

MultiRAG: A Knowledge-guided Framework for Mitigating Hallucination in Multi-source Retrieval Augmented Generation

Wenlong Wu¹, Haofen Wang², Bohan Li^{1,3,4✉}, Peixuan Huang¹, Xinzhe Zhao¹ and Lei Liang⁵

¹College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics,

Key Laboratory of Brain-Machine Intelligence Technology, Ministry of Education

²College of Design & Innovation, Tongji University

³Key Laboratory of Intelligent Decision and Digital Operation, Ministry of Industry and Information Technology

⁴Collaborative Innovation Center of Novel Software Technology and Industrialization

⁵Ant Group Knowledge Graph Team

Email: {wuwenlong, bhli, peixuanh, xinzhe_zhao}@nuaa.edu.cn

carter.whfcarter@gmail.com, leywar.liang@antgroup.com

Abstract—Retrieval Augmented Generation (RAG) has emerged as a promising solution to address hallucination issues in Large Language Models (LLMs). However, the integration of multiple retrieval sources, while potentially more informative, introduces new challenges that can paradoxically exacerbate hallucination problems. These challenges manifest primarily in two aspects: the sparse distribution of multi-source data that hinders the capture of logical relationships and the inherent inconsistencies among different sources that lead to information conflicts. To address these challenges, we propose MultiRAG, a novel framework designed to mitigate hallucination in multi-source retrieval-augmented generation through knowledge-guided approaches. Our framework introduces two key innovations: (1) a knowledge construction module that employs multi-source line graphs to efficiently aggregate logical relationships across different knowledge sources, effectively addressing the sparse data distribution issue; and (2) a sophisticated retrieval module that implements a multi-level confidence calculation mechanism, performing both graph-level and node-level assessments to identify and eliminate unreliable information nodes, thereby reducing hallucinations caused by inter-source inconsistencies. Extensive experiments on four multi-domain query datasets and two multi-hop QA datasets demonstrate that MultiRAG significantly enhances the reliability and efficiency of knowledge retrieval in complex multi-source scenarios. Our code is available in <https://github.com/wuwenlong123/MultiRAG>.

Index Terms—Retrieval Augmented Generation, Large Language Models, Multi-source Retrieval, Knowledge Graphs, Hallucination Mitigation

I. INTRODUCTION

Large Language Models (LLMs) have achieved remarkable success in handling a variety of natural language processing tasks, attributable to their robust capabilities in understanding and generating language and symbols [1]. In knowledge-intensive retrieval tasks, Retrieval Augmented Generation (RAG) has become a standardized solution paradigm [2]–[4]. Previous works [5]–[11] have made significant strides in

addressing the inherent knowledge limitations of LLMs. By introducing external knowledge bases, it has markedly improved the accuracy and fidelity of LLM responses. However, recent studies have highlighted a significant drawback: the retrieval results of RAG are imperfect, including irrelevant, misleading, and even malicious information, ultimately leading to inaccurate LLM responses.

To address these limitations, the synergy between LLMs and Knowledge Graphs (KGs) has been proposed to achieve more efficient information retrieval [12]. On one hand, KG can efficiently store data with fixed characteristics (such as temporal KGs, event KGs, etc.), thereby enhancing the processing capabilities of LLMs on specific data [13]–[20]. On the other hand, the collaboration between LLMs and KGs has significantly improved performance in multi-hop and multi-document question answering, including the credibility and interpretability of retrieval [21]. Furthermore, LLM-KG collaborative methods have also provided the latest solutions for knowledge-intensive retrieval tasks [22]–[26], propelling the deep reasoning capabilities of RAG.

Nevertheless, existing frameworks still fail to account for the complexity of real-world data. Although RAG can mitigate the generation of hallucinations, these hallucinations often stem from the internal knowledge of LLMs [27]–[29]. Inconsistent information sources and unreliable retrieval methods can still lead to retrieval biases and hallucinations in LLMs. This issue becomes particularly pronounced when dealing with information retrieval tasks that involve multi-source knowledge, where hallucinations are more prominent. Research [30] indicates that approximately 70% of retrieved paragraphs do not directly contain the correct query answers but instead include information indirectly related to the answers, causing misguidance and comprehension bias in LLMs.

Building upon the categorization of hallucinations in retrieval [9], we outline the three most common types of hallucinations encountered in multi-source data retrieval:

Wenlong Wu and Haofen Wang contributed equally to this work.
Bohan Li is the corresponding author.

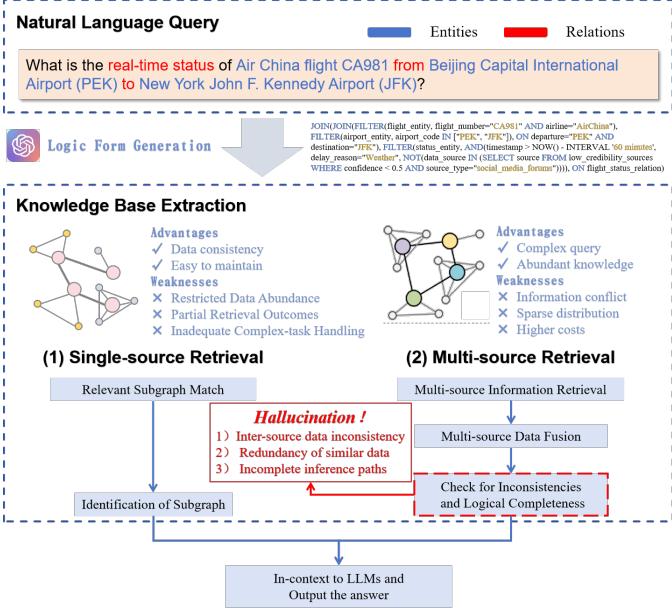


Fig. 1: Single-source Retrieval & Multi-source Retrieval

- 1) **Inter-source data inconsistency:** Discrepancies between different data sources can lead to conflicting information, causing hallucinations in LLMs.
- 2) **Redundancy of similar data:** There often exists data that is highly similar and semantically equivalent across multiple data sources, which can impose significant computational overhead on retrieval.
- 3) **Incomplete inference paths:** Forming a comprehensive inference path from different data sources is challenging. Existing retrievers often fail to capture the complete logical associations within multiple data sources.

Fig. 1 vividly illustrates the differences between single-source and multi-source data retrieval through CA981 flight analysis. The sparse distribution and inconsistency of data are unique issues in multi-source data retrieval, leading to severe hallucination bias in LLMs. Against this backdrop, we focus on addressing the issue of retrieval hallucinations in multi-source data retrieval to empower knowledge-augmented generation. This work primarily explores the following two fundamental challenges:

- 1) **Sparse Distribution of Multi-source Data:** Multi-domain queries require fusing structured (SQL tables), semi-structured (JSON logs), and unstructured data (text reports). Due to the variability in data storage formats and sparsity, the connectivity between knowledge elements is low, making it difficult for RAG systems to effectively capture logical associations across sources, thereby affecting the recall rate and quality of retrieval results.
- 2) **Inter-source Data Inconsistency:** Conversely, the inherent diversity in knowledge representations across multi-source data often leads to inconsistencies in retrieved fragments. These discrepancies may induce information

conflicts during retrieval processes, thereby compromising response accuracy. This challenge becomes particularly pronounced in domain-specific complex reasoning and multi-hop question answering tasks.

To address these issues above, we propose MultiRAG, a novel framework designed to mitigate hallucination in multi-source retrieval augmented generation through knowledge-guided approaches. Initially, we introduce multi-source line graphs for rapid aggregation of knowledge sources to tackle issues arising from sparse data distribution. Subsequently, based on these integrated multi-source line graphs, we propose a multi-level confidence calculation method to ensure the reliability of multi-source data queries. This approach not only enhances query efficiency but also strengthens the accuracy of results, providing an effective solution for the multi-source knowledge-guided RAG.

The contributions of this paper are summarized as follows:

- 1) **Multi-source Knowledge Aggregation:** In the knowledge construction module, we introduce multi-source line graphs as a data structure for rapid aggregation and reconstruction of knowledge structures from multiple query-relevant data sources. This effectively captures inter-source data dependencies within chunk texts, thereby providing a unified and centralized representation of multi-source knowledge.
- 2) **Multi-level Confidence Calculation:** In the retrieval module, we perform graph-level and node-level confidence calculations on the extracted knowledge subgraphs. The aim is to filter out and eliminate low-quality subgraphs and inconsistent retrieval nodes, ultimately enhancing the quality of text embedded in context to alleviate retrieval hallucinations.
- 3) **Experimental Validation and Performance Comparison:** We conducted extensive experiments on existing multi-source retrieval datasets and two complex Q&A datasets, comparing our approach with existing state-of-the-art(SOTA) methods. This demonstrated the robustness and accuracy of our proposed method in retrieval performance. Particularly in multi-source data retrieval tasks, our method significantly outperforms other SOTA methods by more than 10%.

II. PRELIMINARY

In the field of Knowledge-Guided RAG, the primary challenges include efficiently accessing relevant knowledge and achieving reliable retrieval performance. This section introduces the core elements of our approach and precisely defines the problems we address.

Let $Q = \{q_1, q_2, \dots, q_n\}$ be the set of query instances, where each q_i corresponds to a distinct query. Let $E = \{e_1, e_2, \dots, e_m\}$ be the set of entities in the knowledge graph, where each e_j represents an entity. Let $R = \{r_1, r_2, \dots, r_p\}$ be the set of relationships in the knowledge graph, where each r_k represents a relationship. Let $D = \{d_1, d_2, \dots, d_l\}$ be the set of documents, where each d_l represents a document. We

define the knowledge-guided retrieval enhancement generation problem as follows:

$$\arg \max_{d_i \in D} LLM(q_i, d_i), \sum_{e_j \in E} \sum_{r_k \in R} KG(e_j, r_k, d_i) \quad (1)$$

where $LLM(q_i, d_i)$ denotes the score of the relevance between query q_i and document d_i assessed by the LLM, and $KG(e_j, r_k, d_i)$ represents the degree of match between entity e_j , relationship r_k , and document d_i .

Furthermore, we optimize the knowledge construction and retrieval modules by introducing multi-source line graphs to accelerate knowledge establishment and enhance retrieval robustness. Specifically, the proposed approach is formally defined as follows:

Definition 1. Multi-source data fusion. Given a set of sources H , the data $D = \{d, name, c, meta\}$ exists, where d represents the domain of data, c represents the content of the data file, $name$ represents the file/attribute name, and $meta$ represents the file metadata. Through a multi-source data fusion algorithm, we can obtain normalized data $\hat{D} = \{id, d, name, jsc, meta, (cols_index)\}$. Here, id represents the unique identifier for normalization, d indicates the domain where the data file is located, $name$ denotes the data file name, $meta$ denotes the file metadata, and jsc denotes the file content stored using JSON-LD. If the stored data is structured data or other data formats that can use a columnar storage model, the column index $cols_index$ of all attributes will also be stored for rapid retrieval and query. Fig. 2 provides an example of JSON-LD format.

Definition 2. Multi-source line graph [31]. Given a multi-source knowledge graph \mathcal{G} and a transformed knowledge graph \mathcal{G}' (multi-source line graph, MLG), the MLG satisfies the following characteristics:

- 1) A node in \mathcal{G}' represents a triplet.
- 2) There is an associated edge between any two nodes in \mathcal{G}' if and only if the triples represented by these two nodes share a common node.

Based on the definition, it can be inferred that MLG achieves high aggregation of related nodes, which can greatly improve the efficiency of data retrieval and accelerate subsequent retrieval and query algorithms.

Definition 3. Multi-source homologous data. For any two nodes v_1 and v_2 in \mathcal{G} , they are defined as multi-source homologous if and only if they belong to the same retrieval candidate set in a single search.

Definition 4. Homologous node and homologous subgraph. Given a set of multi-domain homologous data $SV = \{v_i\}_{i=1}^n$ in the knowledge graph \mathcal{G} , we define the homologous center node as $snode = \{name, meta, num, C(v)\}$, the set of homologous nodes as U_{snode} , and the set of homologous edges as E_{snode} . Here, $name$ represents the common attribute name, $meta$ denotes the identical file metadata, num indicates the number of homologous data instances, and $C(v)$ represents the data confidence. We define the association edge between $snode$ and v_i as $e_i = \{w_i\}_{i=1}^n$, where w_i represents the weight

```
{
  "@context": "https://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

Fig. 2: Data format of JSON-LD

of node v_i in the data confidence calculation. Thus, the homologous center node and SG together form the homologous subgraph $subSG$.

Definition 5. Homologous triple line graph. For all homologous subgraphs within the knowledge graph \mathcal{G} , they collectively constitute the homologous knowledge graph SG . By performing a linear graph transformation on the homologous knowledge graph, we obtain the homologous triple line graph SG' .

By constructing a homologous triple line graph, multi-source homologous data are aggregated into a single subgraph, centered around homologous nodes, enabling rapid consistency checks and conflict feedback for homologous data. Additionally, the knowledge graph contains a significant number of isolated nodes (i.e., nodes without homologous data), which are also incorporated into the homologous triple line graph.

Definition 6. Candidate graph confidence and candidate node confidence. For a query $Q(q, \mathcal{G})$ on the knowledge graph \mathcal{G} , the corresponding Homologous line graph SG' is obtained. The candidate graph confidence is an estimation of the confidence in the candidate Homologous subgraph, assessing the overall credibility of the candidate graph; the candidate node confidence is an assessment of the confidence in individual node to determine the credibility of single attribute node.

III. METHODOLOGY

A. Framework of MultiRAG

This section elaborates on the implementation approach of MultiRAG. As shown in Fig. 3, the first step involves segmenting and extracting multi-source data to construct the corresponding MLG, achieving preliminary aggregation of multi-source data; the second step requires reconstructing the MLG and performing subgraph extraction to identify candidate homologous subgraphs, ensuring consistent storage of homologous data for subsequent hallucination assessment; the third step involves calculating the graph-level and node-level confidence of the candidate subgraphs, eliminating low-quality nodes to enhance the credibility of the response, and returning the extracted trustworthy subgraphs to the LLM to form the final answer. Finally, integrating the aforementioned steps to form the Multi-source Line Graph Prompting algorithm, MKLGP.

B. Multi-source Line Graph Construction

The MultiRAG method initially employs an adapter structure to integrate multi-source data and standardize its storage

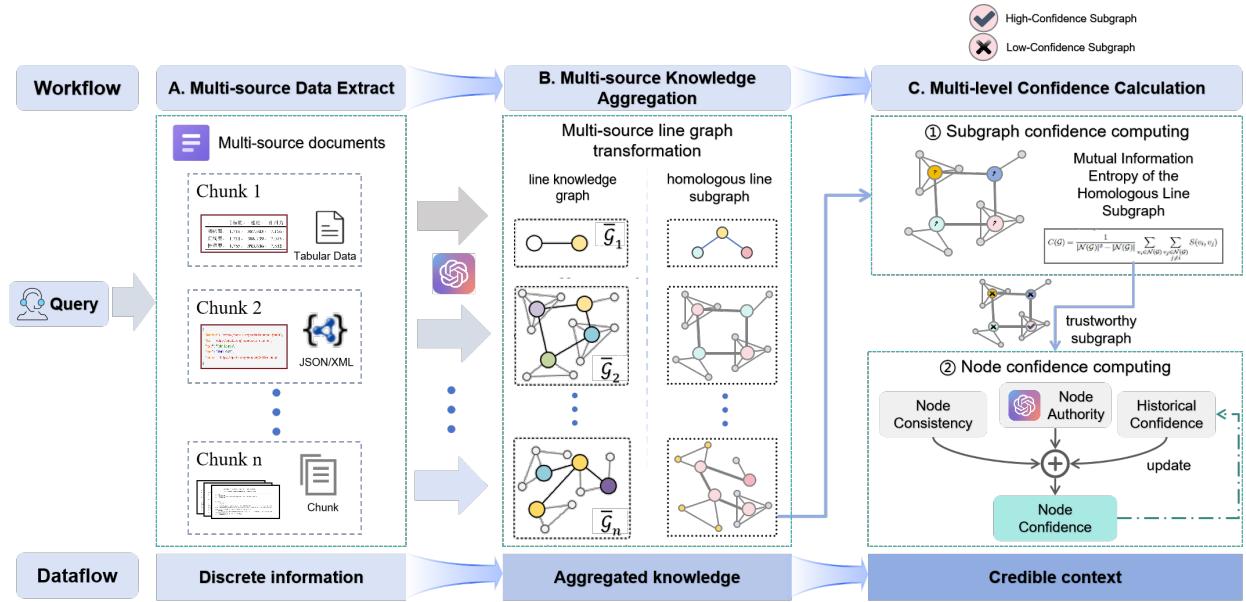


Fig. 3: Framework of MultiRAG, including three modules.

format. For practical application scenarios, data is directly obtained from various non-homologous formats and transformed into a unified, normalized representation. Specifically, file names and metadata are parsed, and the domains to which the files belong are categorized. Subsequently, the data content is parsed and stored in JSON-LD format, thereby transforming it into linked data. Finally, unique identifiers are assigned to the data, resulting in normalized datasets.

Specifically, a unique adapter is designed for each distinct data format to facilitate data parsing. Although the implementation frameworks of these adapters are largely similar, it is essential to differentiate between the parsing processes for structured, semi-structured, and unstructured data.

For structured data, parsing involves storing tabular information in JSON format, where attribute variables within the file are managed using a Decomposition Storage Model (DSM). This approach enables the extraction of all attribute information for consistency checks through the use of column indices. In the case of semi-structured data, parsing corresponds to storing tree-shaped data in JSON format with multi-layer nested structures. This data format lacks column indices and does not support fast retrieval, necessitating the use of tree or graph retrieval algorithms, such as DFS, for efficient searching. Finally, for unstructured data, the focus is currently limited to textual information, which is stored directly. Subsequent steps involve leveraging LLMs for entity and relationship extraction tasks to obtain the relevant information.

The final integration of multi-source data can be expressed by the following formula:

$$D_{\text{Fusion}} = \bigcup_{i=1}^n A_i(D_i) \quad (2)$$

where $A_i \in \{Ada_{\text{stru}}, Ada_{\text{semi-s}}, Ada_{\text{unstru}}\}$, representing the adapter parsing functions for structured data, semi-

structured data, and unstructured data, respectively. $D_i \in \{D_{\text{stru}}, D_{\text{semi-s}}, D_{\text{unstru}}\}$ represents the original datasets of structured data, semi-structured data, and unstructured data, respectively.

Through the parsed data $D_{\text{Fusion}} = \{E_q, R_q\}$, we further extracts key information and links it to the knowledge graph. The knowledge construction process involves three key phases implemented through the OpenSPG framework¹ [26], [32], in which we use the *Custom Prompt* module² to integrate LLM-based knowledge extraction.

For entity recognition, we utilize the **ner.py** prompts within the **kag/builder/prompt/default** directory. We first define relevant entity types in the schema. Then, by adjusting the *example.input* and *example.output* in the **ner.py** prompts, we guide the LLM-based SchemaFreeExtractor to identify entities accurately.

In relationship extraction, the **triple.py** prompts play a crucial role. We define relationships in the schema and use the **triple_prompt** in the SchemaFreeExtractor. The *instruction* in **triple.py** ensures that the extracted Subject-Predicate-Object(SPO) triples are related to the entities in the *entity_list*, enabling effective relationship extraction.

Regarding attribute extraction, we rely on the entity standardization prompts in **std.py**. After entity recognition, the **std_prompt** in the SchemaFreeExtractor standardizes the entities and helps in extracting their attributes. We modify the *example.input*, *example.named_entities*, and *example.output* in **std.py** according to our data characteristics to optimize the attribute extraction process. Through these steps of customizing and applying OpenSPG's prompts, we achieve efficient knowledge extraction.

¹<https://github.com/OpenSPG/openspg>

²<https://openspg.yuque.com/>

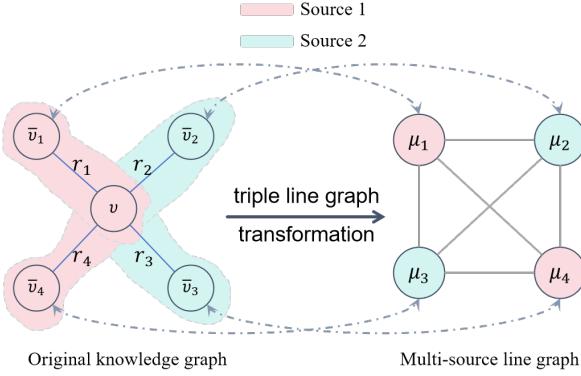


Fig. 4: Example of multi-source line graph transformation

The following formula describes the data extraction process:

$$KB = \sum_{D_i} (\{e_1, e_2, \dots, e_m\} \sqcup \{r_1, r_2, \dots, r_n\}) \quad (3)$$

C. Homologous Subgraph Matching

After the preliminary extraction of information, the next step is to identify the multi-source homologous data group set $\mathcal{SV}s$ and the isolated point set $\mathcal{LV}s$. This process begins by initializing the unvisited node set $\mathcal{U}_{\text{unvisited}} = \mathcal{V}$, while setting the homologous data group $\mathcal{SV}s = \emptyset$ and the isolated point set $\mathcal{LV}s = \emptyset$. By traversing all nodes and retrieving node information from various domains, for matched homologous data, construct the homologous node sg_i and its corresponding associated edge e_i , and add them to the homologous node set \mathcal{U}_{sg} and edge set \mathcal{E}_{sg} , respectively. After the traversal, add $(\mathcal{U}_{sg}, \mathcal{E}_{sg})$ to $\mathcal{SV}s$. If no homologous data is obtained after one round of traversal, add the node to the isolated point set $\mathcal{LV}s$. After the traversal is completed, the node will be removed from the $\mathcal{U}_{\text{unvisited}}$ set. The time complexity of homologous subgraph matching is $O(n \log n)$, where n is the number of nodes in the knowledge graph \mathcal{G} .

For each homologous subgraph in $\mathcal{SV}s$, homologous linear knowledge subgraph $sub\mathcal{SG}'_i$ is constructed by utilizing the homologous node set \mathcal{U}_{sg} and the homologous edge set \mathcal{E}_{sg} . Subsequently, all $sub\mathcal{SG}'_i$ and the isolated point set $\mathcal{LV}s$ are aggregated to obtain the homologous linear knowledge graph \mathcal{SG}' . It should be noted that \mathcal{SG}' is solely used for consistency checks and retrieval queries of homologous data; other types of queries still conduct operations on the original knowledge graph \mathcal{G} .

Here, we provide a simple example of a homologous triple line graph. As shown in Fig. 4, a homologous node is associated with 4 homologous data points. After transformation into a triple line graph, it forms a complete graph of order 4, indicating that the four triples are pairwise homologous.

D. Multi-level Confidence Computing

We define the candidate data from different domains obtained in a single retrieval as multi-source homologous data. These data have been extracted into a homologous line graph

for temporary storage. Although targeting the same query object, they often provide inconsistent reference answers. Considering the varying retrieval errors, the multi-level confidence calculation method is adopted in this framework. First, the confidence of individual homologous line graphs is calculated, followed by the confidence of each candidate node, to determine the final set of answer candidates.

1) *Graph-Level Confidence Computing*: In the first stage, a confidence calculation method based on mutual information entropy is introduced to assess the confidence of homologous line graphs. The core idea of this method is that if two nodes with the same attributes in a homologous line graph are close in content, their similarity is high, and thus their confidence is also high; conversely, if they are not, their confidence is low.

Let \mathcal{G} be a homologous line graph, and $\mathcal{N}(\mathcal{G})$ be the set of nodes in the graph. For any two nodes $v_i, v_j \in \mathcal{N}(\mathcal{G})$ with the same attributes, the similarity $S(v_i, v_j)$ between them is defined based on the calculation method of mutual information entropy. The mutual information entropy $I(v_i, v_j)$ measures the interdependence of the attribute content of the two nodes, and its calculation formula is:

$$I(v_i, v_j) = \sum_{x \in V_i} \sum_{y \in V_j} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (4)$$

where V_i and V_j are the sets of attribute values for nodes v_i and v_j , respectively, $p(x, y)$ is the joint probability distribution of v_i taking attribute value x and v_j taking attribute value y , and $p(x)$ and $p(y)$ are the marginal probability distributions of x and y , respectively.

The similarity $S(v_i, v_j)$ can be defined as the normalized form of mutual information entropy to ensure that its value lies within the interval $[0, 1]$:

$$S(v_i, v_j) = \frac{I(v_i, v_j)}{H(V_i) + H(V_j)} \quad (5)$$

where $H(V_i)$ and $H(V_j)$ are the entropies of the attribute value sets of nodes v_i and v_j , respectively, calculated as:

$$H(V) = - \sum_{x \in V} p(x) \log p(x) \quad (6)$$

Subsequently, the confidence $C(\mathcal{G})$ of the homologous line graph \mathcal{G} can be determined by calculating the average similarity $S(v_i, v_j)$ of all node pairs in the graph:

$$C(\mathcal{G}) = \frac{1}{|\mathcal{N}(\mathcal{G})|^2 - |\mathcal{N}(\mathcal{G})|} \sum_{v_i \in \mathcal{N}(\mathcal{G})} \sum_{\substack{v_j \in \mathcal{N}(\mathcal{G}) \\ j \neq i}} S(v_i, v_j) \quad (7)$$

where $|\mathcal{N}(\mathcal{G})|$ denotes the number of nodes in the graph. Notably, a homologous line graph exhibiting high confidence demonstrates that its constituent nodes maintain strong attribute-level consistency across their content representations.

2) *Node-Level Confidence Computing*: In the second phase, the confidence of individual node $C(v)$ is calculated, which takes into account the node's consistency, authority, and historical confidence. The following are the detailed calculation methods and formulas.

Algorithm 1 Multi-level Confidence Computing Algorithm

```

1: procedure CONFIDENCE_COMPUTING( $v, D$ )
2:    $S_n(v) \leftarrow$  Equation (8)
3:    $Auth_{LLM}(v) \leftarrow$  Equation (10)
4:    $Auth_{hist}(v) \leftarrow$  Equation (11)
5:    $A(v) \leftarrow$  Equation (9)
6:    $C(v) \leftarrow S_n(v) + A(v)$ 
7:   return  $C(v)$ 
8: end procedure
9: procedure MCC( $\mathcal{G}, Q, D$ )
10:   $\mathcal{SVs} \leftarrow \emptyset, \mathcal{LVs} \leftarrow \emptyset$ 
11:   $\mathcal{U}_{unvisited} \leftarrow V$ 
12:  while  $\mathcal{U}_{unvisited} \neq \emptyset$  do
13:     $v \leftarrow$  pop a node from  $\mathcal{U}_{unvisited}$ 
14:    for all  $D \in D$  do
15:      if  $v \in Data(Q, subSG_i)$  then
16:         $C(v) \leftarrow$  Confidence_Computing( $v, D$ )
17:        if  $C(v) > \theta$  then
18:           $\mathcal{U}_{sg} \leftarrow \mathcal{U}_{sg} \cup \{v\}$ 
19:           $\mathcal{E}_{sg} \leftarrow \mathcal{E}_{sg} \cup \{e_i\}$ 
20:        else
21:           $\mathcal{LVs} \leftarrow \mathcal{LVs} \cup \{v\}$ 
22:        end if
23:      end if
24:    end for
25:    if  $\mathcal{U}_{sg} \neq \emptyset$  then
26:       $\mathcal{SVs} \leftarrow \mathcal{SVs} \cup (\mathcal{U}_{sg}, \mathcal{E}_{sg})$ 
27:       $\mathcal{U}_{sg} \leftarrow \emptyset, \mathcal{E}_{sg} \leftarrow \emptyset$ 
28:    end if
29:  end while
30:  return  $\mathcal{SVs}, \mathcal{LVs}$ 
31: end procedure

```

a) *Node Consistency Score*: The node consistency score $S(v)$ reflects the consistency of the node across different data sources. We use mutual information entropy to calculate the similarity between node pairs, thereby assessing consistency. For a node v , its consistency score can be expressed as:

$$S_n(v) = \frac{1}{|N(v)|} \sum_{u \in N(v)} S(v, u) \quad (8)$$

where $N(v)$ is the set of nodes with the same attributes as node v , and $S(v, u)$ is the similarity between nodes v and u as defined in Equation 5.

b) *Node Authority Score*: Authority score is divided into two parts: the node's authority assessed by the LLM and the node's historical authority. This score reflects the importance and authenticity of the node. Additionally, we use an expert LLM to comprehensively evaluate the authority of the node. The node's authority score $A(v)$ can be calculated using the following formula:

$$A(v) = \alpha \cdot Auth_{LLM}(v) + (1 - \alpha) \cdot Auth_{hist}(v) \quad (9)$$

Algorithm 2 Multi-source Knowledge Line Graph Prompting

```

1: procedure MKLGP( $q$ )
2:    $E_q, R_q \leftarrow$  Logic Form Generation( $q$ )
3:    $D_q \leftarrow$  Multi Document Extraction( $V_q$ )
4:    $\mathcal{SG}' \leftarrow$  Prompt( $D_q$ )
5:    $\mathcal{SVs}, \mathcal{LVs} \leftarrow$  MCC( $\mathcal{SG}', q, D_q$ )
6:    $C_{nodes}, \mathcal{G}_A \leftarrow$  Prompt( $\mathcal{SVs}, \mathcal{LVs}$ )
7:    $Answer \leftarrow$  Generating Trustworthy Answers( $C_{nodes}, \mathcal{G}_A$ )
8:   return  $Answer$ 
9: end procedure

```

where α is a weight coefficient that balances the contributions of LLM-assessed authority and historical authority, satisfying $0 \leq \alpha \leq 1$.

Benefiting from the calculation idea of knowledge credibility in the PTCA [33], $Auth_{LLM}(v)$ is assessed by the global influence and local connection strength of the node. The LLM can comprehensively calculate the credibility of knowledge by integrating the association strength between entities, entity type information, and multi-step path information.

$$Autm_{LLM}(v) = \frac{1}{1 + e^{-\beta \cdot C_{LLM}(v)}} \quad (10)$$

where $C_{LLM}(v)$ is the authority score provided by the LLM for node v is the average value of all nodes' $C_{LLM}(v)$, and β is a parameter that controls the steepness of the scoring curve.

c) *Historical Authority*: $Auth_{hist}(v)$ is an authority score based on the node's historical data. Inspired by Zhu's work [34], we expect to use the credibility of historical data sources and current query-related data for incremental estimation.

$$Auth_{hist}(v) = \frac{\mathcal{H} \cdot Pr^h(D) + \sum_{v_p \in D_v[q]} Pr(v_p)}{\mathcal{H} + |Data(q, subSG'_i)|} \quad (11)$$

where \mathcal{H} is the number of entities provided by data source D for all historical queries, $Pr^h(D)$ is the historical credibility of data source D , $D_v[q]$ is the set of correct answers, and $Data(q, subSG'_i)$ is the query-related data obtained from the multi-source line subgraph.

Ultimately, we designed the multi-level confidence computing algorithm, MCC, to calculate the credibility of the data sources in the homologous subgraph, ensuring the quality of the knowledge graph embedded in the LLM. The algorithm is shown in Algorithm1.

It should be noted that the MCC algorithm does not directly provide the final graph confidence and node confidence; these values must be obtained through prompt to achieve the ultimate results.

E. Multi-source knowledge line graph prompting

We propose the Multi-source Knowledge Line Graph Prompting (MKLGP) algorithm for multi-source data retrieval. Given a user query q , LLM is firstly employed to extract the intent, entities, and relationships from q , and generates the corresponding logical relationships. The dataset then undergoes

multi-document filtering to derive text chunks, followed by constructing a Multi-source Line Graph (MLG) for knowledge aggregation. Further, it matches homogeneous subgraphs and utilizes the MCC algorithm to obtain a set of credible query nodes and isolated points $\mathcal{SV}_s, \mathcal{LV}_s$. Finally, by leveraging the prompt, the graph confidence is obtained, and the node confidence is calculated to enhance the credibility of the answer. The results are then embedded into the context of the LLM to generate a credible retrieval answer.

IV. EXPERIMENTS

This section will conduct experiments and performance analysis on the construction of homologous line graphs and the multi-level confidence calculation modules. Baseline methods will be compared with other SOTA multi-document retrieval QA methods, data fusion methods, and KBQA methods. Extensive experiments will be conducted to assess the robustness and efficiency of MultiRAG, which aims to answer the following questions.

- **Q1:** How does the retrieval recall performance of MultiRAG compare with other data fusion models and SOTA data retrieval models?
- **Q2:** What are the respective impacts of data sparsity and data inconsistency on the quality of retrieval recall?
- **Q3:** How effective are the two modules of MultiRAG individually?
- **Q4:** How is the performance of MultiRAG in multi-hop Q&A datasets after incorporating multi-level confidence calculation?
- **Q5:** What are the time costs of the various modules in MultiRAG?

A. Experimental Settings

a) **Datasets:** To validate the efficiency of multi-source line graph construction and its enhancement of retrieval performance, the article conducts multi-source data fusion experiments on four real-world benchmark datasets [35]–[37], as is shown in Table I. (1) The movie dataset comprises movie data collected from 13 sources. (2) The book dataset includes book data from 10 sources. (3) The flight dataset gathers information on over 1200 flights from 20 sources. (4) The stock dataset collects transaction data for 1000 stock symbols from 20 sources. In the experiments, we issue 100 queries for each of the four datasets to verify their retrieval efficiency.

It is noteworthy that the Movies dataset and the Flights dataset are relatively dense, while the Books dataset and the Stocks dataset are relatively sparse, which can impact the model's performance.

Additionally, to validate the robustness of the MultiRAG on complex Q&A datasets, we selected two multi-hop question answering datasets, HotpotQA [38] and 2WikiMultiHopQA [39]. Both datasets are constructed based on Wikipedia documents, allowing us to utilize a consistent document corpus and retriever to provide external references for LLMs. Considering the constraints of experimental costs, we conducted a

subsample analysis on 300 questions from the validation sets of each experimental dataset.

TABLE I: Statistics of the datasets preprocessed

Datasets	Data source	Sources	Entities	Relations	Queries
Movies	JSON(J)	4	19701	45790	
	KG(K)	5	100229	264709	100
	CSV(C)	4	70276	184657	
Books	JSON(J)	3	3392	2824	
	CSV(C)	3	2547	1812	100
	XML(X)	4	2054	1509	
Flights	CSV(C)	10	48672	100835	
	JSON(J)	10	41939	89339	100
Stocks	CSV(C)	10	7799	11169	
	JSON(J)	10	7759	10619	100

b) **Evaluation Metrics:** To assess effectiveness, we adopt the F1 score as the evaluation metric for the data fusion results, following previous experimental metrics [37], [40]–[42]. The F1 score is the harmonic mean of precision (P) and recall (R), calculated as follows:

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (12)$$

Furthermore, to evaluate the retrieval credibility of MKLGP Algorithm, we utilize the recall metric, specifically Recall@K, to assess performance at three distinct stages: before subgraph filtering, before node filtering, and after node filtering. In addition, we employ the query response time T (measured in seconds) as an evaluative metric to verify the efficiency of knowledge aggregation.

c) **Hyper-parameter Settings:** For all baselines, we carefully adjusted the parameters according to the characteristics of MultiRAG. All methods were implemented in a Python 3.10 and CUDA 11.6 environment. Except for the experiments using GPT-3.5-Turbo for CoT, the rest of the work utilized Llama3-8B-Instruct as the base model. For each different data format, after slicing into Chunks, we stored the slice numbers, data source locations, and transformed triple nodes in the multi-source line graph using JSON-LD format, thereby enabling simple cross-indexing.

For hyperparameter settings, the temperature parameter β was set to 0.5. The number of entities in historical queries was initialized to 50, the initial node confidence threshold was defined as 0.7, and the graph confidence threshold was set to 0.5. All experiments were conducted on a device equipped with an Intel(R) Core(TM) Ultra 9 185H 2.30GHz and 512GB of memory.

d) **Baseline Models:** To demonstrate the superiority of the MultiRAG method, we compare it with basic data fusion methods and SOTA methods, including the multi-document question-answering methods and knowledge base question-answering methods.

Thanks to Zhu's work³ [34], we compare with the following baseline methods:

³<https://github.com/JunHao-Zhu/FusionQuery>

TABLE II: Comparison with baseline methods and SOTA methods for multi-source knowledge fusion

Datasets	Data source	Data Fusion Methods (Baseline)				SOTA Methods								Our Method	
		TF		LTM		IR-CoT		MDQA		ChatKBQA		FusionQuery		MCC	
		F1/%	Time/s	F1/%	Time/s	F1/%	Time/s	F1/%	Time/s	F1/%	Time/s	F1/%	Time/s	F1/%	Time/s
Movies	J/K	37.1	9717	41.4	1995	43.2	1567	46.2	1588	45.1	3809	53.2	122.4	<u>52.6</u>	98.3
	J/C	41.9	7214	42.9	1884	45.0	1399	44.5	1360	42.7	3246	52.7	183.1	54.3	75.1
	K/C	37.8	2199	41.2	1576	37.6	1014	45.2	987	40.4	2027	42.5	141.0	49.1	86.0
	J/K/C	36.6	11225	40.8	2346	41.5	2551	49.8	2264	44.7	5151	53.6	137.8	54.8	<u>157</u>
Books	J/C	40.2	1017	42.4	195.3s	35.2	147.6	55.7	124.2	56.1	165.0	58.5	22.7	63.5	13.66
	J/X	35.5	1070	35.6	277.7	36.1	178.7	55.1	115.6	54.7	200.1	57.9	20.6	63.1	13.78
	C/X	43.0	1033	44.1	232.6	42.6	184.5	57.2	115.6	55.6	201.4	<u>60.3</u>	21.5	64.2	13.54
	J/C/X	37.3	2304	41.0	413.2	40.4	342.6	56.4	222.6	57.1	394.1	59.1	47.0	66.8	27.4
Flights	C/J	27.3	6049	79.1	14786	58.3	214.0	76.5	360	76.8	376	74.2	20.2	<u>74.9</u>	<u>80</u>
Stocks	C/J	68.4	2.30	19.2	1337	64.8	53.3	65.2	78.4	64.0	88.9	68.0	0.33	78.6	<u>12.1</u>

* The F1 score is for Q1 and time is for Q5.

* Bold represents the optimal metrics, while underlined text indicates the sub-optimal metrics. The same applies to the following text.

- 1) **TruthFinder(TF)** [37]: the classic iterative data fusion method.
- 2) **LTM** [42]: the probabilistic data fusion method.
- 3) **CoT** [43] is a foundational approach that involves step-by-step reasoning to reach a conclusion, we use GPT-3.5-Turbo as the base model.
- 4) **Standard RAG** [2] is a method that combines the strengths of retrieval and generation models to answer questions.

Moreover, we also summarize these SOTA methods below:

- **IRCoT** [44] is an advanced method that refines the reasoning process through iterative retrieval.
- **ChatKBQA** [45] is a conversational interface-based method for knowledge base question answering.
- **MDQA** [46] is a method designed to extract answers from multiple documents effectively.
- **FusionQuery** [34] is a SOTA method based on the efficient on-demand fusion query framework.
- **RQ-RAG** [47] is a method that integrates external documents and optimizes the query process to handle complex queries.
- **MetaRAG** [9] is a method that employs metacognitive strategies to enhance the retrieval process.

e) **Dataset Preprocessing**: To better align the datasets with real-world application scenarios and to demonstrate the applicability of the proposed method to multi-source data, we have split and reconstructed the four datasets into three categories of data formats: tabular data (structured data), nested JSON data (semi-structured data), and XML data (semi-structured data), stored respectively in csv, json, and xml file formats. We also retained some data directly stored in KG format. Table I displays the detailed statistics after the dataset division.

B. Evaluation of Multi-source Knowledge Aggregation (MKA)

Q1: How does the retrieval recall performance of MultiRAG compare with other data fusion models and SOTA

data retrieval models?

To validate the effectiveness of the multi-source knowledge aggregation module (MKA) in MultiRAG, we assess it using F1 scores and query times across four multi-source query datasets. By substituting the fusion query algorithm with different baseline models and SOTA models, multiple sets of experimental results are obtained to evaluate its performance in multi-domain querying. Table II summarizes the data querying performance of MKLG and baselines on the four datasets; Q1 focuses solely on the F1 scores of the methods, which includes four data fusion methods and three SOTA methods that support data fusion.

Table II demonstrates that the MCC module outperforms all comparative models across four datasets. Experimental results indicate that it achieves an F1 score that is more than 10% higher than the best baseline data fusion model and obtains superior performance compared to other baselines. The MV method performs poorly on all datasets because it can only return a single answer for a query, which fails to accommodate the common scenario where a query has multiple return values. For instance, a movie or a book typically has multiple directors or authors. However, the majority of methods show significantly better performance on the Movies and Flights datasets than on the Books and Stocks datasets. This is because the Movies and Flights datasets are inherently denser, and previous SOTA models can match or outperform our approach in situations where knowledge is abundant, which is acceptable. In contrast, on the more sparse Books and Stocks datasets, our method achieves an average improvement of more than 10% over SOTA methods.

Q2: What are the respective impacts of data sparsity and data inconsistency on the quality of retrieval recall?

MultiRAG demonstrates good robustness in scenarios of varying data sparsity and inconsistency. To validate it, we conducted experiments from the following two perspectives. 1) Sparsity of multi-source data: We applied 30%, 50%, and 70% random relationship masking to four pre-processed datasets,

TABLE III: Ablation experiments of multi-source knowledge aggregation(MKA) and multi-level confidence computing(MCC)

Datasets	Source	MultiRAG			w/o MKA			w/o Graph Level			w/o Node Level			w/o MCC		
		F1/%	QT/s	PT/s	F1/%	QT/s	PT/s	F1/%	QT/s	PT/s	F1/%	QT/s	PT/s	F1/%	QT/s	PT/s
Movies	J/K	52.6	25.7	62.64	48.2	2783	62.64	45.3	50.1	58.2	38.7	21.3	0.31	31.6	25.7	0.28
	J/C	54.3	12.7	61.36	49.1	1882	61.36	46.8	28.9	57.4	40.2	10.5	0.29	30.5	12.7	0.29
	K/C	49.1	31.6	64.40	45.5	4233	64.40	42.7	65.3	61.8	35.9	28.4	-0.27	33.1	31.6	-0.29
	J/K/C	54.8	39.2	60.8	47.5	4437	60.8	48.1	75.6	56.2	41.5	35.8	0.30	34.7	39.2	0.32
Books	J/C	63.5	1.19	2.47	57.1	11.9	2.47	55.2	4.7	2.12	49.8	0.92	0.18	43.4	1.19	0.22
	J/X	63.1	1.22	2.56	59.3	11.7	2.62	54.7	5.1	2.24	48.3	0.89	0.19	42.6	1.22	0.22
	C/X	64.2	1.16	2.38	55.3	8.39	2.38	53.9	3.9	2.05	47.1	0.85	0.16	41.0	1.16	0.17
	J/C/X	66.8	1.31	3.07	57.2	15.8	3.08	59.4	6.3	2.89	52.7	1.12	0.21	36.4	1.31	0.20
Flights	C/J	74.9	29.8	109.9	72.2	NAN	109.9	68.3	142.7	98.5	61.4	25.3	0.85	52.1	29.8	1.07
Stocks	C/J	78.6	2.72	5.36	69.6	450.8	5.36	72.1	8.9	4.12	65.3	1.98	0.15	45.4	2.72	0.17

TABLE IV: Performance comparison on HotpotQA and 2WikiMultiHopQA datasets

Method	HotpotQA		2WikiMultiHopQA	
	Precision	Recall@5	Precision	Recall@5
Standard RAG	34.1	33.5	25.6	26.2
GPT-3.5-Turbo+CoT	33.9	47.2	35.0	45.1
IRCoT	41.6	41.2	42.3	40.9
ChatKBQA	47.8	42.1	46.5	43.7
MDQA	48.6	52.5	44.1	45.8
RQ-RAG	51.6	49.3	45.3	44.6
MetaRAG	51.1	49.9	50.7	52.2
MultiRAG	59.3	62.7	55.7	61.2

making the connections between data sparser while ensuring that the query answers are still retrievable. 2) Consistency of multi-source data: We added 30%, 50%, and 70% of triple increments (the new triples are copies of the original triples) to the four pre-processed datasets, and completely shuffled the relationship edges of the added triples to disrupt the consistency of multi-source data. Subsequently, we employed MultiRAG to experiment with datasets under both perturbation schemes.

Firstly, to address data sparsity, we conducted experiments on MultiRAG (Ours) and ChatKBQA (SOTA). The experimental results demonstrate that MultiRAG exhibits significant robustness when faced with the challenge of data sparsity.

Specifically, after applying 30%, 50%, and 70% relationship masking, the F1 score of MultiRAG on the Books dataset only dropped from 66.8% to 60.0%. On the Stocks dataset, its F1 score decreased from 78.6% to 71.0%, which have been shown in Fig.5b and Fig.5d. These moderate decreases indicate that MultiRAG can effectively maintain its performance even when a substantial number of relationships are masked.

In contrast, ChatKBQA's performance decline under the same conditions is more significant. On the Books dataset,

ChatKBQA's F1 score dropped from 59.1% to 53.0%, and on the Stocks dataset, its F1 score decreased from 68.0% to 62.0%. This outcome reveals the challenges ChatKBQA faces when dealing with sparse data, especially when a large number of data connections are masked, significantly impacting its performance.

Next, we conducted robustness experiments on multi-source data consistency. We perturbed the Books and Stocks datasets to varying degrees to test the performance changes of MultiRAG and ChatKBQA when data consistency is disrupted. The experimental results show that MultiRAG demonstrates excellent robustness in the face of data consistency disruption, while ChatKBQA's performance declines rapidly under perturbation.

Specifically, as is shown in Fig. 5a, on the Movies dataset, we added 30%, 50%, and 70% triple increments to the original dataset and randomized the relationship edges of the added triples. The results show that MultiRAG's F1 score slightly decreased from 54.8% to 52.1%, 51.5%, and 49.9%, while ChatKBQA's F1 score significantly dropped from 53.6% to 51.6%, 47.2%, and 40.8%. On the Flights dataset shown in Fig. 5c, we performed the same perturbation operations, and MultiRAG's F1 score slightly decreased from 74.9% to 73.4%, 72.9%, and 71.4%, while ChatKBQA's F1 score substantially dropped from 74.2% to 69.7%, 64.3%, and 55.8%.

These results indicate that even when data consistency is severely compromised, MultiRAG can still maintain a high level of performance stability, whereas ChatKBQA's performance is more sensitive to disruptions in data consistency.

C. Evaluation of Multi-level Confidence Computing

Calculating the confidence of subgraphs and nodes to filter trustworthy answers is of significant demand in critical domains such as finance and law. Considering the high temporal and spatial overhead of directly calculating the confidence of all nodes, we draw inspiration from the workflow of recommendation systems, mimicking the process of coarse and fine ranking, and adopt the multi-level confidence computing method to filter credible nodes and enhance retrieval perfor-

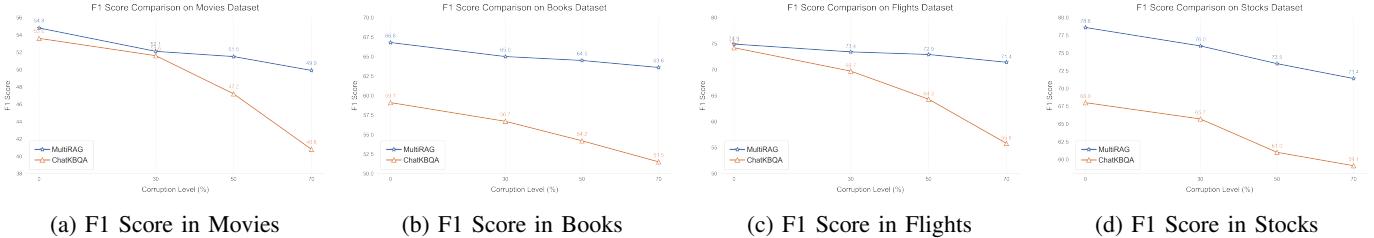


Fig. 5: Experimental results of Q2, where (a) and (b) display the multi-source data sparsity experiments, and (c) and (d) display the multi-source data consistency experiments.

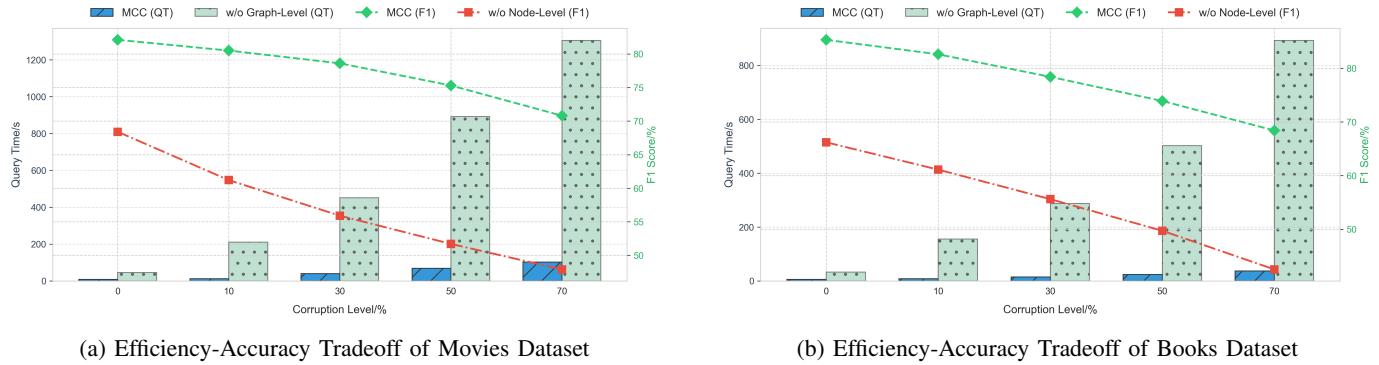


Fig. 6: F1 score and Query Time of Movies and Books with corruption level 0%, 10%, 30%, 50%, 70% in different sources

mance. Calculating the credibility of homologous subgraphs allows us to preliminarily determine whether the subgraphs containing answers can generate highly credible answers. For subgraphs with low confidence, more nodes need to be extracted to ensure the robustness of the overall retrieval; for subgraphs with high confidence, only 1-2 nodes are required to generate the correct answer.

Q3: How effective are the two modules of MultiRAG individually?

a) Ablation Study on Component Effectiveness: The MKA module achieves significant efficiency-accuracy synergy through its MLG architecture. As shown in Table III, MLG construction introduces modest preprocessing time (12.7s-39.2s) while delivering 10-100x query acceleration. Specifically, the flight dataset shows QT reduction from computational infeasibility (marked NAN) to 29.8s through MLG’s compact structure. Concurrently, MKA sustains consistent accuracy improvements. Removing MKA causes F1 drops of 7.3% on Movies and 9.6% on Books, demonstrating MLG’s effectiveness in connecting fragmented knowledge across sources.

The MCC module exhibits more significant effects on performance and hallucination control. Disabling MCC causes drastic F1 degradation of 20.1% on Movies and 33.2% on Stocks, with PT values indicating increased hallucination risks. This validates MCC’s critical role in eliminating unreliable information through hierarchical confidence computation.

b) Hierarchical Analysis of MCC: Stratified ablation reveals the complementary roles of graph-level and node-level computations. For Movies (J/K/C configuration), removing

graph-level filtering reduces F1 to 48.1% (+13.4% vs MCC-disabled) with QT increasing to 75.6s (+93% vs full framework). Conversely, disabling node-level computation yields 41.5% F1 (+6.8% vs baseline), showing graph-level filtering alone cannot resolve local conflicts. The complete MCC framework achieves 54.8% F1 by synergistically combining both layers.

Error analysis shows distinct failure patterns: 38.7% errors under graph-level removal (Movies J/K) stem from cross-source inconsistencies, while 52.7% failures with node-level removal (Books J/C/X) originate from local authority issues. This confirms the functional specialization—graph-level ensures global consistency, node-level verifies local credibility.

Fig.7 demonstrates that an optimal balance between efficiency and accuracy is achieved at $\alpha = 0.5$, where the hybrid weighting of LLM-assessed authority and historical authority peaks with an F1 score of 67.7% and balanced query time. Specifically, increasing α towards 1.0, which emphasizes the LLM, reduces query time from 83.2 seconds ($\alpha = 0.0$) to 51.8 seconds ($\alpha = 1.0$) by minimizing historical data validation. Conversely, the F1 score follows a non-monotonic pattern, reaching its maximum at $\alpha = 0.5$ before declining as reliance on either the LLM or historical data becomes excessive. This equilibrium leverages the LLM’s contextual adaptability (Auth_{LLM}) while maintaining the stability of expert systems ($\text{Auth}_{\text{hist}}$), as evidenced by a 62.4% reduction in errors during ablation studies when both components are utilized. By avoiding complete dependence on the LLM ($\alpha \neq 1.0$) and integrating probabilistic LLM inferences with deterministic historical patterns through multi-level confidence computing (Eq.9), the

TABLE V: Case Study

Query: "What is the real-time status of Air China flight CA981 from Beijing Capital International Airport (PEK) to New York John F. Kennedy Airport (JFK)?"	
Data Sources	
Structured	CA981, PEK, JFK, Delayed, 2024-10-01 14:30
Semi-structured	{"flight": "CA981", "delay_reason": "Weather", "source": "AirChina"}
Unstructured	"Typhoon Haikui impacts PEK departures after 14:00."
MKA Module	Structured parsing: Flight attributes mapping LLM extraction: (CA981, DelayReason, Typhoon) @0.87
MLG Subgraph	<pre> graph TD ForumUser123[ForumUser123
(Source)] <-- Source --> OnTime[On-time
(User Claim)] ForumUser123 <-- Conflict --> Delayed[Delayed
(Status)] Typhoon[Typhoon
(Cause)] -- Reason --> Delayed Typhoon -- Effective --> ImpactTime[After 14:00+
(Impact Time)] CA981[CA981
(Flight)] -- Status --> Delayed CA981 -- Departure --> PEK[PEK
(Origin)] CA981 -- Destination --> JFK[JFK
(Destination)] PEK --> Delayed JFK --> Delayed AirChinaAPP[AirChina APP
(Source)] -- Source --> Delayed </pre> <p>The diagram illustrates the MLG Subgraph. It consists of several nodes and directed edges representing relationships between them. Nodes include 'ForumUser123 (Source)', 'On-time (User Claim)', 'Typhoon (Cause)', 'After 14:00+ (Impact Time)', 'CA981 (Flight)', 'Delayed (Status)', 'PEK (Origin)', 'JFK (Destination)', and 'AirChina APP (Source)'. Relationships are labeled with arrows: 'Source' from ForumUser123 to On-time and from AirChina APP to Delayed; 'Conflict' from ForumUser123 to Delayed; 'Reason' from Typhoon to Delayed; 'Effective' from Typhoon to ImpactTime; 'Status' from CA981 to Delayed; 'Departure' from CA981 to PEK; 'Destination' from CA981 to JFK; and 'Source' from PEK and JFK to Delayed.</p>
MCC Module	With GCC: Graph confidence=0.71 (Threshold=0.5), Filtered: ForumUser123 (0.47) Without GCC: Unfiltered conflict=2 subgraphs
LLM Context	Trusted: CA981.Status=Delayed (0.89), DelayReason=Typhoon (0.85) Conflicts: ForumUser123:On-time (0.47), WeatherAPI:Clear (0.52)
Final Answer	Correct: "CA981 delayed until after 14:30 due to typhoon" Hallucinated: "CA981 [on-time with possible delay after 14:30]"

methodology enhances robustness against data sparsity and noise, particularly in the Books and Stocks datasets.

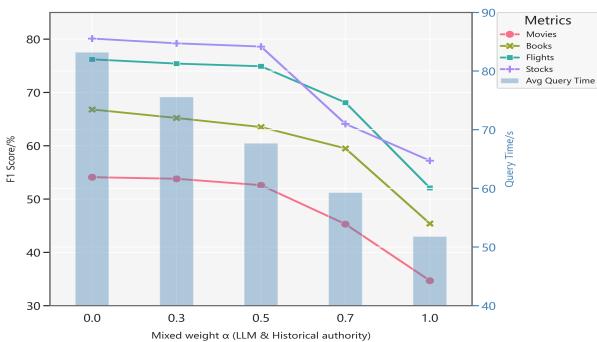


Fig. 7: Influence of hyperparameter α on multi-source retrieval

Q4: How is the performance of MultiRAG in multi-hop Q&A datasets after incorporating multi-level confidence calculation?

To assess the validity of the multi-level confidence computing method in reducing hallucinations generated by large models and enhancing the credibility of Q&A systems, we compare the *Recall@5* scores of different methods on the HotpotQA and 2WikiMultiHopQA datasets.

The outcome of Table IV indicates that the multi-level confidence computing method not only demonstrates a higher average *Recall@5* score but also maintains a lower standard deviation compared to traditional methods. This suggests that the multi-level confidence computing method is more consistent in its performance across different queries, leading to fewer hallucinations and more reliable Q&A responses. The

lower standard deviation is a testament to the robustness of the mechanism in handling the variability in data and the complexity of the queries.

Furthermore, we performed a detailed error analysis to identify the types and frequency of hallucinations in the responses generated by the different methods. The results showed that the multi-level confidence computing method significantly reduced the frequency of hallucinations, particularly in the cases where the context was ambiguous or the information was not readily available in the knowledge base.

Q5: What are the time costs of the two modules in MultiRAG?

Intuitively, MLG aggregates homologous data from several sources, ensuring the density of the retrieval subgraphs without the need to traverse and store an excessive number of invalid nodes, thereby significantly reducing the time cost associated with traversing and querying in traditional knowledge graphs.

Furthermore, although the SOTA methods are not specifically tailored for low-resource, high-noise data scenarios, they still exhibit considerable robustness and retrieval performance in such environments. Both the MDQA and ChatKBQA models employ LLM-based data retrieval approaches, with the primary temporal and spatial overheads focusing on token consumption and LLM-based searching.

In contrast, MultiRAG concentrates its overhead on the construction of the MLG. While in the original context of the MLG, construction times are often within seconds and highly efficient, the introduction of an LLM still incurs additional temporal costs due to text generation, which remains acceptable. Ultimately, these methods all demonstrate satisfactory retrieval performance; however, due to the inherent noise in the

datasets, improvements in the accuracy of question-answering are somewhat limited.

D. Case Study

MultiRAG’s effectiveness in multi-source integration is demonstrated through a real-world flight status query for “CA981 from Beijing to New York”. As detailed in Table V, case study exemplifies MultiRAG’s unique strength in transforming fragmented, conflicting inputs into trustworthy answers through systematic source weighting and consensus modeling.

Firstly, MultiRAG integrated three data formats: structured departure schedules, semi-structured delay codes from airline systems, and unstructured weather alerts. The MKA module extracted key relationships (flight-delay-typhoon) with a confidence score of 0.87. Subsequently, the MCC module resolved conflicts through hierarchical verification by filtering out low-reliability sources, such as user forums (confidence score of 0.47), while prioritizing data from airlines (confidence score of 0.89) and weather reports. This dual-layer validation—combining automated threshold checks (graph confidence of 0.71) with LLM-simulated expert reasoning—enabled the precise reconciliation of contradictory departure time claims. Ultimately, the system generated the verified conclusion, “Delayed until after 14:30 due to typhoon,” while suppressing the inconsistent “on-time” report.

E. Restrictive Analysis

Lastly but not least, we acknowledge several limitations inherent in our current framework.

- 1) Lack of optimization of text chunk segmentation.
- 2) Reliance on LLM-based expert evaluation, which may introduce potential security vulnerabilities.
- 3) Focuses on eliminating factual hallucinations but lacks handling of symbolic hallucinations.

V. RELATED WORK

A. Graph-Structured Approaches for Hallucination Mitigation

Recent advancements have demonstrated unique advantages of graph structures in mitigating hallucinations within RAG systems. MetaRAG [9] establishes knowledge association verification through meta-cognitive graph reasoning paths, enhancing self-correction mechanisms in multi-hop QA. Graph-CoT [48] innovatively leverages Graph Neural Networks to establish bidirectional connections between KGs and the latent space of LLMs. In result, it reduces factual inconsistencies by 37% on KGQA benchmarks. Inspired by neurobiology, HippoRAG [23] constructs offline memory graphs with a neural indexing mechanism, decreasing retrieval latency to one-eighth of traditional methods. While ToG 2.0 [25] further advances this field by introducing a graph-context co-retrieval framework that dynamically balances structured and unstructured evidence, resulting in a 29% reduction in hallucination rates compared to unimodal approaches.

Unlike prior approaches that primarily focus on unimodal confidence calculations, MultiRAG achieves superior hallucination mitigation through the adaptive filtering of conflicting subgraphs (GCC module) while maintaining multi-domain logical associations via its novel knowledge aggregation mechanism (MKA module).

B. Heterogeneous Graph Fusion for RAG

The fusion of multi-source heterogeneous data relies on advanced graph representation techniques. FusionQuery [34] enhances cross-domain retrieval precision by integrating heterogeneous graphs and computing dynamic credibility evaluations. The Triple Line Graph [31] addresses the challenge of knowledge fragmentation by systematically aggregating cross-domain relationships, leading to Multi-source Line Graph proposed in this paper. Additionally, leveraging the structured representation advantages of KAG [26] in knowledge-guided retrieval, we achieve a unified representation approach for multi-source KGs, underscoring the importance of heterogeneous graph fusion in real-world applications.

C. Hallucination Benchmark and Confidence-Aware Computing

The evaluation of hallucinations in LLMs and associated confidence calculation methods are crucial for mitigating hallucinations. HaluEval [49] offers 5,000 annotated samples across five error categories, but lacks granularity for relational hallucinations. RefChecker [50] implements triple decomposition for fine-grained detection, improving precision by 26.1% over sentence-level methods. RAGTruth [51] contains nearly 18,000 RAG-generated responses with detailed manual annotations including word-level hallucination intensities. However, diverse and complex data sources continue to challenge existing evaluation frameworks.

VI. CONCLUSION

In this work, we introduce MultiRAG, a framework designed to mitigate hallucination in multi-source knowledge-augmented generation. To address hallucinations arising from data sparsity and inconsistency, we propose two key innovations: multi-source knowledge aggregation and multi-level confidence calculation. The introduction of multi-source line graphs enables efficient cross-domain data aggregation, enhancing knowledge connectivity and retrieval performance. Meanwhile, our multi-level confidence computing module adaptively filter out low-quality subgraphs and unreliable nodes. Future work will explore more challenging aspects of hallucination mitigation, particularly in multimodal retrieval and ultra-long text reasoning, to better adapt generative retrieval systems to real-world, open multi-source environments.

VII. ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (62176185, U23B2057), and the “14th Five-Year Plan” Civil Aerospace Pre-research Project of China (D020101).

REFERENCES

- [1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.
- [2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [3] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, “Retrieval augmented language model pre-training,” in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.
- [4] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, “Retrieval-augmented generation for large language models: A survey,” *arXiv preprint arXiv:2312.10997*, vol. 2, 2023.
- [5] G. Izacard and E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” *arXiv preprint arXiv:2007.01282*, 2020.
- [6] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, “Few-shot learning with retrieval augmented language models,” *arXiv preprint arXiv:2208.03299*, vol. 2, no. 3, 2022.
- [7] Z. Jiang, L. Gao, J. Araki, H. Ding, Z. Wang, J. Callan, and G. Neubig, “Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer,” *arXiv preprint arXiv:2212.02027*, 2022.
- [8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [9] Y. Zhou, Z. Liu, J. Jin, J.-Y. Nie, and Z. Dou, “Metacognitive retrieval-augmented large language models,” in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 1453–1463.
- [10] H. Zeng, C. Luo, B. Jin, S. M. Sarwar, T. Wei, and H. Zamani, “Scalable and effective generative information retrieval,” in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 1441–1452.
- [11] W. Wu, H. Yin, N. Wang, M. Xu, X. Zhao, Z. Yin, Y. Liu, H. Wang, Y. Ding, and B. Li, “A cross-domain heterogeneous data query framework via collaboration of large language models and knowledge graphs,” *Journal of Computer Research and Development*, vol. 62, no. 3, pp. 605–619, 2025.
- [12] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, “Unifying large language models and knowledge graphs: A roadmap,” *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [13] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, “Reasoning on graphs: Faithful and interpretable large language model reasoning,” *arXiv preprint arXiv:2310.01061*, 2023.
- [14] J. Wang, K. Sun, L. Luo, W. Wei, Y. Hu, A. W.-C. Liew, S. Pan, and B. Yin, “Large language models-guided dynamic adaptation for temporal knowledge graph reasoning,” *arXiv preprint arXiv:2405.14170*, 2024.
- [15] Q. Sun, K. Huang, X. Yang, R. Tong, K. Zhang, and S. Poria, “Consistency guided knowledge retrieval and denoising in llms for zero-shot document-level relation triplet extraction,” in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 4407–4416.
- [16] M. Zamiri, Y. Qiang, F. Nikolaev, D. Zhu, and A. Kotov, “Benchmark and neural architecture for conversational entity retrieval from a knowledge graph,” in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 1519–1528.
- [17] Y. Li, G. Zang, C. Song, X. Yuan, and T. Ge, “Leveraging semantic information for enhanced community search in heterogeneous graphs,” *Data Science and Engineering*, vol. 9, no. 2, pp. 220–237, 2024.
- [18] Y. Hu, C. Chen, B. Deng, Y. Lai, H. Lin, Z. Zheng, and J. Bian, “Decoupling anomaly discrimination and representation learning: self-supervised learning for anomaly detection on attributed graph,” *Data Science and Engineering*, vol. 9, no. 3, pp. 264–277, 2024.
- [19] Z. Li, X. Wang, J. Zhao, W. Guo, and J. Li, “Hycube: Efficient knowledge hypergraph 3d circular convolutional embedding,” *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- [20] Z. Li, C. Wang, X. Wang, Z. Chen, and J. Li, “Hje: joint convolutional representation learning for knowledge hypergraph completion,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 8, pp. 3879–3892, 2024.
- [21] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, “Reasoning on graphs: Faithful and interpretable large language model reasoning,” *arXiv preprint arXiv:2310.01061*, 2023.
- [22] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Pruitt, and J. Larson, “From local to global: A graph rag approach to query-focused summarization,” *arXiv preprint arXiv:2404.16130*, 2024.
- [23] B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, “Hipporag: Neu-robiologically inspired long-term memory for large language models,” *arXiv preprint arXiv:2405.14831*, 2024.
- [24] C. Mavromatis and G. Karypis, “Gnn-rag: Graph neural retrieval for large language model reasoning,” *arXiv preprint arXiv:2405.20139*, 2024.
- [25] S. Ma, C. Xu, X. Jiang, M. Li, H. Qu, C. Yang, J. Mao, and J. Guo, “Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation,” *arXiv preprint arXiv:2407.10805*, 2024.
- [26] L. Liang, M. Sun, Z. Gui, Z. Zhu, Z. Jiang, L. Zhong, Y. Qu, P. Zhao, Z. Bo, J. Yang *et al.*, “Kag: Boosting llms in professional domains via knowledge augmented generation,” *arXiv preprint arXiv:2409.13731*, 2024.
- [27] W. Ding, J. Li, L. Luo, and Y. Qu, “Enhancing complex question answering over knowledge graphs through evidence pattern retrieval,” in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 2106–2115.
- [28] X. Wang, Z. Chen, H. Wang, Z. Li, W. Guo *et al.*, “Large language model enhanced knowledge representation learning: A survey,” *arXiv preprint arXiv:2407.00936*, 2024.
- [29] Y. Gao, Y. Xiong, W. Wu, Z. Huang, B. Li, and H. Wang, “U-niah: Unified rag and llm evaluation for long context needle-in-a-haystack,” *arXiv preprint arXiv:2503.00353*, 2025.
- [30] F. Wang, X. Wan, R. Sun, J. Chen, and S. Ö. Arik, “Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models,” *arXiv preprint arXiv:2410.07176*, 2024.
- [31] V. Fionda and G. Pirrò, “Learning triple embeddings from knowledge graphs,” in *proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 3874–3881.
- [32] P. Yi, L. Liang, D. Zhang, Y. Chen, J. Zhu, X. Liu, K. Tang, J. Chen, H. Lin, L. Qiu, and J. Zhou, “Kgfabric: A scalable knowledge graph warehouse for enterprise data interconnection,” *Proc. VLDB Endow.*, vol. 17, no. 12, p. 3841–3854, Aug. 2024.
- [33] X. ZHANG, W. SUN, and H. WANG, “Evaluation of knowledge credibility based on knowledge representation learning,” *Computer Engineering*, vol. 47, no. 7, pp. 44–54, 2021.
- [34] J. Zhu, Y. Mao, L. Chen, C. Ge, Z. Wei, and Y. Gao, “Fusionquery: On-demand fusion queries over multi-source heterogeneous data,” *Proceedings of the VLDB Endowment*, vol. 17, no. 6, pp. 1337–1349, 2024.
- [35] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, “Truth finding on the deep web: Is the problem solved?” *arXiv preprint arXiv:1503.00303*, 2015.
- [36] X. L. Dong, L. Berti-Equille, and D. Srivastava, “Integrating conflicting data: the role of source dependence,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 550–561, 2009.
- [37] X. Yin, J. Han, and P. S. Yu, “Truth discovery with multiple conflicting information providers on the web,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 1048–1052.
- [38] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, “Hotpotqa: A dataset for diverse, explainable multi-hop question answering,” *arXiv preprint arXiv:1809.09600*, 2018.
- [39] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, “Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps,” *arXiv preprint arXiv:2011.01060*, 2020.
- [40] X. Lin and L. Chen, “Domain-aware multi-truth discovery from conflicting sources,” *Proceedings of the VLDB Endowment*, vol. 11, no. 5, pp. 635–647, 2018.
- [41] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, and T. Derr, “Knowledge graph prompting for multi-document question answering,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 206–19 214.
- [42] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han, “A bayesian approach to discovering truth from conflicting sources for data integration,” *arXiv preprint arXiv:1203.0058*, 2012.
- [43] 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,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [44] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, “Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions,” *ArXiv*, vol. abs/2212.10509, 2022.
 - [45] H. Luo, Z. Tang, S. Peng, Y. Guo, W. Zhang, C. Ma, G. Dong, M. Song, W. Lin *et al.*, “Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models,” *arXiv preprint arXiv:2310.08975*, 2023.
 - [46] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, and T. Derr, “Knowledge graph prompting for multi-document question answering,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 206–19 214.
 - [47] C.-M. Chan, C. Xu, R. Yuan, H. Luo, W. Xue, Y. Guo, and J. Fu, “Rq-rag: Learning to refine queries for retrieval augmented generation,” *arXiv preprint arXiv:2404.00610*, 2024.
 - [48] B. Jin, C. Xie, J. Zhang, K. K. Roy, Y. Zhang, Z. Li, R. Li, X. Tang, S. Wang, Y. Meng, and J. Han, “Graph chain-of-thought: Augmenting large language models by reasoning on graphs,” in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 163–184.
 - [49] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, “Halueval: A large-scale hallucination evaluation benchmark for large language models,” *arXiv preprint arXiv:2305.11747*, 2023.
 - [50] X. Hu, D. Ru, L. Qiu, Q. Guo, T. Zhang, Y. Xu, Y. Luo, P. Liu, Y. Zhang, and Z. Zhang, “Refchecker: Reference-based fine-grained hallucination checker and benchmark for large language models,” *arXiv preprint arXiv:2405.14486*, 2024.
 - [51] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang, “RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 10 862–10 878.