KG-IRAG: A Knowledge Graph-Based Iterative Retrieval-Augmented Generation Framework for Temporal Reasoning

Ruiyi Yang[1] **Hao Xue**[1] **Imran Razzak**[2,1] **Hakim Hacid**[3] **Flora D. Salim**[1]
[1]University of New South Wales, Sydney, NSW, Austalia
[2]Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[3]Technology Innovation Institute, Abu Dhabi, UAE
{ruiyi.yang, hao.xue1, flora.salim}@unsw.edu.au
imran.razzak@mbzuai.ac.ae, hakim.hacid@tii.ae

Abstract

Graph Retrieval-Augmented Generation (GraphRAG) has proven highly effective in enhancing the performance of Large Language Models (LLMs) on tasks that require external knowledge. By leveraging Knowledge Graphs (KGs), GraphRAG improves information retrieval for complex reasoning tasks, providing more precise and comprehensive retrieval and generating more accurate responses to QAs. However, most RAG methods fall short in addressing multi-step reasoning, particularly when both information extraction and inference are necessary. To address this limitation, this paper presents **Knowledge Graph-Based Iterative Retrieval-Augmented Generation (KG-IRAG)**, a novel framework that integrates KGs with iterative reasoning to improve LLMs' ability to handle queries involving temporal and logical dependencies. Through iterative retrieval steps, KG-IRAG incrementally gathers relevant data from external KGs, enabling step-by-step reasoning. The proposed approach is particularly suited for scenarios where reasoning is required alongside dynamic temporal data extraction, such as determining optimal travel times based on weather conditions or traffic patterns. Experimental results show that KG-IRAG improves accuracy in complex reasoning tasks by effectively integrating external knowledge with iterative, logic-based retrieval. Additionally, three new datasets: *weatherQA-Irish*, *weatherQA-Sydney*, and *trafficQA-TFNSW*, are formed to evaluate KG-IRAG's performance, demonstrating its potential beyond traditional RAG applications.

1 Introduction

Knowledge Graphs (KGs) represent entities, their attributes, and relationships in a structured form, often employed to facilitate information retrieval, recommendation systems, and question answering. By encoding real-world facts as triples (e.g., (Sydney Opera House-[located in]-Sydney)), KGs capture intricate relationships between entities, allowing for more contextually rich predictions. These capabilities have been widely utilized in tasks such as link prediction and knowledge graph completion (KGC). Conventional methods, including TransE Bordes et al. (2013), compute embeddings for entities and relationships to enhance the comprehensiveness of KGs, supporting their use in diverse applications, such as directly information retrieval Dietz et al. (2018), or logical problems like question answering Huang et al. (2019).

Recent advances in Retrieval-Augmented Generation (RAG) have highlighted the potential of integrating external knowledge sources like KGs with large language models (LLMs) to enhance their performance in answering queries that require domain-specific information Zhang et al. (2024a); Siriwardhana et al. (2023). GraphRAG, a variant of RAG, utilizes KGs to enhance the retrieval process through relational data, thus offering more precise and contextually relevant responses from LLMs Wu et al. (2024); Matsumoto et al. (2024); Edge et al. (2024). However, despite the effectiveness of RAG methods in improving retrieval, they often fail to address the challenges posed in time-related tasks, such as planning trips or making decisions based on weather or traffic conditions. In those scenarios, LLMs must retrieve relevant information and reason, involving temporal relationships and require dynamic, context-aware reasoning that goes beyond static fact retrieval. LLMs are limited in their capacity to do such complex reasoning on huge amount of data input without substantial model fine-tuning, and existing RAG methods primarily focus on one-time knowledge retrieval rather than iterative retrieval. Figure 1 shows a scenario in which LLMs tend to generate hallucinations facing trip planning.

To address these limitations, this paper introduces a novel framework, **Knowledge Graph-Based Iterative Retrieval-Augmented Genera-**

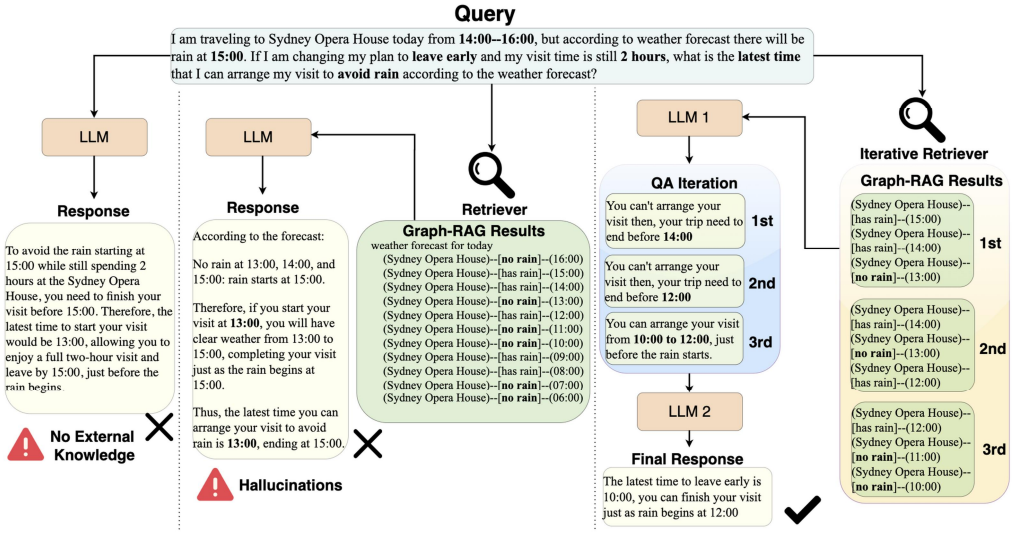

Figure 1: A comparison of LLM performance: Direct LLM output, RAG, and Iterative RAG. Without external knowledge, LLMs generate responses solely based on the provided context. However, when presented with excessive data retrieved by RAG (especially numerical information), LLMs are prone to generating hallucinated content.

tion (KG-IRAG). KG-IRAG integrates KGs with an iterative reasoning process, enabling LLMs to solve problems by incrementally retrieving relevant data through iterative queries and achieve step-by-step reasoning, allowing LLMs to address queries involving temporal dependencies without model fine-tuning. This approach is particularly suited for scenarios where reasoning must be applied alongside dynamic data retrieval. In addition to presenting the KG-IRAG framework, this paper introduces three new datasets: *weatherQA-Irish*, *weatherQA-Sydney* and *trafficQA-TFNSW*. Those datasets are designed to test LLM's ability to answer queries that require both **retrieving uncertain length of temporal information and doing mathematical reasoning**, such as determining the best time to reschedule a trip. The proposed KG-IRAG framework is validated through experiments on temporal-related question answering tasks, demonstrating its ability to significantly enhance LLM performance. By leveraging the rich relational structure of KGs and incorporating iterative reasoning, KG-IRAG offers a robust and efficient solution for addressing complex spatial temporal queries.

2 Related Work

2.1 Combination of Graphs with LLMs

Recent research has increasingly focused on combining LLMs with graphs to achieve mutual enhancement. The combination of LLMs with Graph Neural Networks (GNNs) has been shown to significantly improve the modeling capabilities of graph-structured data. LLMs have been shown to contribute to knowledge graph completion, aiding in downstream tasks such as node classification Chen et al. (2023) and link prediction Shu et al. (2024). Additionally, LLMs play a pivotal role in knowledge graph creation, transforming source texts into graphs Edge et al. (2024); Zhang et al. (2024b).

LLMs also enhance performance in Knowledge Graph-based Question Answering (KBQA) tasks, which leverage external knowledge bases to answer user queries Cui et al. (2019); Wu et al. (2019); Fu et al. (2020). KBQA has been applied across various domains, including text understanding and fact-checking Chen et al. (2019); Suresh et al. (2024). Approaches to KBQA are generally categorized into two types: Information Retrieval (IR)-based and Semantic Parsing (SP)-based. IR-based methods directly retrieve information from KG databases and use the returned knowledge to generate answers Jiang et al. (2022, 2024), whereas SP-based methods generate logical forms for queries, which are then used for knowledge retrieval Chakraborty (2024); Fang et al. (2024). Advanced techniques such as Chain of Knowledge (CoK) Li et al. (2023), G-Retriever He et al. (2024), and Chain of Explorations Sanmartin (2024) have been developed to enhance the precision and efficiency of data retrieval from KGs.

2.2 Retrieval-Augmented Generation (RAG)

RAG enhances the capabilities of LLMs by integrating external knowledge sources during the

response generation process. Unlike traditional LLMs, which rely solely on their pre-trained knowledge, RAG enables models to access real-time or domain-specific information from external databases and knowledge sources. Recent studies have explored RAG from various perspectives, including the modalities of databases Zhao et al. (2024), model architectures, training strategies Fan et al. (2024), and the diverse applications of RAG Gao et al. (2023). Effective evaluation of RAG systems requires attention to both the accuracy of knowledge retrieval and the quality of the generated responses Yu et al. (2024).

Compared to traditional RAG systems, Graph Retrieval-Augmented Generation (GraphRAG) offers a distinct advantage by retrieving knowledge from graph databases, utilizing triplets as the primary data source Peng et al. (2024). Graph-structured data capture relationships between entities and offer structural information, enabling LLMs to interpret external knowledge more effectively Hu et al. (2024); Bustamante and Takeda (2024); Sanmartin (2024).

2.3 LLMs Temporal Reasoning

Recent advancements have increasingly focused on improving the temporal reasoning capabilities of LLMs. Temporal reasoning in natural language processing (NLP) typically falls into three categories: temporal expression detection and normalization, temporal relation extraction, and event forecasting Yuan et al. (2024). The integration of temporal graphs has enabled LLMs to perform more effectively in tasks such as time comparison Xiong et al. (2024). Several temporal QA datasets have been developed to test LLMs' temporal reasoning abilities, including TEMPLAMA Dhingra et al. (2022); Tan et al. (2023), TemporalWiki Jang et al. (2022), and time-sensitive QA datasets Chen et al. (2021). By combining LLMs with temporal KGs Lee et al. (2023); Yuan et al. (2024); Xia et al. (2024), more accurate event forecasting has become possible.

While these methodologies have shown promising results, certain limitations remain: (1) few GraphRAG methods address queries highly dependent on temporal reasoning, and (2) no existing temporal QA dataset requires consecutive retrieval of uncertain amounts of data from a temporal knowledge base. The proposed KG-IRAG system aims to bridge these gaps by introducing a novel **iterative GraphRAG method** and developing datasets specifically focused on **dynamic temporal information retrieval**.

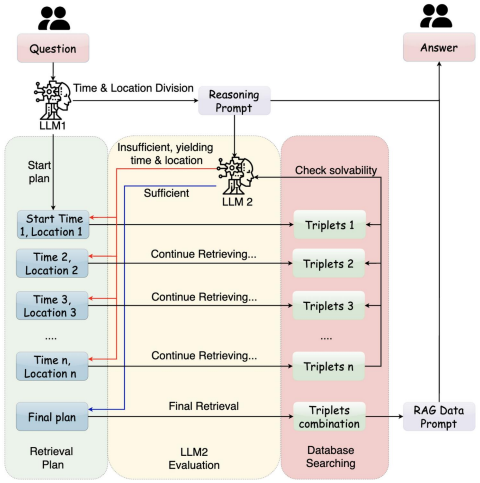

Figure 2: The KG-IRAG framework: LLM1 generates an initial retrieval plan and a reasoning prompt, guiding LLM2 through iterative retrievals or stopping to generate the final answer.

3 Methodology

This section describes the iterative process employed by the KG-IRAG framework, detailing the key components that enable the extraction of temporal and spatial data from a knowledge graph (KG) and the logical reasoning required to generate answers. The framework is designed to handle questions that require multi-step reasoning based on temporal and logical comparison.

Algorithm 1 Start process from LLM1

Require: Question, LLM1 parameter setting
Ensure: ReasoningPrompt RP, Starting time and location $time_0, location_0$

1: $RP \leftarrow$ LLM1Reasoning$(Question)$
2: Potential time and location range $\leftarrow$ LLM1
3: Starting plan $time_0, location_0$
4: Iterative Retrieval for LLM2$(time_0, location_0, RP)$

3.1 Preliminaries and method overview

Two LLMs, LLM1 and LLM2, collaborate throughout this process. LLM1 is responsible for two main tasks: (1) identifying the initial plan for KG exploration, specifically the initial start time 0 and location 0, and (2) generating a reasoning prompt that specifies the information required to answer the query and explains its relevance. The output of LLM1, as outlined in Algorithm 1, is passed to LLM2. LLM2 is responsible for evaluating

whether the retrieved data, combined with the reasoning prompt, is sufficient to resolve the query or if further retrieval steps are required.

Figure 2 illustrates the iterative RAG process. The methodology proceeds as follows: after identifying the starting time and location, KG exploration is performed iteratively to retrieve relevant triplets. Each iteration yields new data, which is evaluated by LLM2 to determine whether the problem can be solved with the current triplets and reasoning. If the answer remains unresolved, the framework adjusts the search criteria—moving to a different time or location—and continues retrieving new triplets until the answer is generated.

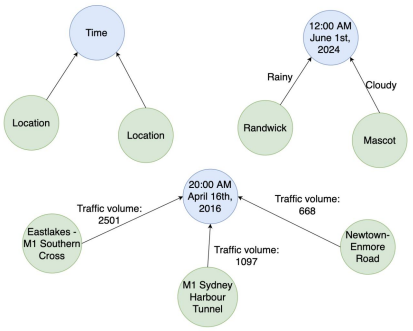

Figure 3: Knowledge graph schema for KG-IRAG

3.2 Knowledge Graph Construction

When transforming raw data into a KG, time, location, and event status (such as rainfall or traffic volume) serve as key entities. Relationships between these entities capture temporal, spatial, and event-based correlations, allowing for structured exploration during query resolution. The construction of the KG ensures that both spatial and temporal dimensions are embedded within the graph, with time being treated as an entity for easier retrieval and reasoning. Each KG provides a foundation for multi-step reasoning, as queries often span across multiple time points and locations. Specifically, locations and time points are modeled as entities, while attributes such as rain status and traffic volume are presented as relations connecting entities. An example KG schema is shown in figure 3.

3.3 Triplet Retrieval

Once the starting time (start time 0) and location (location 0) have been identified by LLM1, KG exploration begins. The retrieval process involves searching the knowledge graph for relevant triplets that match the query's initial time and location. This process is guided by LLM1's reasoning prompt, which provides a high-level explanation of the information needed to solve the query. In each iteration, the system retrieves a set of triplets (triplets1) that are relevant to the current query conditions. The retrieval is conducted through a series of KG searches that target both time and spatial relationships, ensuring that all possible relevant data is retrieved for the given query. The retrieved triplets are then passed to LLM2, which assesses whether the current data, combined with the reasoning prompt, is sufficient to answer the query. If the data is insufficient, the system moves to the next time slot or location, retrieving a new set of triplets (triplets2, triplets3, etc.) until LLM2 determines that the question can be resolved.

Algorithm 2 Iterative Retrieval for LLM2

Require: $time_i$, loc_i, ReasoningPrompt RP, LLM2 parameter setting
Ensure: Final answer after KG-IRAG process
Declare
  k = knowledge
  CombinePrompt$(\cdot)$ = CP
  JudgeSufficiency$(\cdot)$ = JS
  FindAbnormalEvent$(\cdot)$ = FAE
  CombineTimeAndLoc$(\cdot)$ = CTL
  CombinePlan$(\cdot)$ = CPPlan
  KG.retrieval$(\cdot)$ = KGret
end Declare
Algorithm Begin
$k \leftarrow$ CP$(time_i, loc_i, RP)$
$judge\_result \leftarrow$ LLM2.JS(k)
while $judge\_result$ is not sufficient **do**
  $event_{abnormal} \leftarrow$ FAE$(time_i, loc_i)$
  $time_{i'}, location_{i'}$ ▷ Generate next time period or locations based on abnormal events in current time slots or locations
  if $time_{i'}, location_{i'}$ exists **then**
    CTL$(time_i, loc_i, time_{i'}, location_{i'})$
    $k \leftarrow$ CP$(time_i, loc_i, RP)$
    $judge\_result \leftarrow$ LS(k)
  else
    return "no answer" ▷ If no more time and location can be retrieved
  end if
end while
if $judge\_result$ is sufficient **then**
  $final\_plan \leftarrow$ CPPlan$(time_i, loc_i)$
  $final\_triplets \leftarrow$ KGret$(final_plan)$
  Generate RAG_Data_Prompt
  Combine RP and RAG_Data_Prompt
  return $answer$
end if
Algorithm End

3.4 Iterative Reasoning

The iterative reasoning process is the core mechanism of KG-IRAG for handling complex temporal queries. After each retrieval of triplets, the system

evaluates whether current set of triplets, combined with the reasoning prompt generated by LLM1, is sufficient to answer the query. LLM2 is responsible for this evaluation, determining whether the current evidence from the KG provides a valid solution or if further exploration is needed. If LLM2 deems that the current triplets are insufficient to resolve the query, the system yields to a different time or location, refining the search parameters. For example, if the goal is to avoid rain during a trip, and the system detects rainfall at 9:00 AM, it may retrieve data from earlier or later time slots to identify an optimal departure time. This iterative process continues until the system finds a time and location that satisfy the query. Algorithm 2 outlines the step-by-step process of iterative reasoning for temporal queries, where LLM1 directs the search and LLM2 evaluates the retrieved data to determine retrieval should be continued.

3.5 Answer Generation

Once LLM2 confirms that a sufficient set of triplets has been retrieved to resolve the query, the final step is to generate the answer. The triplets from the various iterations of the KG search are combined into a coherent RAG Data Prompt, and the sufficient triplet evidences are then processed together with the former reasoning prompt, together to get the answer generated. If data is insufficient to yield the answer or is out of boundary during KG lookup, 'no answer' is returned as the result.

3.6 Case Study

To further explain the details of 'iteration' in KG-IRAG, a case study was conducted. Presented in section A, it simulate an initial travel plan to Sydney Opera House, including defining whether the initial plan is valid, as well as how to define an optimal time to adjust the trip.

4 Dataset properties

4.1 Data properties

To further test a RAG model's ability to explore and do logical reasoning on temporal data, 3 datasets, weatherQA-Irish, weatherQA-Sydney and trafficQA-TFNSW are prepared for experiments. From the perspective of RAG, the construction of these datasets is tailored to allow the model to perform both entity-based retrieval and time-dependent reasoning. beginning with straightforward entity identification and advancing to dynamic temporal reasoning over multiple steps. Sydney weather data is comprised of half-hourly records, while Irish weather and TFNSW data are collected on an hourly basis. The high-resolution temporal granularity of these datasets allows for the construction of knowledge graphs that not only record spatial relationships but also capture detailed temporal data. What's more, in the TFNSW dataset, there are also traffic directions, offering more details of spatial information. In each dataset, attributes such as date, location, and event status (e.g., rainfall or traffic volume) are structured as KG entities and relations.

4.2 Question Designs

The design of the questions in the QA datasets is structured to progressively increase the complexity of the retrieval tasks. The first question (Q1) is a fundamental entity recognition and retrieval task, while the second (Q2) and third (Q3) questions introduce logical reasoning by incorporating time-dependent queries. The goal is to examine the system's ability to not only retrieve relevant data but also engage in iterative reasoning over time.

4.2.1 Question 1: Abnormal Event Detection

Q1 is designed as a straightforward retrieval task in which the model identifies whether an "abnormal" event, such as rainfall or traffic congestion, occurred during a specific time slot. This task relies primarily on entity recognition and retrieval of static information from the knowledge graph. As a result, it reflects a typical RAG problem, where the system extracts relevant data based on predefined entities and times.

4.2.2 Questions 2 and 3: Leave Early or Late

In contrast to Q1, the design of Q2 and Q3 introduces temporal reasoning components that significantly increases the complexity of the retrieval process. Q2 asks the model to calculate the latest time at which one can leave early to avoid an abnormal event, while Q3 focuses on determining the earliest time to leave late. Both questions require the model to infer optimal departure times by reasoning over a temporal range, making use of the temporal relationships embedded in KGs.

From the RAG perspective, Q2 and Q3 can be seen as decompositions of multiple Q1-type subquestions. Instead of a single, static retrieval, the model must query the knowledge graph iteratively, retrieving data from different time slots and inte-

grating that information to form a cohesive answer. This process illustrates the key idea of KG-IRAG: transforming a larger temporal question into multiple fixed-time subproblems, each solvable via entity and temporal information extraction.

4.2.3 Dynamic Problem Decomposition

The design of Q2 and Q3 exemplifies the concept of dynamic problem decomposition in retrieval-augmented generation. By requiring the model to engage in time-based reasoning, these questions push beyond standard entity recognition and challenge the system to handle a broader scope of temporal logic. LLMs need to solve the query through a combination of logical steps.

5 Experiments

5.1 Datasets

Three datasets are used to evaluate the performance of KG-IRAG: Irish weather data, Sydney weather data, and Traffic Volume of Transport for New South Wales (TFNSW) data. While the two weather datasets are used to identifying abnormal events like rainfall, the TFNSW dataset includes extensive numerical data on traffic volume, adding complexity to the questions by requiring **numerical comparisons**. All three datasets are transferred into KGs that capture location relationships and temporal records. While the details od datasets are presented in B,Table 1 presents attributes used to form the KGs, with time treated as an entity to enhance retrieval capabilities. Table 2 represents the size of each dataset and the corresponding KGs generated.

To form the QA datasets, for each year, 200 time slots were chosen randomly, and the questions are generated based on the following criteria: Question 1 asks about in the chosen time slot, whether there is certain 'abnormal' activity(i.e. rainfall, traffic jam), answer with 'True' or 'False'. If the answer is yes in question 1, question 2 and 3 care about what is the latest time to head off early and the earliest time to head off late under the same length of the trip. (the maximum early or late time is set to be 12 hours for weather datasets and 9 hours for the TFNSW dataset). Table 3 shows the details of question proportion in the three datasets.

5.2 Evaluation

Experiments are conducted to evaluate the datasets. For each question, a minimal subset of data, referred to as "standard data," is selected according to the criteria outlined in equation (1) and (2).

The standard data represents the minimal amount of information necessary to answer the query, ensuring that only the relevant data is used (e.g., if data from 7:00 to 12:00 is required to answer a question related to the status of 7:00 to 10:00, then no additional data outside 7 to 12 is included). Comparisons are made by feeding the standard data into LLMs in raw data format (data frame), context-enhanced data, and KG triplet representations.

$$SD = \arg \min_{D \subseteq D_{\text{all}}} \text{ solves the query} \quad (1)$$

$$\forall d \in SD, (SD \setminus d) \text{ does not solve the query} \quad (2)$$

Standard evaluation metrics are employed to assess QA systems: Exact Match (EM), F1 Score, and Hit Rate (HR). In addition, hallucinations are considered to evaluate the accuracy of KG-IRAG compared with other methods. These metrics collectively measure the model's capability to provide accurate answers while avoiding the retrieval of both excessive and insufficient knowledge. **Exact Match (EM)** calculates the percentage of results that exactly match the ground truth answer, defined as $EM(\%) = \frac{\Sigma_{i=0}^{N} answer_i = truth_i}{N}$. **F1 Score** is calculated based on precision and recall. Precision measures the correctness of the retrieved information, while recall assesses how much of the target data has been retrieved, shown as $Precision = \frac{I_{retrieved} \cap SD}{I_{retrieved}}$, $Recall = \frac{I_{retrieved} \cap SD}{SD}$ and $F1 = \frac{2 * Precision * Recall}{Precision + Recall}$.

Hit Rate (HR) evaluates the portion of information retrieved by the RAG system that is useful for generating the correct answer. The modified Hit Rate ($HR = \frac{I_{retrieved} \cap SD}{I_{retrieved} \cup SD}$) reflects situations where the RAG system retrieves either too much or too little of the necessary information.

Hallucination refers to content generated by LLMs that is not present in the retrieved ground truth Ji et al. (2023); Li et al. (2024); Perković et al. (2024). It is a crucial metric for assessing the accuracy and reliability of QA systems, calculated as shown in (4). Specifically, hallucination is defined as follows during the evaluation: For Question 1, hallucination occurs if the answer is incorrect, as this indicates that the LLM failed to correctly identify the abnormal event despite the relevant data being provided. For Questions 2 and 3, hallucination is detected if the LLM: (a) generates an answer indicating an abnormal event at an

incorrect time; (b) produces an answer that does not appear in the provided data; or (c) in the case of the KG-IRAG system, LLM2 fails to decide when to stop the exploration process.

$$\text{hal}(answer_i, truth_i) = \begin{cases} 1 & \text{hallucination detected} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$Hallucination(\%) = \frac{\Sigma_{i=0}^{N} hal(answer_i, truth_i) = 1}{N} \quad (4)$$

While hallucination can be directly calculated for Question 1, whereas for Questions 2 and 3, **50 questions** are randomly selected from each dataset and **manual reviews** are conducted to assess the hallucinations in the answers from LLMs. To benchmark the performance of the proposed model (KG-IRAG), several LLMs are utilized Touvron et al. (2023); Achiam et al. (2023): 1) Llama-3-8B-Instruct, 2)GPT-3.5-turbo-0125 3) GPT-4o-mini-2024-07-18, 4)GPT-4o-2024-08-06. KG-IRAG is compared against two RAG methods: The first one is a standard one without exploration, by asking LLMs to decide the data needed to solve the problem, retrieving the data then feed into LLMs. The second one is KG-RAG, which contains Chain of Exploration for KG Retrieval Sanmartin (2024).

6 Result

6.1 Experiment Results

Two stages of experiments are set to evaluate 1)LLMs' ability to solve temporal reasoning problems with correct input and 2)How different RAG systems improve LLMs' QA accuracy.

6.1.1 Testing LLMs using Facts

The first set of experiments focuses on testing LLMs directly on the datasets. Standard data are calculated and collected according to equation (1) and (2). This data is converted into three formats: raw data (in table format), text data (by converting the data into text descriptions in various forms), and triplet data (extracted from knowledge graphs). The experiments are designed to assess whether LLMs can effectively use these different data formats, and which format is best recognized by LLMs. For each question, the different formats of the standard data are combined with the query into a single prompt. The goal is to determine whether LLMs can solve problems involving temporal and logical reasoning **without iterative reasoning but with the correct data**. The results are shown in table 4.

6.1.2 Testing different RAG systems

The second stage compares the performance of three different RAG systems: 1)Standard Graph-RAG system, where LLMs are prompted to determine "what data is needed to solve the problem," retrieve the necessary data based on responses and then provide the final answer; 2)KG-RAG Sanmartin (2024), which uses a Chain of Exploration to retrieve data step-by-step, with the exploration plan set to **three steps**, considering the temporal span in the questions; 3)KG-IRAG: our proposed system. For testing KG-IRAG, the same LLM1 and LLM2 models (as illustrated in Figure 2) are used for both data retrieval and iterative reasoning. Detail results of exact match, F1 score, hit rate and hallucination for different models applying RAG systems on the 3 datasets are shown in table 5, 6, and 7, while the plot of exact match is in figure 4.

6.2 Result analysis

6.2.1 Different Format of Data Input

When comparing the use of text data, raw table data, and triplet data, LLMs consistently perform better with **triplet data**. For Question 1, LLMs effectively determine whether there is rain in the provided data and the specified time range, leading to very prominent results for Q1 in the two weather datasets. However, when the numerical comparison is required, as seen in the TrafficQA dataset, only GPT-4o generates satisfied results. For Questions 2 and 3, when all data is directly fed into LLMs within a single prompt, the accuracy is not as promising. Even with background knowledge such as "there is rain/traffic congestion in the original plan", LLMs struggle to determine the correct answers when asked to find the earliest time to delay or the latest time to advance the plan. This highlights the necessity of guiding LLMs through step-by-step reasoning and data exploration.

6.2.2 RAG Comparation: Question 1

For Q1, all three systems (Standard Graph-RAG, KG-RAG, KG-IRAG) perform similarly. This is because LLMs do not require step-by-step reasoning to determine which data to retrieve from the database—they can reliably identify entity names and times in the queries and retrieve correct data through one QA. Consequently, the results for Question 1 are comparable to simply feeding correct data to the LLMs and asking them to detect abnormal activities. Therefore, no significant differences were observed between the three systems

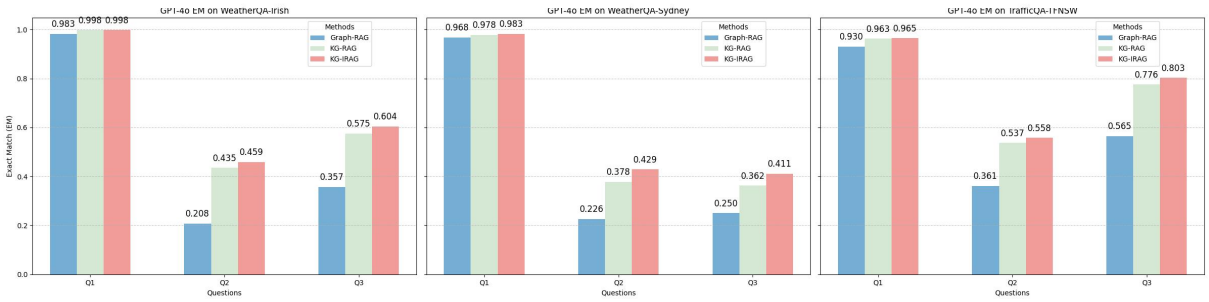

Figure 4: Experiment results(EM) on three datasets using different RAG methods.

in this task.

6.2.3 RAG Comparation: Question 2 & 3

For Q2 and Q3, the standard Graph-RAG system shows notable limitations. When prompted with "what data is needed to solve the problem," LLMs tend to request the maximum range of data due to the uncertainty of the time period. This results in excessive data retrieval, reaching the maximum early or late time (12 hours for weather data and 9 hours for the TFNSW data). The large volume of unnecessary data increases the likelihood of hallucination, resulting in lower accuracy. KG-RAG system uses a limited length of the exploration plan, and the Chain-of-Exploration (CoE) for step-by-step KG retrieval is primarily effective for exploring multi-hop relations in the knowledge graph. However, it struggles with temporal data. In some cases, KG-RAG prematurely stops exploration, resulting in insufficient data to solve the problem and generating incorrect answers.

The **KG-IRAG** system is more flexible and outperforms above two methods in most cases. With the aid of an additional reasoning prompt, LLMs guide data retrieval based on two criteria: (1) whether the current data is sufficient to generate an answer, and (2) which time and entity to explore next, not based on positional relations in the KG, but on the occurrence of abnormal events in the current plan—an aspect LLMs can easily detect.

6.2.4 Hallucinations

It is important to note that when the data contains more numerical values, the KG-IRAG system tends to **generate higher levels of hallucination**, shown the most in TrafficQA-TFNSW dataset. Some observations and conclusions are drawn:

1. **'Late Stop' Phenomenon**: During evaluation, KG-IRAG occasionally tends to stop late and continues retrieving data unnecessarily. Specifically, in manny cases LLM2 in KG-IRAG decided to retrieve one or two more round of data, leading to an overload of information. The 'hallucination' occured more easily in questions which contain too many numbers, like the TrafficQA-TFNSW.

2. In other baselines, hallucination often lead to 'LLMs generate wrong answer using insufficient data', which is considered more harmful since it directly leads to wrong conclusion.

3. Although KG-IRAG indeed tend to retrieve some extra rounds of data, that hallucination won't normally lead to a wrong answer, since sufficient data is already in the retrieval plan.

7 Conclusions

In this paper, a new RAG framework is proposed, i.e., **Knowledge Graph-Based Iterative Retrieval-Augmented Generation(KG-IRAG)**, which enhances the integration of KG with LLMs through iterative reasoning and retrieval. KG-IRAG allows LLMs to make more informed decisions when answering complex, temporally dependent queries by progressively retrieving relevant data from KGs. Unlike traditional RAG methods, KG-IRAG flexibly and effectively employs a step-by-step retrieval mechanism that guides LLMs in determining when to stop exploration, significantly improving response accuracy.

To evaluate the framework's effectiveness, three new datasets, **weatherQA-Irish**, **weatherQA-Sydney**, and **trafficQA-TFNSW**, have been introduced. These datasets, which involve real-world scenarios such as weather and traffic conditions, are designed to test the ability of LLMs to handle time-sensitive and event-based queries requiring both temporal reasoning and logical inference. Experimental results demonstrate that KG-IRAG excels in complex reasoning tasks by generating more accurate and effecient responses.

8 Limitations

Although KG-IRAG presents notable improvements in LLMs' ability to tackle knowledge-intensive tasks, there are still several limitations, including:

(1) There is a need for further enhancement in the reasoning mechanism to ensure more accurate and efficient data management.

(2) The influence of imperfect retrieval(retrieve extra data) in some cases remains a challenge, and future improvements should focus on mitigating its impact on performance.

(3) There would be potential need for balancing the use of internal and external knowledge, such as Wang et al. (2024)

9 Acknowledgements

This research is partially supported by the Technology Innovation Institute, Abu Dhabi, UAE. Additionally, computations were performed using the Wolfpack computational cluster, supported by the School of Computer Science and Engineering at UNSW Sydney.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Diego Bustamante and Hideaki Takeda. 2024. Sparql generation with entity pre-trained gpt for kg question answering. *arXiv preprint arXiv:2402.00969*.

Abir Chakraborty. 2024. Multi-hop question answering over knowledge graphs using large language models. *arXiv preprint arXiv:2404.19234*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2023. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668*.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2019. Kbqa: learning question answering over qa corpora and knowledge bases. *arXiv preprint arXiv:1903.02419*.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Laura Dietz, Alexander Kotov, and Edgar Meij. 2018. Utilizing knowledge graphs for text-centric information retrieval. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1387–1390.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Haishuo Fang, Xiaodan Zhu, and Iryna Gurevych. 2024. Dara: Decomposition-alignment-reasoning autonomous language agent for question answering over knowledge graphs. *arXiv preprint arXiv:2406.07080*.

Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. 2020. A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

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

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. Grag: Graph retrieval-augmented generation. *arXiv preprint arXiv:2405.16506*.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 105–113.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. *arXiv preprint arXiv:2204.14211*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. 2024. Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. *arXiv preprint arXiv:2402.11163*.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. *arXiv preprint arXiv:2212.00959*.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal knowledge graph forecasting without knowledge using in-context learning. *arXiv preprint arXiv:2305.10613*.

Jiarui Li, Ye Yuan, and Zehua Zhang. 2024. Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv preprint arXiv:2403.10446*.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2023. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. *arXiv preprint arXiv:2305.13269*.

Nicholas Matsumoto, Jay Moran, Hyunjun Choi, Miguel E Hernandez, Mythreye Venkatesan, Paul Wang, and Jason H Moore. 2024. Kragen: a knowledge graph-enhanced rag framework for biomedical problem solving using large language models. *Bioinformatics*, 40(6).

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Gabrijela Perković, Antun Drobnjak, and Ivica Botički. 2024. Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pages 2084–2088. IEEE.

Diego Sanmartin. 2024. Kg-rag: Bridging the gap between knowledge and creativity. *arXiv preprint arXiv:2405.12035*.

Dong Shu, Tianle Chen, Mingyu Jin, Yiting Zhang, Mengnan Du, and Yongfeng Zhang. 2024. Knowledge graph large language model (kg-llm) for link prediction. *arXiv preprint arXiv:2403.07311*.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Suryavardan Suresh, Anku Rani, Parth Patwa, Aishwarya Reganti, Vinija Jain, Aman Chadha, Amitava Das, Amit Sheth, and Asif Ekbal. 2024. Overview of factify5wqa: Fact verification through 5w question-answering. *arXiv preprint arXiv:2410.04236*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. *arXiv preprint arXiv:2306.08952*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö Arık. 2024. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. *arXiv preprint arXiv:2410.07176*.

Junde Wu, Jiayuan Zhu, and Yunli Qi. 2024. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187*.

Peiyun Wu, Xiaowang Zhang, and Zhiyong Feng. 2019. A survey of question answering over knowledge base. In *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding: 4th China Conference, CCKS 2019, Hangzhou, China, August 24–27, 2019, Revised Selected Papers 4*, pages 86–97. Springer.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiaoyu Zhang. 2024. Enhancing temporal knowledge graph forecasting with large language models via chain-of-history reasoning. *arXiv preprint arXiv:2402.14382*.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *arXiv preprint arXiv:2401.06853*.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 1963–1974.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024a. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Yuzhe Zhang, Yipeng Zhang, Yidong Gan, Lina Yao, and Chen Wang. 2024b. Causal graph discovery with retrieval-augmented generation based large language models. *arXiv preprint arXiv:2402.15301*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

A Case study

To further explain the details of ‘iteration‘ in KG-IRAG, a case study was conducted. Considering the following Q1, Q2 and related data:

Q1: *‘I plan to visit the Sydney Opera House from 3:00 to 5:00 on December 5th. Considering the weather, can I avoid rain during the trip? ‘*

Q2: *‘If I can’t avoid the rain, what is the earliest time I can postpone my trip to avoid rain? ‘*

Sydney Opera House’s weather: 3:00 cloudy; **3:30 rain**; 4:00 cloudy; 4:30 cloudy; 5:00 cloudy; **5:30 rain**; **6:00 rain**; 6:30 cloudy; 7:00 cloudy; 7:30 cloudy; 8:00 cloudy; 8:30 cloudy

Considering the predefined step are set to be **1-hour increments**. For Q1, the initial plan to retrieve the data is: (Sydney Opera House, December 5th, **3:00 to 4:00**). Based on the retrieved data, LLM2 judges there is rain(‘abnormal‘ event) on 3:30, thus the answer is false for Q1. For Q2, the reasoning logic is to ‘find a earliest 2-hour time slot after 3:30 that has no rain‘, after the initial plan, the next plans are: (Sydney Opera House, December 5th, **4:00 to 5:00**); (Sydney Opera House, December 5th, **5:00 to 6:00**). After LLM2 judging there is no rain from 4-5 but there is from 5-6, the next time plan ‘jumps‘ to be started from 6:30, which is: (Sydney Opera House, December 5th, **6:30 to 7:30**), following by (Sydney Opera House, December 5th, **7:30 to 8:30**)

After multiple rounds of iterations, LLMs generate the conclusion: **The earliest time that I can postpone the trip is to 6:30**.

B Dataset Properties

Three datasets are used to evaluate the performance of KG-IRAG. The first one is Irish weather data from January 2017 to December 2019, obtained from the Met Eireann website, including 25 stations across 15 counties in Ireland on an **hourly basis**. The second one is Sydney weather data from January 2022 to mid-August 2024, also **collected every 30 minutes** from the National Oceanic and Atmospheric Administration, offering a **higher data frequency** than the Irish dataset. The last one is Traffic Volume of Transport for New South Wales (TFNSW) data, which comprises traffic volume data from permanent traffic counters and classifiers in Sydney, recorded **hourly** from 2015-2016.

Table 1: Attributes for constructing KGs and QAs in datasets

Irish Weather Dataset	
attribute	detail explanation
date	date of record
county	location of county
station ID	station ID
station	station name
rain	hourly rainfall volume
Sydney Weather Dataset	
attribute	detail explanation
date	date of record
station ID	station ID
station	station name
rain	rainfall volume counted every 30 minutes
TFNSW Dataset	
attribute	detail explanation
date	date of record
direction	cardinal direction of the traffic (northbound, southbound)
volume	hourly traffic volume counter

Table 2 presents the quantities of data used in three different datasets: the Irish Weather Dataset, the Sydney Weather Dataset, and the TFNSW Dataset. For each dataset, the table includes the period of data collection, the number of entities, the number of relationships, and the total number of records.

The ’record set’ refers to the number of records in the **raw data**. The raw data includes information that may not be directly relevant or useful for the

Table 2: Quantities of data used in datasets

Irish Weather Dataset			
data	entities	relations	records
2017-2019	227760	876000	219000
Sydney Weather Dataset			
data	entities	relations	records
2022-2024	332433	559673	279837
TFNSW Dataset			
data	entities	relations	records
2015-2016	132042	683002	683002

Table 3: Properties of Questions in the QA datasets

Question 1: 'abnormal' event detection			
Dataset	numbers	'True' Ans	'False' Ans
weatherQA-Irish	600	393	207
weatherQA-Sydney	600	211	389
trafficQA-TFNSW	400	253	147
Question 2&3: leave early or late			
data	entities	has ans	no ans
weatherQA-Irish	207	204	3
weatherQA-Sydney	389	339	50
trafficQA-TFNSW	147	135	12

QA dataset. For example, in the Irish and Sydney weather datasets, the raw data contains additional attributes such as humidity, sunlight, and even duplicate entries. These elements are excluded during preprocessing, resulting in a smaller number of records compared to the number of relations. In contrast, the TFNSW data is more structured, containing only traffic volume data, all of which is fully utilized in our QA dataset.

C Experiment on QA datasets-Direct Data Input

In the first round of experiments, four LLMs are used to test how LLMs can directly answer the constructed three QA datasets (*weatherQA-Irish*, *weatherQA-Sydney*, and *trafficQA-TFNSW*), with some settings:

1) To ensure LLMs are not interrupted by useless information, input prompts only contain questions as well as **least needed data**.

2) To test the effeciency of data input, three formats of data are used: raw data (table) format, text data (by transferring data into text description), and triplet format (KG structure). Tests using threee formats of data are done separately.

3) The final answer are compared directly with correct answers, the exact match(EM) value is shown in table 4.

D Experiment on QA datasets-Different RAG Methods

In second stage of experiments, KG-IRAG is compared with Graph-RAG and KG-RAG Sanmartin (2024). None data are provided in the beginning, data retrieval plan are generated based on different framework. The exact match, like last stage of experiment, is the result of comparison between generated final answer and true answer. F1 Score and Hit Rate focus more on the whether the retrieval is both enough and non-excessive. Hallucinations are judged based on answers generated by LLMs under different framework. For Q1, F1 Score and Hit Rate is not considered, since it contain less temporal reasoning compared with Q2 and Q3. The results of WeatherQA-Irish are shown in table 5, while WeatherQA-Sydney results are shown in table 6, and those of TrafficQA-TFNSW are in table 7.

Table 4: Comparison of Direct Data Inputs on Three Datasets Based on Exactly Match(EM)

Data		Llama-3-8b			GPT-3.5-turbo			GPT-4o-mini			GPT-4o		
		Raw	Text	Triplet	Raw	Text	Triplet	Raw	Text	Triplet	Raw	Text	Triplet
weatherQA-Irish	Q1	0.905	0.9016	**0.9316**	0.9483	0.9417	**0.9783**	0.9817	0.9833	**0.9933**	0.9950	0.9967	**1**
	Q2	0.1159	0.1159	**0.1256**	**0.14**	0.0966	0.1111	0.1932	0.1498	**0.2415**	0.3865	0.3314	**0.396**
	Q3	0.3233	**0.343**	0.3382	0.2946	0.2995	**0.314**	0.4203	0.4348	**0.4492**	0.4831	0.5072	**0.5169**
weatherQA-Sydney	Q1	**0.8867**	0.855	0.8683	0.9117	0.7083	**0.9383**	**1**	0.9667	0.9717	0.9867	0.9667	**0.99**
	Q2	**0.1542**	0.1285	0.1491	0.1259	0.162	**0.1722**	0.2314	0.2596	**0.3033**	0.2545	0.2879	**0.347**
	Q3	0.1722	0.1517	**0.1877**	0.1671	0.2108	0.2005	0.2416	**0.2699**	**0.2699**	0.2956	**0.3316**	0.3265
trafficQA-TFNSW	Q1	0.52	0.515	**0.5325**	**0.4125**	0.395	0.41	0.5975	0.6475	**0.73**	0.95	0.965	**0.9725**
	Q2	0.1633	**0.1837**	0.1701	0.1293	0.1293	**0.1565**	0.2721	0.2857	**0.3061**	0.4354	0.381	**0.4762**
	Q3	**0.2381**	0.1973	0.2313	**0.2109**	0.2041	**0.2109**	0.4014	0.381	**0.4285**	0.6871	0.66	**0.7279**

Table 5: Comparison of Different RAG Methods on WeatherQA-Irish Dataset

WeatherQA-Irish											
Model	Data Type	Question 1		Question 2				Question 3			
		EM	Hall.	EM	F1 Score	HR	Hall.	EM	F1 Score	HR	Hall.
Llama-3-8b	Graph-RAG	0.916	0.086	0.063	0.673	0.509	0.5	0.15	0.684	0.517	0.48
	KG-RAG	0.892	0.108	0.217	0.815	0.694	0.34	0.386	0.809	0.686	**0.22**
	KG-IRAG	**0.927**	**0.081**	**0.242**	**0.821**	**0.702**	**0.32**	**0.415**	**0.831**	**0.715**	**0.22**
GPT-3.5-turbo	Graph-RAG	**0.958**	0.042	0.087	0.673	0.509	0.48	0.14	0.684	0.517	0.42
	KG-RAG	0.945	0.055	0.208	0.807	0.683	**0.32**	0.362	0.803	0.677	**0.18**
	KG-IRAG	**0.958**	**0.036**	**0.222**	**0.817**	**0.697**	**0.32**	**0.386**	**0.824**	**0.703**	0.2
GPT-4o-mini	Graph-RAG	0.97	0.03	0.145	0.673	0.509	0.32	0.251	0.684	0.517	0.28
	KG-RAG	0.975	**0.025**	0.304	0.832	0.718	0.24	0.546	0.813	0.695	0.18
	KG-IRAG	**0.983**	0.028	**0.323**	**0.84**	**0.721**	**0.22**	**0.585**	**0.859**	**0.754**	**0.16**
GPT-4o	Graph-RAG	0.983	0.017	0.208	0.673	0.509	0.24	0.357	0.684	0.517	0.22
	KG-RAG	**0.998**	**0.002**	0.435	0.857	0.751	**0.14**	0.575	0.839	0.728	0.16
	KG-IRAG	**0.998**	**0.002**	**0.459**	**0.875**	**0.783**	0.18	**0.604**	**0.88**	**0.789**	**0.14**

Table 6: Comparison of Different RAG methods on WeatherQA-Sydney Dataset

WeatherQA-Sydney											
Model	Data Type	Question 1		Question 2				Question 3			
		EM	Hall.	EM	F1 Score	HR	Hall.	EM	F1 Score	HR	Hall.
Llama-3-8b	Graph-RAG	0.877	0.123	0.141	0.514	0.349	0.44	0.167	0.503	0.326	0.42
	KG-RAG	0.908	0.092	0.183	0.762	0.63	0.26	0.226	0.741	0.606	**0.24**
	KG-IRAG	**0.916**	**0.084**	**0.192**	**0.841**	**0.721**	**0.24**	**0.239**	**0.833**	**0.711**	0.26
GPT-3.5-turbo	Graph-RAG	0.892	0.108	0.139	0.514	0.349	0.46	0.157	0.503	0.326	0.42
	KG-RAG	0.933	0.067	0.205	0.751	0.617	**0.24**	0.231	0.738	0.602	**0.22**
	KG-IRAG	**0.938**	**0.062**	**0.246**	**0.837**	**0.716**	0.22	**0.257**	**0.844**	**0.725**	**0.22**
GPT-4o-mini	Graph-RAG	0.947	0.053	0.195	0.514	0.349	0.3	0.188	0.503	0.326	0.36
	KG-RAG	**0.973**	0.027	0.29	0.775	0.644	0.18	0.337	0.78	0.651	**0.2**
	KG-IRAG	0.967	**0.033**	**0.352**	**0.862**	**0.761**	**0.16**	**0.362**	**0.856**	**0.757**	**0.2**
GPT-4o	Graph-RAG	0.968	0.032	0.226	0.514	0.349	0.26	0.25	0.503	0.326	0.22
	KG-RAG	0.978	0.022	0.378	0.785	0.657	0.16	0.362	0.793	0.663	0.14
	KG-IRAG	**0.983**	**0.017**	**0.429**	**0.852**	**0.75**	**0.12**	**0.411**	**0.869**	**0.769**	**0.12**

Table 7: Comparison of Different RAG methods on TrafficQA-TFNSW dataset

TrafficQA-TFNSW											
Model	Data Type	Question 1		Question 2				Question 3			
		EM	Hall.	EM	F1 Score	HR	Hall.	EM	F1 Score	HR	Hall.
Llama-3-8b	Graph-RAG	0.465	0.535	0.095	0.694	0.537	0.48	0.122	0.709	0.548	0.4
	KG-RAG	0.503	0.497	0.218	0.843	0.725	**0.2**	0.306	0.835	0.716	**0.22**
	KG-IRAG	**0.52**	**0.48**	**0.229**	**0.863**	**0.759**	0.21	**0.333**	**0.851**	**0.745**	0.26
GPT-3.5-turbo	Graph-RAG	0.418	0.582	0.075	0.694	0.537	0.5	0.088	0.709	0.548	0.44
	KG-RAG	0.43	0.57	0.224	0.841	0.722	**0.22**	0.252	0.833	0.712	**0.22**
	KG-IRAG	**0.46**	**0.54**	**0.239**	**0.858**	**0.761**	**0.22**	**0.279**	**0.868**	**0.766**	0.26
GPT-4o-mini	Graph-RAG	0.635	0.365	0.238	0.694	0.537	0.36	0.32	0.709	0.548	0.32
	KG-RAG	0.722	0.263	0.374	0.852	0.736	**0.14**	0.503	0.857	0.743	**0.16**
	KG-IRAG	**0.748**	**0.251**	**0.381**	**0.882**	**0.785**	0.22	**0.51**	**0.894**	**0.813**	0.18
GPT-4o	Graph-RAG	0.93	0.07	0.361	0.694	0.537	0.25	0.565	0.709	0.548	0.26
	KG-RAG	0.963	0.047	0.537	0.878	0.783	**0.14**	0.776	0.869	0.771	**0.12**
	KG-IRAG	**0.965**	**0.035**	**0.558**	**0.886**	**0.802**	0.2	**0.803**	**0.914**	**0.843**	0.16