

A White-box/Black-box Dual-modal Structured Query Method for Flood Disaster Spatiotemporal Knowledge Graph Question Answering

Hao Li

College of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China

Email: qingyuepei@gmail.com

How to cite this paper: Li, H. (2026). A White-box/Black-box Dual-modal Structured Query Method for Flood Disaster Spatiotemporal Knowledge Graph Question Answering. *Journal of Computer Science and Frontier Technologies*, 3(2), 14-29. ISSN Print: 3104-4204; ISSN Online: 3104-4212.

<https://doi.org/10.63313/JCSFT.9067>

Published: 2026-05-06

Copyright © 2026 by author(s) and Erytis Publishing Limited.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Abstract

Flood disaster emergency question answering (QA) requires both structured fact retrieval and complex compositional reasoning. Existing large language model (LLM)-based knowledge graph QA methods still struggle to balance query flexibility and execution reliability. To address this contradiction, this paper proposes a white-box/black-box dual-modal structured query method for flood disaster spatiotemporal knowledge graph QA. The method takes the ReAct reasoning mechanism as its scheduling core, integrating LLM-generated dynamic Cypher queries (white-box path) and predefined scripted queries (black-box path) into a unified reasoning chain. For standardized, high-frequency fact queries, the system preferentially employs black-box scripts; for tasks involving causal chain tracing, trend statistics, and integrated decision-making, the appropriate query or mixed retrieval path is invoked based on question characteristics. A query routing strategy is designed, and a data flow/control flow separation mechanism is introduced to manage large-scale query results through a "summary-file bypass" approach, alleviating context bloat during multi-round reasoning. Based on a flood disaster spatiotemporal knowledge graph and emergency response plan corpus, 30 test questions are constructed around three complex task types—causal analysis, trend assessment, and integrated decision-making—with current status query cases used to validate the standard script path. Experimental results show that the dual-modal method achieves a 100% query success rate, 98.3% answer accuracy, and an average reasoning round count of 1.47, outperforming both the pure white-box and pure black-box methods overall.

Keywords

Flood Disaster; Spatiotemporal Knowledge Graph; Large Language Model; React; Cypher; Structured Query

1. Introduction

In flood disaster emergency decision-making, time constraints are typically severe.

Decision-makers must rapidly grasp current disaster conditions, causal evolution, temporal trends, and supporting information from emergency response plans simultaneously. Unlike general-purpose QA, these questions involve attribute extraction, temporal constraint filtering, causal chain tracing, and regulatory document matching at the same time, imposing higher demands on knowledge organization and query mechanisms. Most existing emergency information systems focus primarily on data aggregation and static display, with insufficient capability for structured knowledge retrieval and reasoning over complex questions, making it difficult to support a higher level of intelligent emergency QA.

The combination of knowledge graphs and large language models (LLMs) provides a new technical pathway for emergency QA.[1], [2] Retrieval-augmented generation (RAG) can leverage external knowledge to mitigate factual hallucinations in LLMs[3], while structured query mechanisms based on knowledge graphs facilitate precise fact retrieval and relational reasoning[4]. However, in vertical-domain spatiotemporal knowledge graph scenarios, these methods face two key contradictions: white-box queries that rely on LLMs to dynamically generate Cypher statements offer strong adaptability but suffer from reduced execution reliability under complex graph structures, strict temporal constraints, and multi-condition combinations[5]; black-box query paths based on predefined scripts are more controllable but constrained by template coverage and ill-suited for open-ended questions. Striking a balance between dynamic query capability and execution reliability remains a key challenge in flood disaster emergency QA.

Beyond the query mechanism itself, complex emergency QA typically cannot be resolved in a single retrieval step. The system must iterate through "understanding questions—invoking tools—observing results—adjusting strategies." If all raw results returned by graph queries are injected directly into the LLM context, this not only rapidly exhausts the context window but may also cause the model to become trapped in local details, affecting subsequent reasoning. Therefore, the way retrieval results are organized within the reasoning chain also affects overall QA performance.

To address these issues, this paper proposes a white-box/black-box dual-modal structured query method for flood disaster spatiotemporal knowledge graph QA. The method takes the ReAct reasoning mechanism as its scheduling core and constructs a dual-modal query framework in which dynamic Cypher generation and scripted queries operate collaboratively[6]. For standardized, high-frequency fact questions, black-box scripts are prioritized; for questions involving multi-condition combinations, causal chain analysis, and trend statistics, white-box dynamic query generation is employed. A collaborative routing strategy for different query types is further designed, and a data flow/control flow separation mechanism is introduced to alleviate context bloat from large result sets.

The main contributions of this paper are as follows:

(1) A white-box/black-box dual-modal structured query framework for flood disaster emergency QA is proposed, which coordinates dynamic Cypher generation and predefined script queries within a unified ReAct reasoning chain.

(2) A query routing strategy targeting different task types—including current status queries, causal analysis, trend assessment, and integrated decision-making—is designed to improve overall query adaptability across multi-type task scenarios.

(3) A data flow/control flow separation result management mechanism is introduced, combining summarized returns with file bypass storage to reduce context burden during multi-round reasoning.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 presents the dual-modal structured query method; Section 4 describes experimental design and results; Section 5 concludes the paper and discusses future work.

2. Related Work

2.1. Disaster Knowledge Graphs and Emergency QA

Research on disaster knowledge graphs primarily centers on knowledge objects such as disaster entities, disaster attributes, emergency resources, and response measures, organizing disaster domain information in a structured manner to support monitoring analysis, risk assessment, and decision-making assistance. Existing studies have constructed knowledge graphs for disaster processes and emergency resource management in scenarios including floods, earthquakes, and geological disasters, demonstrating the effectiveness of knowledge graphs in integrating heterogeneous disaster data and semantic organization.[7], [8] However, most work remains focused on knowledge modeling and extraction, with relatively limited discussion of subsequent QA reasoning mechanisms.

In the area of emergency QA, traditional methods rely heavily on rule templates, keyword matching, or simple fact queries against knowledge bases, answering basic questions such as "what happened at location X at time Y" but struggling with multi-hop causal analysis, cross-temporal statistics, and integrated decision-making tasks combining graph knowledge with regulatory documents[9], [10]. With the development of LLMs, researchers have begun exploring the integration of knowledge graphs and language models to improve QA flexibility and natural language expressiveness. However, in vertical-domain complex structured query scenarios, simultaneously ensuring precision, reliability, and interpretability remains an unresolved challenge.

2.2. LLM-Driven Knowledge Graph Query and QA

LLM-driven knowledge graph QA research has developed rapidly in recent years, with approaches broadly categorizable into three main paths. The first converts knowledge graph content into textual descriptions for LLMs to perform reasoning

and generation—simple to implement but prone to losing structural relational information. The second employs graph retrieval-augmented generation (Graph-RAG), retrieving relevant subgraphs, entities, or relations from the graph prior to generation as external knowledge input to enhance factual grounding. The third directly prompts LLMs to generate structured query statements such as SPARQL or Cypher, executed by graph databases.

Compared with pure text retrieval, structured queries are better suited for handling questions involving numerical filtering, temporal constraints, relation traversal, and aggregation statistics, making them attractive for vertical-domain knowledge services. In particular, within graph databases using property graph storage, Cypher offers strong expressive power to intuitively describe entity nodes, state nodes, and relationship paths. However, Text-to-Cypher is not always reliable: when the graph schema is complex, field naming is non-standard, or user questions contain ambiguous conditions, LLMs are prone to generating syntactically incorrect, field-mismatched, or semantically inconsistent queries.

To enhance tool-calling and multi-step reasoning capability, reasoning frameworks such as ReAct have been widely introduced into LLM systems. These methods organize model reasoning and tool invocation through a "think-act-observe" loop, enabling the system to continuously revise its strategy based on intermediate results. ReAct provides strong support for knowledge graph QA, but does not itself directly resolve "how to query the graph" or "how to organize large result sets"—problems that still require task-specific query mechanisms.

2.3. Limitations of Existing Work

Existing research provides an important foundation for disaster knowledge organization and graph QA, but several issues remain insufficiently addressed in the context of flood disaster spatiotemporal knowledge graph QA. First, dynamic querying and scripted querying each have advantages, yet effective collaboration between them is lacking. Fully relying on LLMs to generate Cypher offers strong adaptability but is susceptible to graph schema complexity, field naming ambiguity, and temporal constraint issues; fully relying on scripted queries is more reliable but struggles to cover open-ended problems. Second, existing methods underexploit differences in question types. Emergency QA involves both standardized fact queries and complex tasks such as causal chain tracing, temporal analysis, and graph-document joint decision-making, and a single-path processing approach cannot adequately accommodate these differences. Third, result management in multi-round reasoning remains weak. When large-scale query results are injected directly into the model context, reasoning costs increase and subsequent judgment is disrupted. Based on the above analysis, this paper designs its method from three aspects—dual-modal query collaboration, question type routing, and large result set management—to build a structured query framework better suited to flood disaster

emergency QA scenarios.

3. White-box/Black-box Dual-modal Structured Query Method

3.1. Knowledge Graph Organization Foundation and Problem Definition

This paper takes an already-constructed flood disaster spatiotemporal knowledge graph as its foundation, focusing on QA query mechanisms for state-centric graphs rather than elaborating on the graph construction process itself. To support fact extraction, trend analysis, and causal tracing in flood disaster emergency QA, the spatiotemporal knowledge graph employs a "base entity–state entity–state relation" organizational scheme. Unlike the traditional approach of attaching dynamic attributes directly to entity nodes, this graph abstracts change information from disaster processes into a state layer, enabling queries to center on "what state is an object in at a specific time."

Base entities represent relatively stable objective objects such as locations, events, and facilities. State entities describe the state information of base entities within specific time intervals, such as water level, rainfall, affected population, and economic loss. State relations express associations between different states, particularly causal influences, temporal succession, and spatial correlations. Under this organizational scheme, dynamic attribute queries no longer act directly on base entities but are mainly completed through state nodes and their relationships.

This graph organization directly influences subsequent query mechanism design. For current status queries, the system locates the state node corresponding to the target entity within a given time range and extracts its attribute values. For trend assessment, it retrieves attributes of the same type from multiple state nodes and constructs a time series. For causal analysis, it requires multi-hop traversal along state relations. The state layer is thus both the central carrier of disaster evolution knowledge and the primary operational object of dual-modal structured queries. Accordingly, this paper encapsulates standardized state queries as black-box scripts and, for complex open-ended questions, has the white-box module dynamically generate state-layer-oriented Cypher statements.

State-centric graph organization distributes temporal, spatial, and causal information from flood disasters across different representational layers. Base entities serve as spatial anchors; state entities record dynamic attributes—water level, rainfall, affected population, economic loss—for specific time intervals; state relations characterize causation, influence, temporal succession, and spatial correlation among states. This design shifts the query object from "what attributes does an entity have" to "what state is an entity in during a given period," better reflecting the continuous evolution of disaster processes.

In the graph construction phase, an "extract-and-simplify–map-and-complete" processing approach is adopted: an LLM extracts base entities, state entities, and

causal relations from disaster text, while temporal and spatial relations are automatically completed by a mapping algorithm according to ID rules and timestamps. This approach reduces the model's burden during extraction, providing a stable structural foundation for query design. State IDs encode entity, date, and state range information, making them directly usable as candidate filtering cues at query time.

The task addressed by this paper is: given a flood disaster emergency question posed by a user in natural language, the system retrieves and integrates relevant knowledge from the spatiotemporal knowledge graph and emergency response plan documents, outputting answers that are factually correct, interpretable, and capable of supporting decision-making. Compared with general QA, this task features heterogeneous knowledge sources, diverse question types, and highly variable result scales.

The core idea of white-box/black-box dual-modal querying is to select different execution paths based on question type. For standardized tasks with clearly specified entities, attributes, and temporal constraints, the system preferentially invokes black-box scripts. For questions requiring multi-condition combinations, causal chain tracing, or trend statistics, the white-box module dynamically generates Cypher queries. If a question involves normative judgments such as response levels and procedures, document retrieval is further incorporated. The entire process runs under ReAct reasoning to support multi-round observation and strategy adjustment; query results are organized through control-flow summaries and data-flow bypass.

3.2. Emergency Query Type Classification and Collaborative Routing Strategy

3.2.1. Emergency Query Type Classification

Table 1 presents the task classification for flood disaster emergency QA and the corresponding recommended query modes. Current status queries are better suited for stable execution by black-box scripts; causal analysis and trend assessment are better suited for dynamic generation and iterative refinement by the white-box module; integrated decision-making questions typically require invoking both graph facts and plan documents, making a hybrid mode more appropriate.

Table 1. Flood Disaster Emergency QA Task Types and Recommended Query Modes

Task Type	Semantic Feature	Typical Constraints	Recommended Mode	Example Question
Current Status Query	Query state attributes of a single entity/region at a specific time	Entity, attribute, and temporal constraint are explicit	Black-box scripted query	What was the water level of Panchang Reservoir on June 15, 2024 ?
Causal Analysis	Trace causal or impact chains between disaster states	Path length variable; relation types require filtering	White-box dynamic Cypher	What cascading impacts followed the landfall of Typhoon Koinu in 2023 ?

Task Type	Semantic Feature	Typical Constraints	Recommended Mode	Example Question
Trend Assessment	Retrieve state change sequences over a time range	Time range, attribute type, aggregation granularity	White-box + post-processing	How did the affected population in Nanning, Guangxi change from 2020 to 2023 ?
Integrated Decision-making	Form recommendations from graph facts and plan rules	Graph facts, plan provisions, response standards	Hybrid mode	Based on current flood conditions, what level of emergency response should be activated ?

For current status queries, questions have clearly specified entities, attributes, and temporal constraints that can be mapped to relatively stable structured query templates, making black-box script execution preferred. Causal analysis requires multi-hop traversal along causal relations with uncertain path lengths, favoring white-box dynamic Cypher generation. Trend assessment requires attribute extraction from multiple state nodes with temporal sorting and aggregation, typically accomplished through white-box queries with post-processing. Integrated decision-making requires both graph facts and unstructured plan knowledge, making a hybrid mode the typical approach. This classification directly serves the routing strategy: by identifying the question category, the system can pre-emptively determine appropriate tools and execution paths.

3.2.2. Collaborative Routing Strategy

Figure 1 illustrates the selection logic for white-box/black-box query modes. This logic uses heuristic judgment based on: whether the question can be mapped to a standard query template, whether it contains multi-condition combinations or multi-hop causal relations, and whether plan documents need to be incorporated. If the question is a standardized current status query with clearly specified entity, attribute, and temporal constraints, black-box scripts are preferentially invoked. If the question requires flexible traversal along causal chains, time series, or spatial hierarchies, white-box dynamic querying is prioritized. If the question requires both graph facts and plan rules, document retrieval is appended after structured querying, and the model integrates both result types within a unified reasoning chain. This collaborative routing strategy selects the more appropriate execution path based on task characteristics, balancing query reliability and adaptability to open questions.

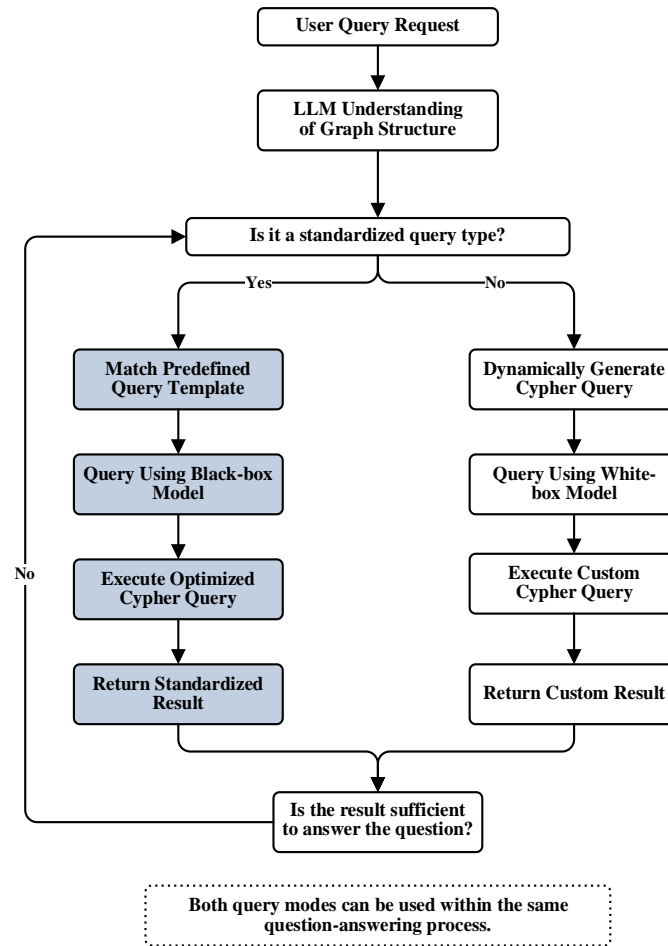


Figure 1. White-box/Black-box Query Mode Selection Logic

3.3. White-box/Black-box Query Execution Mechanisms

3.3.1. White-box Dynamic Cypher Query

The core of the white-box query module is the dynamic generation of Cypher query statements by an LLM based on its understanding of question intent and graph structure. This module is suitable for causal analysis, trend assessment, and complex questions involving multi-condition combination constraints. Its execution process advances progressively through a loop of structure awareness, query generation, result observation, and strategy refinement. Specifically, the system first provides the model with graph schema summaries, entity types, state node organization, and common query rules as needed. The model then generates a Cypher statement based on the user question and current observations, specifying filter conditions, sorting methods, and return fields. After query execution, the system feeds back a summary of result scale, field information, and sample records to judge whether current results satisfy question requirements. If results are empty, oversized, or misaligned with the original intent, query conditions are adjusted in the next

reasoning round.

In white-box querying, this paper follows a "broad-to-narrow" progressive strategy. Compared to adding multiple strong constraints at once, this strategy is better suited for complex emergency questions: the system first retrieves candidate states, then gradually refines the time range, attribute types, or causal paths based on observations. This reduces query failures caused by overly strict initial conditions, improving robustness under complex tasks.

To reduce the probability of the model generating incorrect queries, the white-box module adheres to several query constraint principles: preferentially filtering at the state layer, placing temporal constraints first, using OPTIONAL MATCH to extend attributes, and explicitly validating relation types. These rules enhance the model's adaptability to the spatiotemporal knowledge graph structure.

3.3.2. Black-box Scripted Query

The black-box query module encapsulates high-frequency, standardized, and structurally stable query tasks through predefined scripts. Unlike the white-box module, the black-box module completes query logic verification at the design stage, with input parameters and output fields both constrained, making it better suited for questions with clear facts and fixed patterns. This paper encapsulates high-frequency tasks—entity detail queries, time-series data extraction, causal chain tracing, and regional aggregation statistics—as black-box scripts. By fixing input parameters, constraining output formats, and building in data cleaning and aggregation logic, it improves execution reliability for standardized queries.

Black-box scripts' advantages lie primarily in verified query paths, high parameter standardization, and clear post-processing logic. However, their limitations are equally apparent: they can only cover question types foreseen at design time. When user questions exceed existing templates or require joint reasoning across multiple scripts, the black-box path's adaptability diminishes markedly. Therefore, the black-box module is not a replacement for the white-box module but complements it.

3.4. Data Flow and Control Flow Separation Mechanism

To formally describe the result management process, this paper expresses the query result size estimation as Equation (1):

$$S_{total} = \frac{S_{sample}}{n} \times N \quad (1)$$

Where S_{total} denotes the estimated total size of the complete result, S_{sample} denotes the serialized size of sample records, n denotes the number of sample records, and N denotes the total number of result records. When S_{total} exceeds the system-defined threshold, the complete result is no longer injected directly into the context; instead, it is stored in the data flow bypass, with only statistical summaries and file paths fed back to the model.

In the multi-round ReAct reasoning process, query result scales can vary enormously. Injecting all raw results directly into the model context not only consumes substantial context space but also makes the model more prone to distraction by irrelevant details. To this end, this paper defines the summary information participating in reasoning as the control flow, and the complete raw results as the data flow. The control flow includes result count, field names, time ranges, spatial ranges, representative samples, and result file locations; the data flow is saved as independent temporary files, readable by subsequent tools on demand. When query results are small and simple, the system returns the complete JSON to the model. When result scale exceeds the threshold, only record count, field meanings, spatiotemporal coverage, representative samples, and the complete result path are returned.

In concrete implementation, the system first samples list or dictionary results to estimate the serialized size of the complete result, then decides whether to return the complete JSON or a summary. The summary typically includes total record count, field names, temporal and spatial coverage ranges, the first several representative samples, and the complete result file location. After this processing, the model can determine whether the next step requires further filtering, aggregation, or transition to document retrieval, while the complete data remains available in the tool chain.

This data organization approach is particularly critical for trend assessment and integrated decision-making questions. Trend assessment often returns multiple state records spanning years or regions; integrated decision-making may simultaneously include graph results and plan fragments. The control flow summary retains "information necessary for the next reasoning step," while the data flow bypass retains "information that subsequent tools may continue to use." With the two separated, both the controllability and interpretability of the QA chain are improved.

4. Experiments and Results Analysis

4.1. Experimental Setup

Table 2 presents the composition of experimental data and the question set. To validate the proposed method, this paper constructs an experimental environment based on a flood disaster spatiotemporal knowledge graph and emergency response plan documents. The knowledge graph is stored in a property graph database, containing base entities, state entities, and their causal, temporal, and spatial association information. Plan knowledge is organized as document segments with vector indexes, used for supplementary retrieval in integrated decision-making questions.

Table 2. Experimental Data and Question Set Composition

Data/Task Category	Scale	Description
Base entity nodes	351	Administrative divisions, disaster events, reservoirs, rivers, monitoring stations, etc.
State entity nodes	577	State info (water level, rainfall, losses) for entities over specific time intervals
State relations	1,200	Temporal succession, causal influence, spatial association, etc.
Attribute records	1,537	Rainfall, water level, affected population, economic loss, relocated population, etc.
Causal analysis questions	10	Multi-hop causal chain tracing and impact analysis
Trend assessment questions	10	Temporal change, regional aggregation, and statistical analysis
Integrated decision-making questions	10	Combined graph facts and plan rule judgment
Total test questions	30	Each type includes multiple natural-language formulations

The experimental question set covers three complex task types--causal analysis, trend assessment, and integrated decision-making--totaling 30 test questions. Causal analysis questions primarily test the system's ability to trace multi-hop state relations and influence chains. Trend assessment questions examine cross-temporal state record extraction, attribute aggregation, and statistical analysis capabilities. Integrated decision-making questions require simultaneous integration of graph facts and plan rules. Current status queries, given their clearer structure, serve primarily as baseline validation for the standardized script path and are elaborated in case analysis. Each test question is judged based on graph query results, plan original text, and manually verified answers.

Experimental data come from a pre-constructed flood disaster spatiotemporal knowledge graph and its associated emergency response plan corpus. The graph side is stored as a Neo4j property graph, covering administrative divisions, disaster events, reservoirs, rivers, and monitoring stations, with state nodes recording water, rain, and disaster condition indicators across different time intervals. The document side establishes a vector index after segmenting emergency plans into semantic segments. The two knowledge sources are stored independently and combined on demand within the ReAct reasoning chain.

To reduce the impact of randomness on results, the three comparison methods use the same question set, graph data, and plan documents. For structured fact questions, graph return results and manually verified answers serve jointly as judgment criteria. For integrated decision-making questions, correctness of factual indicators, matching of plan basis, and consistency of final recommendations with rules are all checked.

4.2. Comparison Methods and Evaluation Metrics

Table 3 presents the comparison method configurations. The pure white-box method relies on LLM-generated dynamic Cypher and has strong question adaptability but relatively insufficient reliability in vertical-domain complex graph scenarios. The pure black-box method relies on predefined scripts with a more controllable execution process but cannot cover open-ended complex questions. The dual-modal method dynamically switches between white-box and black-box via a collaborative routing mechanism, incorporating document retrieval when necessary.

Table 3. Comparison Method Configurations

ID	Method	Query Mechanism	Tool Invocation	Characteristics
M1	Pure White-box	LLM dynamically generates Cypher for all questions	Single-path dynamic query	High flexibility; susceptible to schema understanding and field matching
M2	Pure Black-box	All questions mapped to predefined scripts	Fixed script invocation	Stable execution; insufficient coverage of open-ended questions
M3	Dual-modal	Routes among white-box, black-box, and document retrieval by question type	ReAct multi-round tool invocation	Balances standard query stability with complex query adaptability

Evaluation metrics include query success rate, answer accuracy, average reasoning rounds, and complex task stability. Query success rate measures whether the query process can normally return valid results. Answer accuracy measures the consistency of final answers with standard or manually judged results. Average reasoning rounds counts ReAct cycles per QA session. Complex task stability examines the system's ability to continuously output usable results in long causal chain, multi-constraint, and graph-document joint reasoning scenarios.

4.3. Overall Results Analysis

Table 4 presents results for different methods across causal analysis, trend assessment, and integrated decision-making tasks. Figure 2 further illustrates differences in answer accuracy. Current status queries primarily validate the standardized script path and are not included in the complex task statistics. Overall, the dual-modal structured query method performs more stably across the three complex tasks, achieving a combined query success rate of 100%, answer accuracy of 98.3%, and average reasoning rounds of 1.47.

Looking at overall results, the pure white-box method achieves query success rate and answer accuracy both at 93.3%, with average reasoning rounds of 1.57. The pure black-box method also achieves a query success rate of 93.3%, but answer accuracy drops to only 68.3%, and average reasoning rounds rises to 2.23. The dual-modal method achieves a 100% query success rate, 98.3% answer accuracy, and 1.47 average reasoning rounds across 30 test questions.

Task-by-task analysis reveals different error sources. In causal analysis tasks, all

three methods achieve high success rates, but the pure black-box method's answer accuracy is 90%--lower than the 100% of the pure white-box and dual-modal methods. The difference is most pronounced in trend assessment: the pure black-box method's answer accuracy drops to only 35%, mainly due to limited script coverage of attribute names, temporal granularity, and aggregation methods. Integrated decision-making places higher demands on graph-plan collaboration: both pure white-box and pure black-box achieve 80% query success rate, while the dual-modal method improves to 100%.

Table 4. Experimental Results Comparison Across Three Complex Task Types

Task Type	Method	Query Success Rate (%)	Answer Accuracy (%)	Avg. Reasoning Rounds
Causal Analysis	Pure White-box	100	100	1.8
	Pure Black-box	100	90	2.8
	Dual-modal	100	100	1.7
Trend Assessment	Pure White-box	100	100	1.0
	Pure Black-box	100	35	1.6
	Dual-modal	100	100	1.1
Integrated Decision-making	Pure White-box	80	90	1.9
	Pure Black-box	80	80	2.3
	Dual-modal	100	95	1.6
Overall	Pure White-box	93.3	93.3	1.57
	Pure Black-box	93.3	68.3	2.23
	Dual-modal	100	98.3	1.47

Note: Overall results are based on 30 test questions across causal analysis, trend assessment, and integrated decision-making. Current status queries are discussed in case analysis.

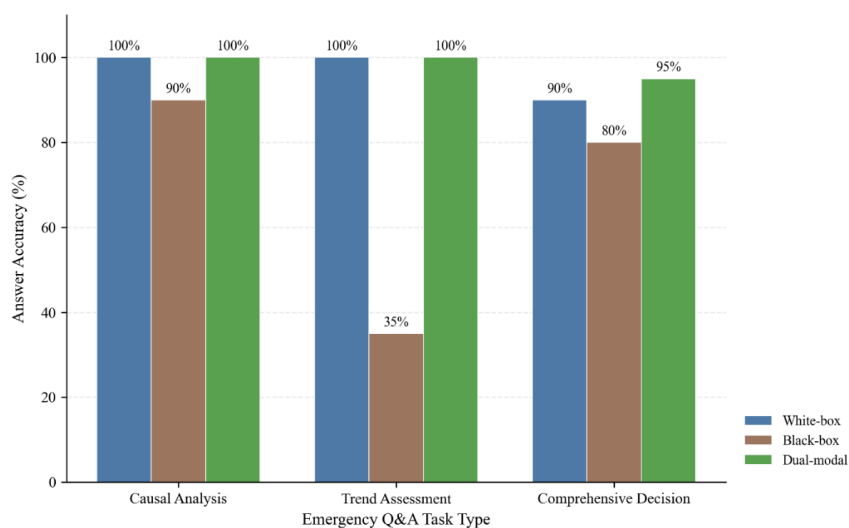


Figure 2. Answer Accuracy Comparison for Different Methods

Current status queries typically involve a single entity, attribute, and temporal constraint aligned with predefined script input formats. The black-box method reliably returns results at low reasoning cost. The dual-modal method tends to directly select the black-box script path for current status queries, reserving dynamic query generation for more complex questions.

In causal analysis tasks, both pure white-box and dual-modal methods achieve 100% query success rate and answer accuracy. The dual-modal method's average reasoning rounds of 1.7 is slightly lower than the pure white-box method's 1.8, suggesting scripts can reduce exploration costs. In trend assessment tasks, the pure black-box method's answer accuracy drops to 35%; the dual-modal method maintains 100% accuracy with 1.1 average reasoning rounds.

Integrated decision-making tasks most clearly demonstrate the necessity of the dual-modal method. Both pure white-box and pure black-box methods achieve 80% query success rate with answer accuracy at 90% and 80% respectively; the dual-modal method improves query success rate to 100% and achieves 95% answer accuracy. Graph queries provide factual evidence such as rainfall, water levels, and disaster status; document retrieval supplements response level determination rules; the model compares both within a unified reasoning chain.

5. Case Analysis

To further illustrate how the dual-modal structured query framework adapts to different task scenarios, this paper selects three representative question types--current status query, trend assessment, and integrated decision-making--for case analysis. Current status queries demonstrate the reliability of the black-box script path; trend assessment explains the reasons for the lower accuracy of the pure black-box method; integrated decision-making showcases the collaborative process between graph facts and plan rules.

In the current status query case "What was the water level of Panchang Reservoir on June 15, 2024?", the system first identifies the question as a standardized fact query with a single entity, single attribute, and single temporal constraint, and routes it to the black-box script. The script directly locates the state record of the target entity within the corresponding time range and returns the water level attribute value. Since this task type has a clear structure and unambiguous constraints, the black-box path achieves reliable answers at low reasoning cost--which is also why current status queries are not included in the complex task comparison statistics.

Trend assessment cases more clearly explain the pure black-box method's lower answer accuracy. Taking "What was the trend of affected population change in Nanning, Guangxi from 2020 to 2023?" as an example, the question not only requires locating relevant state nodes for Nanning City but also identifying the actual field names for the "affected population" attribute across different years and performing temporal sorting and trend summarization over multiple state records.

The pure black-box method can return records for some questions (query success rate remains 100%), but due to insufficient script coverage of attribute aliases, statistical calibers, and temporal granularity, it easily returns only single-year results, misses adjacent fields, or fails to complete trend summarization--resulting in answer accuracy of only 35%. The dual-modal method can first use scripts to probe available attributes and time ranges, then use white-box queries to generate state-layer filtering and aggregation statements, thereby maintaining 100% answer accuracy.

In the integrated decision-making case "Based on current flood conditions, what level of emergency response should be activated?", the system identifies that the question requires both current disaster indicators and reference to response rules in the emergency plan, and thus adopts a hybrid mode. Graph queries provide factual information such as current rainfall, water levels, and disaster status; document retrieval supplements response level determination rules; and the model compares facts and rules within a unified reasoning chain to generate response recommendations with supporting evidence. This case demonstrates that the dual-modal framework supports not only structured fact retrieval but also joint decision-making integrating normative knowledge with factual knowledge.

6. Conclusion

Targeting the challenge that dynamic queries offer strong adaptability but insufficient reliability while scripted queries are reliable but limited in coverage for flood disaster spatiotemporal knowledge graph QA, this paper proposes a white-box/black-box dual-modal structured query method. The method takes the ReAct reasoning mechanism as its scheduling core and, through white-box dynamic Cypher generation, black-box scripted queries, task-driven routing, and a data flow/control flow separation mechanism, constructs a unified QA framework for multiple types of emergency questions.

Experimental results show that across causal analysis, trend assessment, and integrated decision-making tasks, the dual-modal method achieves a 100% query success rate, 98.3% answer accuracy, and 1.47 average reasoning rounds. Compared with the pure white-box method, the proposed method improves query success rate in integrated decision-making tasks and slightly reduces overall reasoning rounds. Compared with the pure black-box method, it markedly improves answer accuracy, particularly alleviating insufficient script coverage in trend assessment and integrated decision-making. Current status query cases further demonstrate that standardized fact questions are better handled by black-box scripts, while complex open questions require white-box queries and mixed routing.

The paper still has certain limitations. The current routing strategy primarily relies on heuristic rules, leaving room for improvement in handling boundary-ambiguous questions. The white-box module remains relatively sensitive to graph schema

description quality, and graph-document joint reasoning in integrated decision-making can be further refined. Future research will focus on learnable routing mechanisms, complex spatiotemporal reasoning, and deeper graph-plan collaboration, to further improve the method's adaptability in open emergency QA scenarios.

Future improvement will focus not merely on expanding the number of scripts, but on enhancing the system's ability to judge query paths. By training lightweight routing models incorporating historical query logs, failure types, and task semantic features, the dual-modal query framework could advance from heuristic scheduling toward data-driven adaptive scheduling.

References

- [1] A. Hogan et al., "Knowledge Graphs," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–37, 2021.
- [2] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022, doi: 10.1109/TNNLS.2021.3070843.
- [3] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, in NIPS '20. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 9459–9474. Accessed: Mar. 09, 2026. [Online]. Available: <https://dl.acm.org/doi/10.5555/3495724.3496517>
- [4] N. Bhutani, X. Zheng, and H. V. Jagadish, "Learning to answer complex questions over knowledge bases with query composition," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, in CIKM '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 739–748. doi: 10.1145/3357384.3358033.
- [5] B. G. Ascoli, Y. S. R. Kandikonda, and J. D. Choi, "ETM: Modern insights into perspective on text-to-SQL evaluation in the age of large language models," Jun. 16, 2025, arXiv: arXiv:2407.07313. doi: 10.48550/arXiv.2407.07313.
- [6] S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=WE_vluYUL-X
- [7] M. Chen, Z. Tao, W. Tang, T. Qin, R. Yang, and C. Zhu, "Enhancing emergency decision-making with knowledge graphs and large language models," *Int. J. Disaster Risk Reduct.*, vol. 113, p. 104804, Oct. 2024, doi: 10.1016/j.ijdr.2024.104804.
- [8] W. Chen and J. Fang, "Optimizing AI-driven disaster management through LLMs," Jul. 17, 2024, Preprints: 2024071446. doi: 10.20944/preprints202407.1446.v1.
- [9] L. Cai, X. Mao, Y. Zhou, Z. Long, C. Wu, and M. Lan, "A Survey on Temporal Knowledge Graph: Representation Learning and Applications," *ArXiv Prepr. ArXiv240304782*, 2024, [Online]. Available: <https://arxiv.org/abs/2403.04782>
- [10] B. Cai, Y. Xiang, L. Gao, H. Zhang, Y. Li, and J. Li, "Temporal Knowledge Graph Completion: A Survey," *ArXiv Prepr. ArXiv220108236*, 2022, [Online]. Available: <https://arxiv.org/abs/2201.08236>