How to Mitigate Information Loss in Knowledge Graphs for GraphRAG: Leveraging Triple Context Restoration and Query-Driven Feedback

Manzong Huang[1] , **Chenyang Bu**[1*] , **Yi He** [2] , **Xindong Wu**[1*]
[1] Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), School of Computer Science and Information Engineering, Hefei University of Technology, China
[2] School of Computing, Data Sciences, and Physics, College of William and Mary, USA
manzonghuang@mail.hfut.edu.cn, yihe@wm.edu, {chenyangbu, xwu}@hfut.edu.cn

Abstract

Knowledge Graph (KG)-augmented Large Language Models (LLMs) have recently propelled significant advances in complex reasoning tasks, thanks to their broad domain knowledge and contextual awareness. Unfortunately, current methods often assume KGs to be complete, which is impractical given the inherent limitations of KG construction and the potential loss of contextual cues when converting unstructured text into entity-relation triples. In response, this paper proposes the Triple Context Restoration and Query-driven Feedback (TCR-QF) framework, which reconstructs the textual context underlying each triple to mitigate information loss, while dynamically refining the KG structure by iteratively incorporating query-relevant missing knowledge. Experiments on five benchmark question-answering datasets substantiate the effectiveness of TCR-QF in KG and LLM integration, where it achieves a 29.1% improvement in Exact Match and a 15.5% improvement in F1 over its state-of-the-art GraphRAG competitors. The code is publicly available at https://github.com/HFUT-DMiC-Lab/TCR-QF.git.

1 Introduction

Large Language Models (LLMs) augmented with Knowledge Graphs (KGs) have achieved remarkable successes across diverse domains, from social sciences to biomedicine [Pan *et al.*, 2024; Peng *et al.*, 2024; Yang *et al.*, 2024; Soman *et al.*, 2024]. By harmonizing the structured information in KGs and the sophisticated language understanding and processing capabilities of LLMs, such hybrid systems enable more accurate and context-aware reasoning for complex tasks.

Despite these advances, the performance of current KG–LLM integration methods is often hindered by the underlying assumption that the KG is complete. Typical integration strategy involves retrieving relational data from a constructed KG and feed it into LLMs via prompt augmentation [Peng *et al.*, 2024; Sun *et al.*, 2023; Edge *et al.*, 2024; Zhang *et al.*, 2025b; Dehghan *et al.*, 2024], assuming that critical entities and relationships relevant to the query are already captured within the KG. In practice, however, KG construction itself is beset by inherent constraints, where vital contextual information can be discarded in the process of converting unstructured text into structured triples, leading to missing or incomplete relations [Zhu *et al.*, 2024; Zhong *et al.*, 2024]. Such missing information in KGs can significantly degrade the LLM reasoning capabilities.

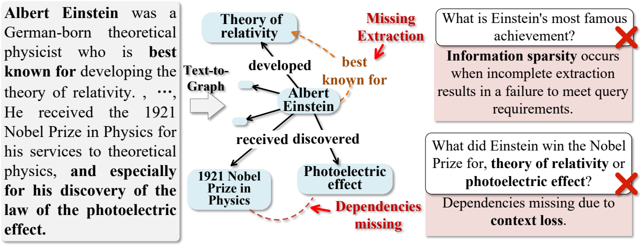
Albert Einstein was a German-born theoretical physicist who is best known for developing the theory of relativity. , ···, He received the 1921 Nobel Prize in Physics for his services to theoretical physics, and especially for his discovery of the law of the photoelectric effect.
Text-to-Graph
Theory of relativity
developed
Albert Einstein
received discovered
1921 Nobel Prize in Physics
Photoelectric effect
best known for
Missing Extraction
Dependencies missing
What is Einstein's most famous achievement?
Information sparsity occurs when incomplete extraction results in a failure to meet query requirements.
What did Einstein win the Nobel Prize for, theory of relativity or photoelectric effect?
Dependencies missing due to context loss.

Figure 1: Illustration of two main factors of information loss in KGs: **information sparsity** and **context loss**. These issues hinder LLMs from accurately answering questions based on KGs.

To wit, Figure 1 illustrates two primary sources of information loss. First, **information sparsity** arises when information extraction falls short, omitting potentially important triples and thus failing to provide sufficient coverage for specific queries [Biswas *et al.*, 2024; Xu *et al.*, 2024b; Li *et al.*, 2023; Zhang and Soh, 2024; Chen *et al.*, 2024a; Sun *et al.*, 2024; Cohen *et al.*, 2023]. This sparsity can be exacerbated by data noise, long-tail entities, and complex relationships, where extraction algorithms often falter. Second, **context loss** occurs when transforming rich yet unstructured text into discrete triples, sacrificing crucial semantic nuances and relational dependencies [Trisedya *et al.*, 2019; Paulheim, 2017; Xu *et al.*, 2024a]. While prior studies attempt to mitigate this issue by refining graph structures or retrieval algorithms [Liang *et al.*, 2024; Chen *et al.*, 2024b; Panda *et al.*, 2024; Munikoti *et al.*, 2023; Cohen *et al.*, 2023], their subgraphs still lack the broader contextual information that is vital for robust reasoning, resulting in suboptimal performance in downstream tasks.

To address these challenges, we propose the **T**riple **C**ontext **R**estoration and **Q**uestion-driven **F**eedback (**TCR-QF**) framework, which aims to restore the missing contextual information and dynamically enrich the KG during the reasoning process. Specifically, our TCR-QF approach presents

[*] Corresponding author.

a *triple context restoration* component that retrieves the original text passages associated with each triple, thereby recapturing the semantic details often lost during KG construction. We further enhance KG coverage through a *query-driven feedback* mechanism, which iteratively identifies missing information relevant to the query and enriches the KG accordingly. These two components together form a synergistic cycle in which contextual fidelity and KG completeness are continuously reinforced, resulting in more accurate and context-aware responses from the LLM. Empirical study on five benchmark question-answering datasets substantiates that TCR-QF significantly outperforms the state-of-the-art GraphRAG methods in both response accuracy and completeness, demonstrating its effectiveness.

Specific Contributions of this paper are as follows:

1) We provide a systematic analysis of the key challenges in KG–LLM integration, highlighting the loss of contextual information and incomplete information extraction during KG construction, both of which hinder an advanced LLM reasoning performance.
2) We propose the TCR-QF framework, which restores the semantic context associated with triples and employs a query-driven feedback mechanism to iteratively enrich the KG, thereby significantly enhancing the LLM reasoning capabilities.
3) Extensive experiments on five benchmark question-answering datasets are carried out, showing that TCR-QF achieves an average 29.1% improvement in Exact Match and a 15.5% improvement in F1 over its GraphRAG competitors. These results validate the merit of restoring contextual information and dynamically updating KGs for effective KG–LLM integration.

2 Related Work

GraphRAG has emerged as a powerful paradigm for integrating knowledge graphs (KGs) with large language models (LLMs) to advance complex reasoning tasks [Pan *et al.*, 2024; Peng *et al.*, 2024; Yang *et al.*, 2024; Zhang *et al.*, 2024; Zhang *et al.*, 2025a]. A widely adopted strategy involves retrieving relevant subgraphs from preconstructed KGs to augment LLMs during inference [Yasunaga *et al.*, 2021; Taunk *et al.*, 2023], with techniques such as extracting hop-k paths around topic entities [Yasunaga *et al.*, 2021] or focusing on the shortest paths relevant to query entities [Delile *et al.*, 2024]. More sophisticated methods optimize subgraph retrieval by assigning edge costs [He *et al.*, 2024] or leverage LLMs themselves to generate new relations or invoke function calls [Kim *et al.*, 2023; Jiang *et al.*, 2023].

While these approaches have demonstrated effectiveness, most remain limited by their dependence on the initial completeness of the KG and often overlook the contextual information lost during KG construction. In reality, KGs are frequently incomplete due to information loss during construction and the difficulties in extracting all relevant triples, especially in noisy or complex scenarios [Biswas *et al.*, 2024; Cohen *et al.*, 2023]. These constraints can hinder the ability of LLMs to formulate coherent and context-rich reasoning paths. Addressing these gaps calls for a more dynamic strategy that restores missing contextual details and continuously refines the KG, ensuring that the retrieved and generated knowledge is both accurate and semantically complete.

However, the lack of essential data negatively impacts the inference results of LLMs. To address this, efforts have been made to enhance KG comprehensiveness through refined indexing methods and innovative graph structures for retrieving both triples and texts [Chen *et al.*, 2024b; Munikoti *et al.*, 2023; Liang *et al.*, 2024; Cohen *et al.*, 2023], as well as using LLMs to improve automated KG construction [Zhang and Soh, 2024; Xu *et al.*, 2024b; Li *et al.*, 2023]. These methods may retrieve texts related to the query without fully meeting its requirements. Additionally, the retrieved subgraphs can result in the loss of crucial information due to the absence of contextual data within triples, which is essential for maintaining semantic integrity. As a result, the constructed KG may lack critical information necessary for accurate reasoning, leading to suboptimal performance in downstream tasks.

The proposed **TCR-QF** framework addresses these limitations by dynamically enriching the KG during the reasoning process. By restoring the original textual context of triples, TCR-QF recovers lost semantic information. Additionally, it employs a query-driven feedback mechanism to identify and fill in missing information relevant to a query, enabling the KG to continuously update. This mutual enhancement between KG and LLM improves reasoning performance and better adapts to task requirements.

3 Proposed Method

Task Definition. Given a set of documents $D = \{d_1, d_2, ..., d_n\}$ and a question Q, the task requires the model to read and reason over multiple relevant documents, extract and aggregate the necessary information, and finally generate the answer A based on the information from the documents.

In this section, we present the **TCR-QF** framework, designed to mitigate the loss of contextual information when building knowledge graphs (KGs) from unstructured text and to dynamically enrich these graphs during the reasoning process. As shown in Figure 2, the framework comprises four key components: **(1) Knowledge Graph Construction**, which builds a unified KG from textual sources; **(2) Subgraph Retrieval**, responsible for extracting task-relevant subgraphs composed of potential reasoning paths; **(3) Triple Context Restoration**, which traces back the original textual context of each triple to recover lost semantic nuances; and **(4) Iterative Reasoning with Query-Driven Feedback**, where an iterative cycle that both generates answers and identifies missing knowledge, thereby refining the KG on-the-fly. Together, these components ensure that contextual details are preserved and the KG remains up-to-date, ultimately enhancing the quality and depth of the system reasoning.

The above steps establishes a synergistic cycle for two-way knowledge enhancement, namely,

Forward Flow: The KG informs the LLM during answer generation, represented as

$$G^{(i)} \longrightarrow G'^{(i)} \longrightarrow A_{\text{LLM}}^{(i)}.$$

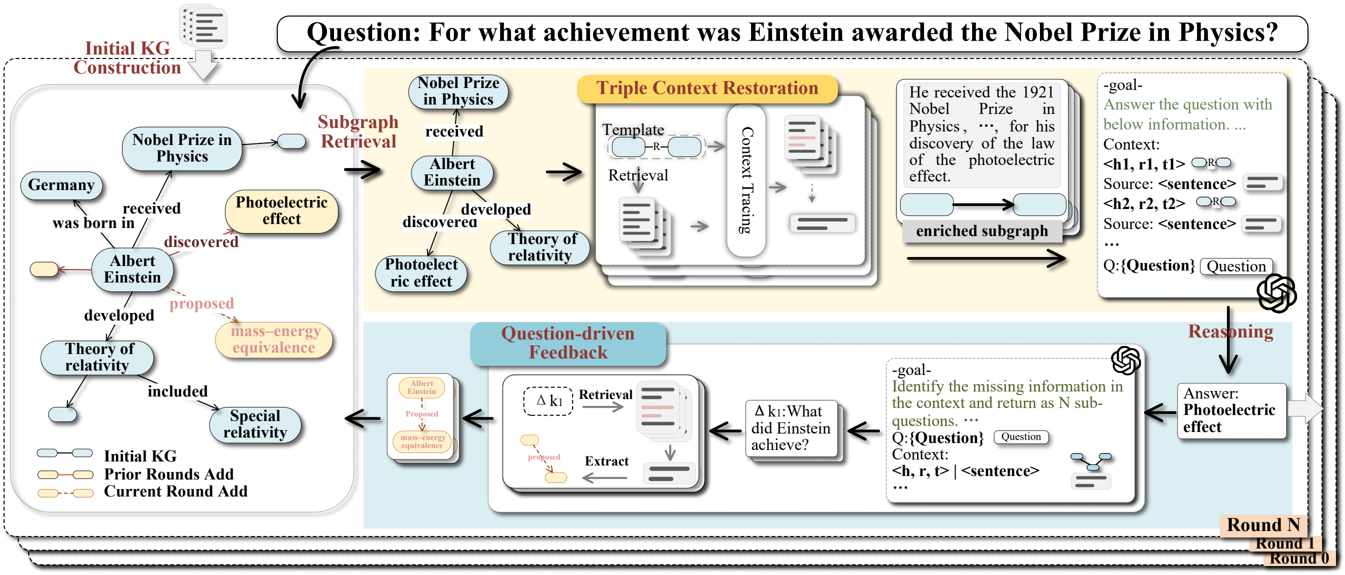

Figure 2: Workflow of TCR-QF. Including a continuously mutual enhancement knowledge flow: (1) Forward Flow: The KG enhances the LLM during answer generation with triple context restoration. (2) Feedback Flow: Identified missing knowledge through query-driven feedback and reinforced into the KG.

In the i-th iteration, a subgraph is retrieved from the KG $G^{(i)}$ and enhanced through triple context restoration to form $G'^{(i)}$. The LLM then infers the answer $A_{\text{LLM}}^{(i)}$ based on $G'^{(i)}$.

Feedback Flow: The missing knowledge in the KG is identified and subsequently integrated back into the KG:

$$G^{(i+1)} \longleftarrow \Delta K^{(i)} \longleftarrow A_{\text{LLM}}^{(i)},$$

where $\Delta K^{(i)}$ represents the knowledge increment corresponding to the missing knowledge. This increment is updated in the KG as triples, resulting in the more comprehensive KG $G^{(i+1)}$.

3.1 Knowledge Graph Construction

An initial KG was constructed from raw textual data using LLMs to extract entities and relations as triples (e_h, r, e_t), where entities include names, types $\text{Type}(e)$, and descriptions $\text{Desc}(e)$. Source document information is retained in each node for provenance. The construction involves:

Document Splitting Each document D of length L is divided into overlapping chunks C_i with maximum length $\text{MAX\_LEN} = 512$ and overlap $\text{OVERLAP} = 64$ tokens:

$$\begin{aligned} C_i &= D[s_i : e_i], \\ s_i &= (i-1) \times (\text{MAX\_LEN} - \text{OVERLAP}) + 1, \\ e_i &= \min(s_i + \text{MAX\_LEN} - 1, L). \end{aligned}$$

The overlap ensures entities and relations spanning across chunks are captured.

Triple Extraction From each chunk C_i, the LLM extracts triples:

$$T_i = \text{ExtractTriples}(C_i),$$

where T_i is the set of triples from C_i. A specialized prompt guides the LLM to output structured information, including entity types and descriptions.

3.2 Subgraph Retrieval

The subgraph retrieval phase focuses on extracting pertinent information from the KG in response to the query. Specifically, given a query Q expressed in natural language, the retrieval stage aims to extract the most relevant elements (e.g., entities, triplets, paths, subgraphs) from KGs, which can be formulated as:

$$\begin{aligned} G^* &= \text{G-Retriever}(Q, G) \\ &= \arg \max_{G \subseteq \mathcal{R}(G)} \text{Sim}(Q, G), \end{aligned}$$

where $G^* = \{(h_0, r_0, s_1), (h_1, r_1, s_2), \ldots, (h_t, r_t, s_{t+1})\}$ is the optimal retrieved graph elements and $\text{Sim}(\cdot, \cdot)$ is a function that measures the semantic similarity between user queries and the graph data. $\mathcal{R}(G)$ represents a function to narrow down the search range of subgraphs, considering efficiency. The retrieval method employed in the TCR-QF builds upon existing KG retrieval method [Sun *et al.*, 2023], which utilize LLMs to perform a beam search over the KG, with iterative pruning guided by the LLM.

3.3 Triple Context Restoration

The structuring of unstructured text into triples can lead to a loss of semantic context. To address this issue, a triple context restoration mechanism was implemented in TCR-QF to restore semantic integrity by tracing back to the original textual context of the triples.

Context Retrieval For each triple (e_h, r, e_t) in the retrieved subgraphs, the source documents associated with the head and tail entities were retrieved:

$$\text{Sources}_{(e_h, e_t)} = \text{Sources}(e_h) \cup \text{Sources}(e_t).$$

These sources were the documents which the entities were originally extracted during KG construction. This set encompassed all documents potentially containing contextual information about the relationship between e_h and e_t.

Triple Context Tracing To trace the context of the triple (e_h, r, e_t), the most relevant sentence from source documents were identified. A template $T_{(e_h,r,e_t)}$ was used, such as:

$$T_{(e_h,r,e_t)} = \text{``}e_h\ r\ e_t\text{''}.$$

A pretrained embedding model f_{embed} was used to generate embeddings for both the template and candidate sentences. The context relevance was assessed via cosine similarity:

$$\mathbf{v}_T = f_{\text{embed}}(T_{(e_h,r,e_t)}), \mathbf{v}_s = f_{\text{embed}}(s), \quad \forall s \in \mathcal{S},$$

$$\text{sim}(\mathbf{v}_T, \mathbf{v}_s) = \frac{\mathbf{v}_T^\top \mathbf{v}_s}{\|\mathbf{v}_T\| \cdot \|\mathbf{v}_s\|}$$

where $\mathcal{S}$ is the set of all sentences extracted from $\text{Sources}_{(e_h,e_t)}$. The sentence with the highest similarity score was selected to provide contextual information into the triple.

Triple Augmentation Each triple was augmented with its associated contextual sentence:

$$(e_h, r, e_t) \longrightarrow (e_h, r, e_t, \mathcal{S}_{\text{top}}).$$

This augmentation restored the contextual information of the triples, improving the accuracy and depth of inference tasks that rely on the KG.

3.4 Iterative Reasoning with Query-driven Feedback

To generate accurate answers to the original queries Q, an iterative reasoning process incorporating a query-driven feedback mechanism was implemented. This approach dynamically enriches the KG by identifying and updating missing information during the reasoning process, thereby enhancing the LLM‘s capability to produce more accurate responses.

Initially, the enriched subgraph $G'^{(0)}$ obtained from triple context restoration was used to prompt the LLM:

$$\mathcal{I}^{(0)} = \text{FormatInput}(Q, G'^{(0)}).$$

The LLM then generated an initial answer $A_{\text{LLM}}^{(0)}$ by processing this prompt:

$$A_{\text{LLM}}^{(0)} = \text{LLM\_Generate}(\mathcal{I}^{(0)}),$$

where LLM_Generate refers to generating a response based on the formatted input $\mathcal{I}^{(0)}$.

Missing Knowledge Identification The initial answer and contexts were analyzed to identify missing information required for the query:

$$\Delta K^{(0)} = \text{IdentifyMissing}(Q, A_{\text{LLM}}^{(0)}, G'^{(0)}),$$

where $\Delta K^{(0)}$ represents the set of missing knowledge, formalized as a series of sub-questions. The function IdentifyMissing utilizes the LLM to compare Q with $A_{\text{LLM}}^{(0)}$ and $G'^{(0)}$, effectively harnessing its understanding to identify gaps in knowledge.

Knowledge Graph Enrichment For each missing component $k_q \in \Delta K^{(0)}$, a dense retriever interacted with the original text sources $\mathcal{D}$ to retrieve relevant textual information and extract the missing knowledge:

$$\mathcal{D}_{\text{relevant}} = \text{DenseRetrieve}(k_q, \mathcal{D}),$$
$$k = \text{ExtractTriples}(k_q, \mathcal{D}_{\text{relevant}}),$$

where ExtractTriples employs the LLM to find and extract the needed information, resulting in triples k corresponding to the missing knowledge. The KG was then updated:

$$G^{(1)} = G^{(0)} \cup \Delta K^{(0)} \quad \text{with} \quad \forall\, k \in \Delta K^{(0)},\ k \notin G^{(0)}.$$

Duplicate relationships were filtered based on edit distance from elements in $G^{(0)}$ to maintain uniqueness in $G^{(1)}$. A dense passage retriever, implemented using OpenAI's `text-embedding-small`, was employed due to its effectiveness in retrieving semantically relevant passages.

Iterative Reasoning and Update The updated KG $G^{(1)}$ was used to generate a new answer by following the reasoning steps:

$$A_{\text{LLM}}^{(1)} = \text{LLM\_Generate}(\text{FormatInput}(Q, G^{(1)})).$$

This iterative process continued, repeating the steps of Missing Knowledge Identification and Knowledge Graph Enrichment:

$$\begin{aligned} \Delta K^{(i)} &= \text{IdentifyMissing}(Q, A_{\text{LLM}}^{(i-1)}, G^{(i-1)}), \\ G^{(i)} &= G^{(i-1)} \cup \Delta K^{(i)}, \\ A_{\text{LLM}}^{(i)} &= \text{LLM\_Generate}(\text{FormatInput}(Q, G^{(i)})), \end{aligned}$$

for $i = 2, 3, \ldots$, until $\Delta K^{(i)} = \emptyset$ or a predefined maximum number of iterations $I_{\max} = 20$ was reached. By analyzing retrieved contexts and generated responses at each iteration, gaps in the KG were detected and addressed, continuously optimizing the KG and enhancing the reasoning capabilities of the LLM.

Due to space constraints, the detailed prompts used for the LLM at each step are provided in the appendix[1].

4 Experiments

To evaluate the effectiveness of the TCR-QF on question-answering tasks, experiments were conducted on 5 question-answering datasets: 2WikiMultiHopQA [Ho *et al.*, 2020], HotpotQA [Yang *et al.*, 2018], ConcurrentQA [Arora *et al.*, 2023], MuSiQue-Ans and MuSiQue-Full [Trivedi *et al.*, 2022] . Followed the settings outlined in [Yang *et al.*, 2018], utilizing a collection of related contexts for each pair as the retrieval corpus. Exact Match (EM) and F1 score were presented as the evaluation metrics across all datasets.

We compared TCR-QF with representative methods from LLMs and RAG:
(1) LLM Only: Methods that directly use LLMs for obtaining answers, including models such as `gpt-4o-mini` and

[1] The appendix is available in the arXiv version: https://arxiv.org/abs/2501.15378.

`gpt-4o`, as well as chain-of-thought (CoT) [Wei *et al.*, 2022] prompting strategies.

(2) Text-based RAG: Methods that employ a dense retriever to retrieve relevant text chunks from a text corpus and generate answers by leveraging this information. For this category, LangChainQ&A[2] was used as a representative naive RAG method, which is well-known and widely used.

(3) Graph-based RAG: Methods that retrieve subgraphs from KG to enhance LLM. ToG [Sun *et al.*, 2023] was selected as a representative for comparison in this category.

(4) Hybrid RAG: Methods like GraphRAG [Edge *et al.*, 2024] that retrieve information from both KG and textual documents to augment LLM.

Experimental Settings: For all comparison methods and the TCR-QF, unless otherwise specified, the `gpt-4o-mini-2024-07-18` model was utilized. Due to the high computational costs associated with inference on the full dataset, 1,200 samples were randomly selected from each of the larger datasets—2WikiMultiHopQA, HotpotQA, MuSiQue-Full, and MuSiQue-Ans—for testing to conserve computational resources. For ConcurrentQA, 1,600 samples from the complete test set were evaluated.

4.1 Results and Findings

Table 1 presents the comparative results, from which we answer the following Research Question (RQ).

RQ1 *How does the TCR-QF improve the completeness and accuracy of information retrieval in question answering tasks compared to the existing GraphRAG methods?*

Table 1 demonstrates the superiority of the TCR-QF compared to different methods on five benchmark question answering datasets. TCR-QF consistently achieves the highest EM and F1 scores across all datasets, demonstrating its superior effectiveness in enhancing LLMs for complex reasoning tasks. Compared to the **LLM-only** approaches (**GPT-4o-mini**, **GPT-4o** and **CoT**), TCR-QF shows substantial improvements. For instance, on the HotpotQA dataset, TCR-QF attains an EM score of 0.558, which is **0.207** higher than GPT-4o's score of 0.351, representing a relative improvement of approximately **59%**. This indicates that while LLMs possess strong language understanding capabilities, integrating external knowledge as TCR-QF does markedly enhances their accuracy in answering complex questions.

When contrasting TCR-QF with the **text-based** method (**Naive RAG**) and the **graph-based** method (**ToG**), TCR-QF exhibits notable performance gains. Specifically, on the 2WikiMultiHopQA dataset, TCR-QF achieves an EM score of **0.598**, which is an absolute increase of **0.259** over Naive RAG's EM score of **0.339**—a relative improvement of approximately **76.4%**. Similarly, TCR-QF surpasses TOG's EM score of **0.400** by an absolute margin of **0.198**, reflecting a **49.5%** improvement. This significant enhancement indicates that TCR-QF's approach of enriching the LLM with more comprehensive knowledge markedly improves reasoning, outperforming methods that rely solely on retrieved texts or static KGs.

Furthermore, TCR-QF outperforms the **hybrid** method (**GraphRAG**), which combines text and graph information. On the MuSiQue-Full dataset, TCR-QF achieves an EM score of **0.303**, compared to GraphRAG's EM score of **0.189**. This represents an absolute increase of **0.114**, amounting to an improvement of approximately **60.3%**. These significant gains demonstrate that TCR-QF effectively leverages knowledge to enhance the LLM's performance beyond what is achieved by simply combining text and graph data. By dynamically restoring lost semantic information and enriching the KG during reasoning, TCR-QF provides a more comprehensive context for the LLM, leading to better reasoning and answer generation in complex tasks.

The consistent superiority of TCR-QF across multiple datasets—ranging from general question-answering to those requiring multi-hop reasoning—highlights TCR-QF's robustness and general applicability. TCR-QF effectively addresses the challenges posed by incomplete KGs and information loss, leading to more accurate and complete responses.

4.2 Ablation Study

To evaluate the individual contributions of the proposed components, namely *triple context restoration* (TCR) and *query-driven feedback* (QF), to the overall performance of the TCR-QF, an ablation study was conducted on the 2WikiMultiHopQA and HotpotQA datasets to answer the question:

RQ2 *In what ways does each component in TCR-QF enhance the reasoning of the LLM?*

Table 2 presents the results of the ablation experiments. The full **TCR-QF** is compared with several ablated variants:

- **ToG (w/o TCR & QF)**: The baseline method operating on the KG.
- **TCR (w/o QF)**: Incorporates triple context restoration alone to address contextual information loss.
- **QF (w/o TCR)**: Employs query-driven feedback alone to approach incomplete information extraction.
- **TCR-AF**: Integrate triple context restoration with *answer-driven feedback* (AF) which involves directly extracting triples from the LLM's answer and adding them to the KG.

From the results we can draw the following insights.

Effectiveness of Triple Context Restoration (TCR). Comparing the baseline **ToG** method with the **TCR** variant, it can be observed that introducing triple context restoration leads to significant performance improvements. On the 2WikiMultiHopQA dataset, the EM score increases from 0.400 to 0.481, representing an improvement of 20.25%, while the F1 score rises from 0.476 to 0.561. Similarly, on the HotpotQA dataset, the EM score improves from 0.420 to 0.494 (a 17.62% improvement), and the F1 score increases from 0.555 to 0.642. These enhancements confirm that triple context restoration effectively mitigates contextual information loss by reconnecting structured triples with their original textual context, thereby enriching the semantic information available for reasoning.

[2] https://python.langchain.com/docs/tutorials/rag

Method Type	Method	2WikiMultiHopQA		HotpotQA		MuSiQue-Full		MuSiQue-Ans		ConcurrentQA	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
LLM only	GPT-4o-mini	0.266	0.320	0.273	0.381	0.048	0.132	0.052	0.135	0.112	0.178
	GPT-4o	0.311	0.364	0.351	0.475	0.089	0.193	0.104	0.215	0.176	0.247
	CoT	0.287	0.354	0.299	0.420	0.093	0.196	0.117	0.219	0.134	0.203
Text-based	Naive RAG	0.339	0.391	0.411	0.530	0.111	0.207	0.122	0.221	0.363	0.443
Graph-based	TOG	0.400	0.476	0.420	0.555	0.136	0.237	0.160	0.269	0.278	0.359
Hybrid	GraphRAG	0.485	0.626	0.495	0.645	0.189	0.326	0.258	0.395	0.459	0.582
Proposed	**TCR-QF**	**0.598**	**0.680**	**0.558**	**0.708**	**0.303**	**0.432**	**0.366**	**0.489**	**0.492**	**0.597**

Table 1: Main Results. Performance comparison of different methods across five question answering datasets.

Methods	2WikiMultiHopQA		HotpotQA	
	EM	F1	EM	F1
ToG(w/o TCR&QF)	0.400	0.476	0.420	0.555
TCR(w/o QF)	0.481	0.561	0.494	0.642
QF(w/o TCR)	0.568	0.651	0.515	0.656
TCR-AF	0.538	0.619	0.531	0.682
TCR-QF	**0.598**	**0.680**	**0.558**	**0.708**

Table 2: Ablation experiment results on the 2WikiMultiHopQA and HotpotQA datasets. **TCR** stands for triple context restoration, **QF** stands for query-driven feedback. **TCR-AF** indicates replacing query-driven feedback with answer-driven feedback which directly extract triples from the answers and feed them back into the KG.

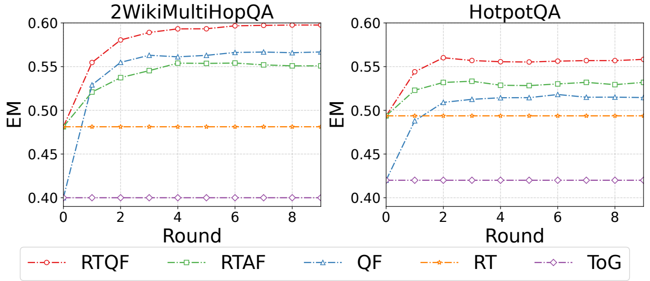

Figure 3: Comparative ressults from the ablation study. EM performance of different methods across rounds on 2WikiMultiHopQA and HotpotQA.

Effectiveness of Query-Driven Feedback (QF). The **QF** variant, which focuses on dynamically updating the KG based on the requirements of the query, shows even greater improvements over the baseline. The EM scores rise to 0.568 (a 42.00% improvement) on 2WikiMultiHopQA and 0.522 (a 24.29% improvement) on HotpotQA. These substantial gains indicates that query-driven feedback significantly addresses the issue of incomplete information extraction. By dynamically enriching the KG based on the specific requirements of the query, the model fills in the missing knowledge that static KGs might overlook due to limitations in initial extraction algorithms. This adaptive approach continually enhances the relevance and comprehensiveness of the knowledge graph throughout the reasoning process.

Synergy of TCR and QF. The full **TCR-QF** method, which combines both triple context restoration and query-driven feedback, achieves the highest performance. EM scores reach 0.598 on 2WikiMultiHopQA and 0.558 on HotpotQA, with relative improvements of 49.50% and 32.86% over the baseline, respectively. These results underscore a synergistic effect when combining TCR and QF, as the model benefits from both restored contextual semantics and a dynamically enriched KG. The integration of both components effectively addresses the dual challenges of information loss, leading to more accurate and complete reasoning.

Comparison with Answer-Driven Feedback (TCR-AF). The **TCR-AF** variant replaces query-driven feedback with answer-driven feedback, where triples are extracted from the model's answers to update the KG. While TCR-AF outperforms **ToG**, achieving EM scores of 0.538 on 2WikiMultiHopQA and 0.523 on HotpotQA, it falls short compared to **TCR-QF**. TCR-QF scores 0.598 on 2WikiMultiHopQA, an 11.12% increase over TCR-AF. This suggests that enriching the KG proactively based on the query is more effective than reactively updating it based on answers, likely because it prevents error propagation from incomplete initial reasoning.

Performance Trends Across Rounds. Figure 3 illustrates the EM performance across multiple reasoning rounds for each method. It is evident that **TCR-QF** consistently outperforms other variants from the initial rounds and maintains its lead as the reasoning progresses. The performance gain from TCR-QF is additive, with TCR-QF achieving the highest accuracy. The diminishing returns after a few rounds indicate that the most significant knowledge enrichment occurs early in the reasoning process, emphasizing the performance of proposed method.

In conclusion, the ablation study corroborates our initial hypotheses, demonstrating that both *triple context restoration* and *query-driven feedback* are vital in addressing the inherent limitations of integrating KGs with LLMs. Individually, each component contributes significantly to performance improvements by targeting specific sources of information loss—triple context restoration restores essential contextual semantics lost during the structuring process, while query-driven feedback dynamically enriches the KG to address incomplete information extraction. These results highlight the effectiveness of restoring semantic integrity and continuously updating the KG during reasoning, fulfilling our research objectives and underscoring the importance of a bidirectional knowledge flow in optimizing reasoning outcomes.

4.3 Statistical and Convergence Analysis

To evaluate the effectiveness and convergence of the TCR-QF, statistical analyses were conducted over multiple inference rounds. Table 3 presents key metrics from the initial round to the 10th round, including the numbers of nodes and edges in the KG, as well as the EM and F1 scores on the 2WikiMultiHopQA dataset. These experiments and results provide answers to the following question:

RQ3 *How do TCR-QF continuously enhance KG and boost LLM reasoning?*

	2WikiMultiHopQA			
Rounds	**Nodes**	**Edges**	**EM**	**F1**
0	74,571	69,866	0.481	0.562
1	76,441	74,006	0.555	0.637
2	77,377	76,615	0.581	0.662
3	77,937	78,265	0.589	0.671
4	78,259	79,258	0.593	0.676
5	78,450	79,840	0.593	0.675
6	78,570	80,150	0.597	0.679
7	78,630	80,310	0.597	0.679
8	78,650	80,403	0.598	0.680
9	78,656	80,446	0.598	0.680
10	78,661	80,472	0.598	0.680
Δ	4,090	10,606	0.117	0.118

Table 3: Statistics from the initial round to the 10th round on 2WikiMultiHopQA dataset, where Δ denotes the cumulative increase.

From the results we can draw the following insights.

Continuous Improvement of KG Completeness and Model Reasoning Performance. As demonstrated in Table 3, the TCR-QF significantly enriches the KG over successive inference rounds. Specifically, on the 2WikiMultiHopQA dataset, the number of nodes in the KG increased by 4,090 (from 74,571 to 78,661), and the number of edges increased by 10,606 (from 69,866 to 80,472) over 10 rounds. This enrichment directly addresses the issue of information sparsity by incorporating previously missing triples and expanding the KG's coverage to meet query demands. Correspondingly, the model's reasoning performance improved substantially. The Exact Match (EM) score increased from 0.481 to 0.598, a 24.3% improvement, and the F1 score rose from 0.562 to 0.680, a 21.0% improvement. These significant performance gains indicate that the enriched KG provides the LLM with more comprehensive and contextually rich information, directly mitigating the effects of context loss and enhancing reasoning accuracy.

Alignment of KG Completeness and Reasoning Performance Enhancement. As depicted in Figure 4, the parallel upward trends in KG metrics and performance scores affirm a strong correlation between the enriched KG and the model's improved reasoning ability. By restoring the contextual information associated with triples and integrating new, relevant knowledge through query-driven feedback, the TCR-QF enhances the semantic integrity of the KG. This comprehensive knowledge base enables the LLM to perform more accurate and context-aware reasoning, directly addressing the limitations posed by information sparsity and context loss.

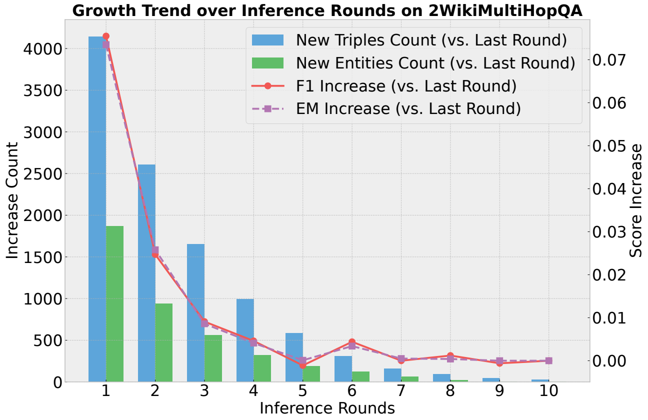

Figure 4: Trends in KG growth and inference performance improvement over rounds on 2WikiMultiHopQA.

Convergence of KG Enrichment and Performance Improvements. The TCR-QF not only enriches the KG but also exhibits convergence over inference rounds, ensuring efficient use of computational resources. As illustrated in Figure 4, both the growth of the KG and the improvement in performance metrics begin to plateau after several rounds, specifically between the 8th and 10th iterations. The incremental increases in nodes and edges diminish, and the EM and F1 scores stabilize at 0.598 and 0.680, respectively. This convergence suggests that the TCR-QF effectively enriches the KG to an optimal level, beyond which additional iterations yield minimal benefits.

The experimental results validate the effectiveness of the TCR-QF in overcoming the foundational challenges outlined in the introduction. By continuously and efficiently enhancing the KG's completeness and restoring lost contextual nuances, the TCR-QF significantly boosts the model's reasoning performance. These findings confirm that addressing information loss through dynamic KG enrichment and context restoration is a viable and efficient strategy for advancing the integration of KGs and LLMs in complex reasoning tasks.

5 Conclusion

This paper introduces **TCR-QF**, a novel framework that integrates knowledge graphs (KGs) with large language models (LLMs) to enhance complex question answering. By mitigating *context loss* through triple context restoration (TCR) and addressing *incomplete extraction* with query-driven feedback (QF), TCR-QF recovers key semantic details and dynamically expands the KG during reasoning. Experiments on five benchmarks show that TCR-QF outperforms state-of-the-art methods, demonstrating the benefits of contextualized triples and iterative KG updates. These results highlight the potential of TCR-QF to bridge the gap between structured and unstructured knowledge, paving the way for more accurate and robust AI-driven reasoning across diverse domains.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62120106008), the Anhui Provincial Science and Technology Fortification Plan (Grant No. 202423k09020015), and the Youth Talent Support Program of the Anhui Association for Science and Technology (Grant No. RCTJ202420). The authors gratefully acknowledge these supports.

References

[Arora *et al.*, 2023] Simran Arora, Patrick S. H. Lewis, Angela Fan, Jacob Kahn, and Christopher Ré. Reasoning over public and private data in retrieval-based systems. *Trans. Assoc. Comput. Linguistics*, 11:902–921, 2023.

[Biswas *et al.*, 2024] Russa Biswas, Harald Sack, and Mehwish Alam. MADLINK: attentive multihop and entity descriptions for link prediction in knowledge graphs. *Semantic Web*, 15(1):83–106, 2024.

[Chen *et al.*, 2024a] Hanzhu Chen, Xu Shen, Qitan Lv, Jie Wang, Xiaoqi Ni, and Jieping Ye. SAC-KG: exploiting large language models as skilled automatic constructors for domain knowledge graphs. *CoRR*, abs/2410.02811, 2024.

[Chen *et al.*, 2024b] Weijie Chen, Ting Bai, Jinbo Su, Jian Luan, Wei Liu, and Chuan Shi. Kg-retriever: Efficient knowledge indexing for retrieval-augmented large language models. *CoRR*, abs/2412.05547, 2024.

[Cohen *et al.*, 2023] William W. Cohen, Wenhu Chen, Michiel de Jong, Nitish Gupta, Alessandro Presta, Pat Verga, and John Wieting. QA is the new KR: question-answer pairs as knowledge bases. In *Proceedings of 37th Conference on AAAI*, pages 15385–15392. AAAI Press, 2023.

[Dehghan *et al.*, 2024] Mohammad Dehghan, Mohammad Ali Alomrani, Sunyam Bagga, David Alfonso-Hermelo, Khalil Bibi, Abbas Ghaddar, Yingxue Zhang, Xiaoguang Li, Jianye Hao, Qun Liu, Jimmy Lin, Boxing Chen, Prasanna Parthasarathi, Mahdi Biparva, and Mehdi Rezagholizadeh. EWEK-QA : Enhanced web and efficient knowledge graph retrieval for citation-based question answering systems. In *Proceedings of the 62nd ACL 2024*, pages 14169–14187. ACL, 2024.

[Delile *et al.*, 2024] Julien Delile, Srayanta Mukherjee, Anton Van Pamel, and Leonid Zhukov. Graph-based retriever captures the long tail of biomedical knowledge. *CoRR*, abs/2402.12352, 2024.

[Edge *et al.*, 2024] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130, 2024.

[He *et al.*, 2024] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *CoRR*, abs/2402.07630, 2024.

[Ho *et al.*, 2020] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th ICCL*, pages 6609–6625. ICCL, December 2020.

[Jiang *et al.*, 2023] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of Conference on EMNLP 2023*, pages 9237–9251. ACL, 2023.

[Kim *et al.*, 2023] Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *Findings of the EMNLP 2023*, pages 9410–9421. ACL, 2023.

[Li *et al.*, 2023] Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *CoRR*, abs/2304.11633, 2023.

[Liang *et al.*, 2024] Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, Huaidong Xiong, Lin Yuan, Jun Xu, Zaoyang Wang, Zhiqiang Zhang, Wen Zhang, Huajun Chen, Wenguang Chen, and Jun Zhou. Kag: Boosting llms in professional domains via knowledge augmented generation. *CoRR*, abs/2409.13731, 2024.

[Munikoti *et al.*, 2023] Sai Munikoti, Anurag Acharya, Sridevi Wagle, and Sameera Horawalavithana. ATLANTIC: structure-aware retrieval-augmented language model for interdisciplinary science. *CoRR*, abs/2311.12289, 2023.

[Pan *et al.*, 2024] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.*, 36(7):3580–3599, 2024.

[Panda *et al.*, 2024] Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, and Prathosh A P. HOLMES: hyper-relational knowledge graphs for multi-hop question answering using llms. In *Proceedings of 62nd Conference on ACL*, pages 13263–13282. ACL, 2024.

[Paulheim, 2017] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508, 2017.

[Peng *et al.*, 2024] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *CoRR*, abs/2408.08921, 2024.

[Soman *et al.*, 2024] Karthik Soman, Peter W Rose, John H Morris, Rabia E Akbas, Brett Smith, Braian Peetoom, Catalina Villouta-Reyes, Gabriel Cerono, Yongmei Shi, Angela Rizk-Jackson, et al. Biomedical knowledge graph-optimized prompt generation for large language models. *Bioinformatics*, 40(9):btae560, 2024.

[Sun *et al.*, 2023] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *CoRR*, abs/2307.07697, 2023.

[Sun *et al.*, 2024] Qiang Sun, Yuanyi Luo, Wenxiao Zhang, Sirui Li, Jichunyang Li, Kai Niu, Xiangrui Kong, and Wei Liu. Docs2kg: Unified knowledge graph construction from heterogeneous documents assisted by large language models. *CoRR*, abs/2406.02962, 2024.

[Taunk *et al.*, 2023] Dhaval Taunk, Lakshya Khanna, Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. Grapeqa: Graph augmentation and pruning to enhance question-answering. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben, editors, *Companion Proceedings of the Conference on WWW 2023*, pages 1138–1144. ACM, 2023.

[Trisedya *et al.*, 2019] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of Conference on AAAI*, volume 33, pages 297–304, 2019.

[Trivedi *et al.*, 2022] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 2022 Conference of NeurIPS*, 2022.

[Xu *et al.*, 2024a] Chengjin Xu, Muzhi Li, Cehao Yang, Xuhui Jiang, Lumingyuan Tang, Yiyan Qi, and Jian Guo. Move beyond triples: Contextual knowledge graph representation and reasoning. *arXiv preprint arXiv:2406.11160*, 2024.

[Xu *et al.*, 2024b] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: a survey. *Frontiers Comput. Sci.*, 18(6):186357, 2024.

[Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the Conference on EMNLP 2018*, pages 2369–2380. ACL, 2018.

[Yang *et al.*, 2024] Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *IEEE Trans. Knowl. Data Eng.*, 36(7):3091–3110, 2024.

[Yasunaga *et al.*, 2021] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of NAACL*, pages 535–546. ACL, 2021.

[Zhang and Soh, 2024] Bowen Zhang and Harold Soh. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. In *Proceedings of the 2024 Conference on EMNLP*, pages 9820–9836. ACL, 2024.

[Zhang *et al.*, 2024] Qinggang Zhang, Junnan Dong, Hao Chen, Daochen Zha, Zailiang Yu, and Xiao Huang. Knowgpt: Knowledge graph based prompting for large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *In Proceedings of Conference on NeurIPS 2024*, 2024.

[Zhang *et al.*, 2025a] Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. RATT: A thought structure for coherent and correct LLM reasoning. In *Proceedings of 39th Conference on AAAI 2025*, pages 26733–26741. AAAI Press, 2025.

[Zhang *et al.*, 2025b] Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. A survey of graph retrieval-augmented generation for customized large language models. *CoRR*, abs/2501.13958, 2025.

[Zhong *et al.*, 2024] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.*, 56(4):94:1–94:62, 2024.

[Zhu *et al.*, 2024] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities. *World Wide Web (WWW)*, 27(5):58, 2024.