating the robustness and stability of our method. (2) The performance of our method improves consistently as the model scale increases, reflecting enhanced reasoning capabilities. This trend suggests that our approach has significant potential for further exploration with larger-scale models, indicating promising scalability and adaptability.

Performance on domain-specific KG

Given the risk of data leakage due to Wikidata being used as pretraining corpora for LLMs, we further evaluate the performance of CogGRAG on a recently released domain-specific dataset GRBENCH (Jin et al. 2024b) and report the results in Table 3. This dataset requires all questions to interact with a domain-specific KG. We use Rouge-L (RL), Exact match (EM) and F1 Score (F1) as metrics for this dataset. The backbone model for all the methods is LLaMA2-13B (Touvron et al. 2023). The table reveals the following observations: (1) CogGRAG continues to outperform in most cases. This demonstrates that our method consistently achieves stable and reliable results even on domain-specific KGs. (2) Both CogGRAG and Graph-CoT outperform LLaMA2-13B by more than 20%, which can be ascribed to the fact that LLMs are typically not trained on domain-specific data. In contrast, graph-based RAG methods can effectively supplement LLMs with external knowledge, thereby enhancing their reasoning capabilities. This result underscores the effectiveness of the RAG approach in bridging knowledge gaps and improving performance in specialized domains.

Method	E-commerce			Literature			Academic			Healthcare		
	RL	EM	F1	RL	EM	F1	RL	EM	F1	RL	EM	F1
LLaMA2-13B	7.1%	6.8%	6.9%	5.4%	5.1%	5.3%	5.4%	4.7%	5.1%	4.3%	3.1%	3.6%
Graph-CoT	26.4%	24.0%	25.3%	26.7%	23.3%	24.9%	19.3%	14.8%	16.9%	**28.1%**	25.2%	**26.7%**
CogGRAG	**30.2%**	**28.7%**	**29.5%**	**32.4%**	**30.1%**	**31.3%**	**23.6%**	**21.5%**	**22.7%**	27.4%	**25.6%**	26.2%

Table 3: Overall results of our CogGRAG on GRBENCH dataset. We highlight the best score on each dataset in bold.

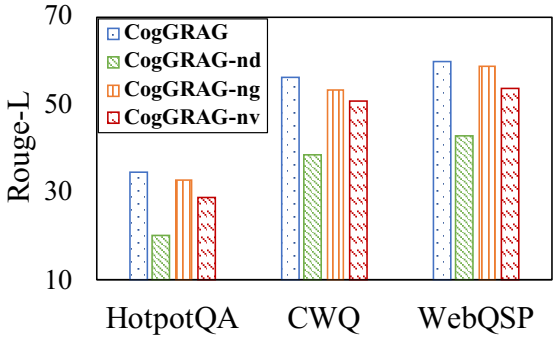

Figure 3: Ablation study on the main components of CogGRAG.

Method	Correct (↑)	Missing (↑)	Hallucination (↓)
LLaMA2-13B	19.1%	25.7%	55.2%
ToG	29.3%	20.2%	50.5%
MindMap	27.9%	22.4%	49.7%
CogGRAG	**34.4%**	**40.6%**	**25.0%**

Table 4: Overall results of our CogGRAG on GRBENCH dataset. We highlight the best score on each dataset in bold.

Ablation Study

The ablation study is conducted to understand the importance of main components of CogGRAG. We select HotpotQA, CWQ and WebQSP as three representative datasets. First, we remove the problem decomposition module, directly extracting information for the target question, and referred to this variant as CogGRAG-nd (**n**o **d**ecomposition). Next, we eliminate the global-level retrieval phase, naming this variant CogGRAG-ng (**n**o **g**lobal level). Finally, we remove the self-verification mechanism during the reasoning stage, designating this variant as CogGRAG-nv (**n**o **v**erification). These experiments aim to systematically assess the impact of each component on the overall performance of the framework. We compare CogGRAG with these three variants, and the results are presented in Figure 3. Our findings show that CogGRAG outperforms all the variants on the three datasets. Furthermore, the performance gap between CogGRAG and CogGRAG-nd highlights the importance of decomposition for complex problem reasoning in KGQA tasks.

Hallucination and Missing Evaluation

In the reasoning and self-verification phase of CogGRAG, we prompt the LLM to respond with "I don't know" when encountering questions with insufficient or incomplete relevant information during the reasoning process. This approach is designed to mitigate the hallucination issue commonly observed in LLMs. To evaluate the effectiveness of this strategy, we test the model on the HotpotQA dataset, with results reported in Table 4. We categorize the responses into three types: "Correct" for accurate answers, "Missing" for cases where the model responds with "I don't know," and "Hallucination" for incorrect answers. As shown in the table results, our model demonstrates the ability to refrain from answering questions with insufficient information, significantly reducing the occurrence of hallucinations.

Method	**Academic**	**Amazon**	**Goodreads**	**Disease**
Graph-CoT	22.39	29.04	15.79	32.97
GoG	21.50	32.20	18.80	37.50
CogGRAG	18.24	31.57	16.32	35.64

Table 5: Average inference time per question (s).

Performance in Inference Times

We conduct the verification on four datasets GRBENCH-Academic, GRBENCH-Amazon, GRBENCH-Goodreads, and GRBENCH-Disease. Table 5 presents the average runtime of CogGRAG, Graph-CoT and GoG. From the table, we can see that CogGRAG is on par with Graph-CoT and GoG. Although our reasoning process introduces self-verification, which increases the time cost, CogGRAG does not require iterative repeated reasoning and retrieval. Instead, it retrieves all relevant information in one step based on the mind map, avoiding redundant retrieval, which provides an advantage in large-scale KGs. Moreover, while self-verification adds additional time costs, it also improves the accuracy of reasoning results, as confirmed in the ablation experiments.

Conclusion

In this paper, we proposed CogGRAG, a graph-based RAG framework to enhance LLMs' complex reasoning for KGQA tasks. CogGRAG generates tree-structured mind maps, explicitly encoding semantic relationships among subproblems, which guide both local and global knowledge retrieval. A self-verification module further detects and revises potential errors, forming a dual-phase reasoning paradigm that reduces hallucinations and improves factual consistency. Extensive experiments show that CogGRAG substantially outperforms existing RAG baselines on complex multi-hop QA benchmarks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 42375146 and National Natural Science Foundation of China No. 62202172.

References

AI@Meta. 2024. Llama 3 Model Card.

Basyal, L.; and Sanghvi, M. 2023. Text summarization using large language models: a comparative study of MPT-7B-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT models. *arXiv preprint arXiv:2310.10449*.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17682–17690.

Cao, S.; Zhang, J.; Shi, J.; Lv, X.; Yao, Z.; Tian, Q.; Li, J.; and Hou, L. 2023. Probabilistic tree-of-thought reasoning for answering knowledge-intensive complex questions. *arXiv preprint arXiv:2311.13982*.

Chen, L.; Tong, P.; Jin, Z.; Sun, Y.; Ye, J.; and Xiong, H. 2024. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *Advances in Neural Information Processing Systems*, 37: 37665–37691.

Chu, Z.; Chen, J.; Chen, Q.; Wang, H.; Zhu, K.; Du, X.; Yu, W.; Liu, M.; and Qin, B. 2024. BeamAggR: Beam Aggregation Reasoning over Multi-source Knowledge for Multi-hop Question Answering. *arXiv preprint arXiv:2406.19820*.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; and Larson, J. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Frederick, S. 2005. Cognitive reflection and decision making. *Journal of Economic perspectives*, 19(4): 25–42.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; and Wang, H. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

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. *arXiv preprint arXiv:2402.07630*.

Hu, Y.; Lei, Z.; Zhang, Z.; Pan, B.; Ling, C.; and Zhao, L. 2024. GRAG: Graph Retrieval-Augmented Generation. *arXiv preprint arXiv:2405.16506*.

Hui, B.; Yang, J.; Cui, Z.; Yang, J.; Liu, D.; Zhang, L.; Liu, T.; Zhang, J.; Yu, B.; Dang, K.; et al. 2024. Qwen2. 5-Coder Technical Report. *arXiv preprint arXiv:2409.12186*.

Jin, B.; Liu, G.; Han, C.; Jiang, M.; Ji, H.; and Han, J. 2024a. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*.

Jin, B.; Xie, C.; Zhang, J.; Roy, K. K.; Zhang, Y.; Li, Z.; Li, R.; Tang, X.; Wang, S.; Meng, Y.; et al. 2024b. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*.

Li, M.; Miao, S.; and Li, P. 2024. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*.

Li, X.; Zhao, R.; Chia, Y. K.; Ding, B.; Joty, S.; Poria, S.; and Bing, L. 2023. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. *arXiv preprint arXiv:2305.13269*.

Luo, L.; Li, Y.-F.; Haffari, G.; and Pan, S. 2024a. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *International Conference on Learning Representations*.

Luo, L.; Zhao, Z.; Haffari, G.; Li, Y.-F.; Gong, C.; and Pan, S. 2024b. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *arXiv preprint arXiv:2410.13080*.

Luo, L.; Zhao, Z.; Haffari, G.; Phung, D.; Gong, C.; and Pan, S. 2025. GFM-RAG: graph foundation model for retrieval augmented generation. *arXiv preprint arXiv:2502.01113*.

Ma, S.; Xu, C.; Jiang, X.; Li, M.; Qu, H.; Yang, C.; Mao, J.; and Guo, J. 2024. Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation. arXiv:2407.10805.

Mavromatis, C.; and Karypis, G. 2024. GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning. *arXiv preprint arXiv:2405.20139*.

Niu, C.; Wu, Y.; Zhu, J.; Xu, S.; Shum, K.; Zhong, R.; Song, J.; and Zhang, T. 2023. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *arXiv preprint arXiv:2401.00396*.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2023. Unifying Large Language Models and Knowledge Graphs: A Roadmap, 2023. *arXiv preprint arXiv:2306.08302*.

Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Shum, H.-Y.; and Guo, J. 2023. Think-on-graph: Deep and responsible reasoning of large language model with 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*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaisey, S. 2009. Motivation and justification: A dual-process model of culture in action. *American journal of sociology*, 114(6): 1675–1715.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.

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. *Advances in neural information processing systems*, 35: 24824–24837.

Wen, Y.; Wang, Z.; and Sun, J. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729*.

Wu, J.; Cai, H.; Yan, L.; Sun, H.; Li, X.; Wang, S.; Yin, D.; and Gao, M. 2025. PA-RAG: RAG alignment via multi-perspective preference optimization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 9091–9112.

Wu, Y.; Hu, N.; Bi, S.; Qi, G.; Ren, J.; Xie, A.; and Song, W. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*.

Xin, A.; Liu, J.; Yao, Z.; Lee, Z.; Cao, S.; Hou, L.; and Li, J. 2024. AtomR: Atomic Operator-Empowered Large Language Models for Heterogeneous Knowledge Reasoning. *arXiv preprint arXiv:2411.16495*.

Xiong, G.; Bao, J.; and Zhao, W. 2024. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. *arXiv preprint arXiv:2402.15131*.

Xu, Y.; He, S.; Chen, J.; Wang, Z.; Song, Y.; Tong, H.; Liu, G.; Liu, K.; and Zhao, J. 2024. Generate-on-graph: Treat llm as both agent and kg in incomplete knowledge graph question answering. *arXiv preprint arXiv:2404.14741*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Fan, Z. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

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.

Zhao, Y.; Zhu, J.; Guo, Y.; He, K.; and Li, X. 2025. Eˆ 2GraphRAG: Streamlining Graph-based RAG for High Efficiency and Effectiveness. *arXiv preprint arXiv:2505.24226*.

Zhu, W.; Liu, H.; Dong, Q.; Xu, J.; Huang, S.; Kong, L.; Chen, J.; and Li, L. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.