algorithms

MDPI

Article

CausalAgent: A Hierarchical Graph-Enhanced Multi-Agent Framework for Causal Question Answering in Production Safety Accident Reports

Tianyi Wang [1], Tao Shen [1], Zhiyuan Zhang [1], Shuangping Huang [1], Huiguo He [1], Qingguang Chen [2,3,*] and Houqiang Yang [2,3]

[1] School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China; 202520111489@mail.scut.edu.cn (T.W.); 202521065619@mail.scut.edu.cn (T.S.); eezzy@mail.scut.edu.cn (Z.Z.); eehsp@scut.edu.cn (S.H.); hehuiguo@scut.edu.cn (H.H.)
[2] Guangdong Academy of Safety Production and Emergency Management Science and Technology, Guangzhou 510060, China; 13791528158@126.com
[3] Guangdong Provincial Science and Technology Collaborative Innovation Center for Production Safety, Guangzhou 510060, China
* Correspondence: 13316206596@126.com

Abstract

Accident reports provide a detailed account of environmental causes, unsafe human behaviors, and subsequent chain reactions. These records serve as essential resources for analyzing accident mechanisms and exploring potential risk patterns within production safety processes. Currently, Graph based Retrieval-Augmented Generation (RAG), which integrates Large Language Models (LLMs) with Knowledge Graphs (KGs), has emerged as a leading approach for complex causal question answering over extensive unstructured accident documentation. However, the application of this technology in the production safety domain still encounters two primary challenges. First, knowledge graph construction using a single granularity fails to capture fine-grained case details and macro-level standard systems. Second, traditional one-step retrieval paradigms lack the capacity to track deep causal chains or interpret the complex logic of multi-factor coupling. To address these limitations, we propose CausalAgent, a hierarchical graph-enhanced multi-agent framework for causal question answering in production safety accident reports. This framework innovatively combines a Hierarchical Causal Graph (HC-Graph) and a Multi-Agent Collaborative Reasoning (MACR) mechanism. Specifically, the HC-Graph employs a two-layer architecture that links a fine-grained instance layer with a national standard causation layer to resolve conflicts in semantic granularity. The MACR mechanism converts complex natural language queries into executable structured queries and logic verification steps through the sequential cooperation of four specialized agents, namely the Graph Parsing Agent, the Problem Analysis Agent, the Query Generation Agent, and the Reasoning Insight Agent. CausalAgent enables in-depth mining of accident causation mechanisms and provides scientific, robust and interpretable intelligent support for data-driven risk assessment and emergency decision-making. Experiments on real-world accident datasets demonstrate that CausalAgent achieves a 100.0% query execution rate and an 87.3% reasoning accuracy, outperforming the SOTA baseline by 45.2% in terms of absolute accuracy.

Keywords: retrieval-augmented generation; causal question answering; hierarchical causal graph; multi-agent collaboration; production safety accident report

Academic Editor: Xiaoqiang Hua

Received: 8 March 2026
Revised: 28 April 2026
Accepted: 28 April 2026
Published: 2 May 2026

1. Introduction

Accident reports are highly valuable data assets for production safety and emergency management [1]. These documents provide comprehensive documentation of diverse environmental triggers, unsafe human behaviors, and the intricate chain reactions that culminate in accidents, serving as the fundamental basis for analyzing accident causation mechanisms [2]. Such analysis enables researchers to identify latent risk patterns and profound causal correlations that often remain obscured in individual case studies [3,4]. In recent years, the advancement of artificial intelligence has enabled researchers to transcend the constraints of manual analysis by rapidly and efficiently transforming complex, unstructured raw text into actionable high-value safety evidence [5–7]. This paradigm shift from traditional manual verification to automated knowledge discovery provides empirical data support for risk prevention and control, significantly improving the overall effectiveness of safety governance.

In the field of artificial intelligence, Large Language Models (LLMs) have become a pivotal technology for automated causal reasoning and question answering due to their superior natural language understanding capabilities [8–12]. To address the challenges of knowledge gaps and factual hallucinations in specialized domains, Retrieval-Augmented Generation (RAG) [13–15] has emerged as a dominant paradigm. By integrating external knowledge bases, this approach enables models to ground their responses in retrieved evidence [16]. Within this framework, Knowledge Graph-based RAG (KG-RAG) is particularly effective because it organizes domain knowledge into structured entities and relations, which captures explicit logical connections that are often lost in unstructured vector spaces [17–19]. Such structured modality facilitates precise relational reasoning and has demonstrated success in specialized domains, including maritime safety [20] and the process industries [21,22].

Existing KG-RAG reasoning frameworks mainly fall into two categories [23]: graph-search-based methods, which retrieve evidence by following abstract relational paths [24–27], and semantic-parsing-based methods, which translate natural language questions into executable structured queries or interface calls [28–31]. Although these paradigms have achieved encouraging progress, their direct application to causal question answering in production safety remains limited by two fundamental challenges.

The first challenge lies in semantic granularity. Accident causation analysis requires both fine-grained case evidence for traceability and standardized abstractions for large-scale statistical reasoning. For example, in a typical chemical accident, an explosion may be caused by material leakage from aging flange gaskets and ignited by unauthorized hot work. If the graph is built only from raw instance descriptions, it preserves details such as “aging flange gasket” and “Tank 3 leakage,” but suffers from sparsity and weak generalizability. In contrast, if these details are mapped only to coarse-grained standards such as GB/T 13861-2022 [32], the graph may retain categories like “equipment and facility defects” or “unsafe operation,” while losing the site-specific evidence needed for accident tracing and liability analysis. Existing single-granularity representations therefore struggle to support both precise causal tracing and macro-level safety analysis.

The second challenge lies in deep causal reasoning. Production safety accidents rarely result from a single factor; instead, they typically emerge from the interaction of environmental hazards, equipment failures, unsafe behaviors, and latent management deficiencies. In the above example, the final explosion is not explained solely by leakage or ignition, but by the coupling between equipment degradation, flammable material release, illegal hot work, and underlying management failures such as inadequate work-permit control. Practical safety question answering thus requires models to recover long-range causal dependencies from surface outcomes to root causes. However, existing one-shot

retrieval paradigms often capture only salient local facts, while failing to reconstruct the full causal chain linking latent organizational defects, direct triggering behaviors, and final accident consequences. This substantially limits their usefulness for safety supervision, risk diagnosis, and early warning.

To overcome these two key difficulties, we propose CausalAgent(as shown in Figure 1), a hierarchical knowledge graph multi-agent reasoning framework for causal question answering in production safety accident reports. CausalAgent integrates a Hierarchical Causal Graph (HC-Graph) with a Multi-Agent Collaborative Reasoning (MACR) mechanism to facilitate the in-depth analysis of large-scale accident reports through natural language interaction. Specifically, we address the semantic granularity trade-off through the HC-Graph, which utilizes a dual-layer architecture to bridge the gap between heterogeneous case descriptions and standardized safety taxonomies. In this hierarchical structure, the instance layer preserves the high-fidelity, fine-grained details necessary for understanding specific accident contexts, while the national standard causation layer provides a unified conceptual framework for grounding case-specific insights in macro-level regulatory consistency. Additionally, the complexity and opacity of reasoning over large-scale reports are tackled through the MACR mechanism, which decomposes the reasoning process into a modular "Chain-of-Thought" across four specialized agents to enhance reliability and controllability. By carefully allocating cognitive responsibilities, the Graph Parsing and Problem Analysis agents handle the decoupling of natural language ambiguity from logical intent, the Query Generation agent ensures technical precision by translating intent into executable code, and the Reasoning Insight agent validates causality and provides the final layer of verification. This modular design transforms complex accident analysis into a traceable and interpretable workflow, ensuring every step of the reasoning path is transparent and consistent.

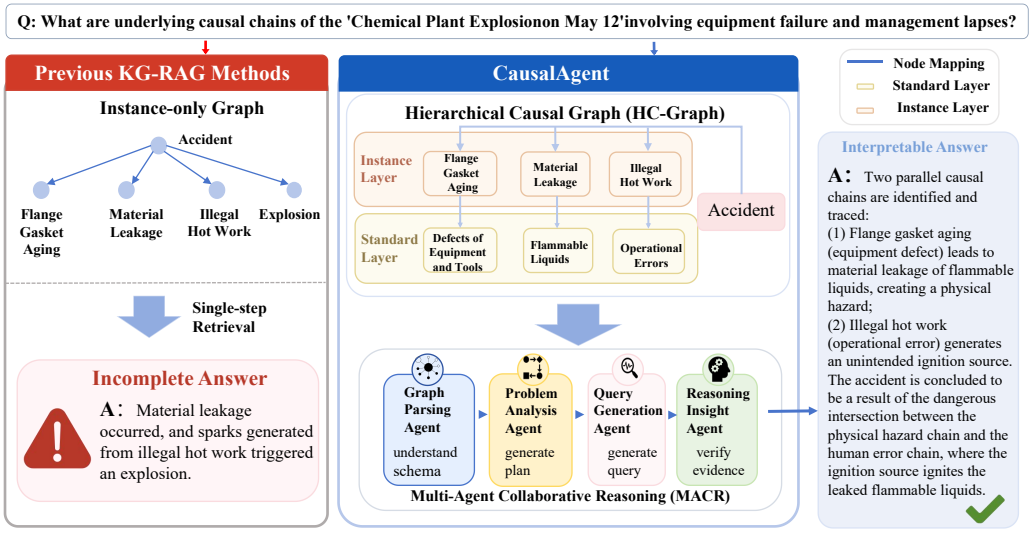

Figure 1. A comparison between previous KG-RAG methods and the proposed CausalAgent for accident causation analysis. Previous methods (**left**) often rely on single-granularity instance graphs and one-step retrieval, leading to incomplete answers. In contrast, CausalAgent (**right**) introduces a Hierarchical Causal Graph (HC-Graph) that bridges fine-grained instances with national standard layers. It leverages a Multi-Agent Collaborative Reasoning (MACR) mechanism to decompose complex queries and trace coupled causal chains, resulting in more comprehensive and interpretable answers.

The main contributions of this work are summarized as follows:

- We propose the CausalAgent framework. To address the limitations of existing RAG methods in processing complex causal logic, we establish a novel reasoning paradigm that integrates explicit hierarchical graphs with multi-agent collaboration. This ap-

proach enables in-depth semantic analysis and precise causal question answering for large-scale unstructured accident reports.
- We develop the HC-Graph and the MACR mechanism. The HC-Graph fuses instance-level details with national standard classifications to resolve semantic granularity conflicts effectively. Additionally, the MACR mechanism decomposes complex safety analysis tasks into a structured and controllable collaborative workflow, which enhances the reliability and interpretability of the reasoning process.
- Extensive evaluations on real-world accident datasets demonstrate that CausalAgent significantly outperforms mainstream RAG baselines, yielding a 49% improvement in accuracy. These findings provide robust and scientific support for large-scale risk assessment and policy optimization in production safety accident analysis.

2. Related Work

This section reviews three research areas foundational to the CausalAgent framework: knowledge graph construction in the safety domain, KG-RAG, and Large Language Model (LLM)-driven multi-agent systems.

2.1. Knowledge Graph Construction in the Safety Domain

Transforming unstructured accident reports into structured knowledge is essential for automated safety analysis. Early methods relied on traditional supervised learning techniques for entity and relation extraction [5,7]. Recently, Information Extraction utilizing LLMs has become the mainstream approach [10,33]. This advancement has facilitated the development of event logic graphs and causal chain graphs to capture accident evolution dynamics [3]. However, existing graph construction methods face a significant semantic granularity challenge. Flat graphs map text directly to nodes (e.g., "Valve A Leakage"). This fine-grained representation preserves instance details but lacks mapping to standardized concepts like "equipment failure". Consequently, this leads to graph sparsity and hinders macro-level statistical analysis. Constructing a hierarchical graph that integrates both instance details and standard classifications remains an unresolved issue.

2.2. Knowledge Graph-Based Retrieval-Augmented Generation

Traditional RAG systems alleviate LLM hallucinations by retrieving document chunks via vector similarity [13]. However, similarity matching struggles with logic-intensive queries that require cross-document aggregation. Integrating Knowledge Graphs (KGs) provides structural support to address this limitation. Current KG-RAG technologies primarily follow two paradigms. The first is Graph Search-based methods, such as GraphRAG [17] and RoG [24]. These methods perform subgraph sampling, linearize the retrieved triples, and feed them into the LLM context. They enhance contextual awareness but remain constrained by the LLM context window and struggle with complex statistical questions. The second is Semantic Parsing-based methods, such as StructGPT [28] and ChatKBQA [29]. These frameworks convert natural language queries into executable logical forms like Cypher or SPARQL. By offloading the reasoning process to a deterministic database engine, Semantic Parsing-based methods guarantee the accuracy of statistical queries. Nevertheless, these semantic parsing methods suffer from poor robustness. Generating complex query statements requires a deep understanding of the graph schema. When facing heterogeneous graphs with numerous relations, a single LLM frequently generates incorrect query code due to relation confusion.

2.3. LLM-Driven Multi-Agent Systems

To tackle complex tasks, research is shifting from single LLMs to Multi-Agent Systems (MAS). By simulating human collaboration, MAS decomposes intricate problems into

manageable subtasks such as planning, execution, and reflection [34]. Frameworks like MetaGPT [35] automate workflows through defined roles. In question-answering applications, multi-agent frameworks enhance response reliability by assigning distinct agents to handle knowledge retrieval, verification, and output generation [36]. Despite these advancements, most existing MAS frameworks are general-purpose and lack optimization for production safety accident analysis. In accident analysis, reasoning requires not only task decomposition but also bridging the semantic gap between non-professional queries and complex graph structures. Furthermore, the system must interpret discrete statistical data from a safety engineering perspective. Currently, there is a lack of a specialized multi-agent framework that integrates graph schema perception, query generation, and domain-specific intelligence analysis. Our proposed CausalAgent fills this gap by combining hierarchical graph construction, semantic-parsing-based reasoning, and a multi-agent collaboration mechanism to ensure both accuracy and interpretability. A capability-level comparison with representative baseline paradigms is summarized in Table 1.

Table 1. Comparison of representative methods for production safety accident analysis from the perspective of reasoning mechanisms and analytical capabilities. ✓ denotes supported capability, △ denotes partially supported capability, and × denotes unsupported capability.

Method	Dual-Granularity Representation	Multi-Agent Collaboration	Macro-Statistical Analysis	Micro-Level Traceability	Multi-Step Verification	Causal Interpretability
Embedding RAG [13]	×	×	×	△	×	×
Graph Search-based KG-RAG [17,24]	×	×	△	✓	×	△
Semantic Parsing-based KG-RAG [28,29]	×	×	△	△	×	△
CausalAgent (Ours)	✓	✓	✓	✓	✓	✓

3. Methodology

This section details the CausalAgent framework, which consists of three core components. First, Section 3.1 formally defines the causal question answering task for production safety accident reports. Section 3.2 introduces the HC-Graph, which balances instance-level details with national safety standards through a dual-layer architecture. Subsequently, Section 3.3 describes the MACR mechanism, which coordinates four specialized agents to execute task decomposition, query generation, and result synthesis. Together, HC-Graph and MACR modules enable CausalAgent to transform unstructured accident narratives into verifiable and interpretable safety insights. The overall architecture and workflow of the proposed framework are illustrated in Figure 2.

3.1. Problem Formulation

Causal Question Answering (CQA) in the production safety domain is regarded as a multi-dimensional association mining and causal reasoning task. It employs Large Language Models (LLMs) as semantic parsers to extract multi-dimensional safety elements—including accident types, industries, locations, and causal factors—from expert-authored unstructured accident reports. The task focuses on uncovering latent associations among safety elements to support both macro-level statistical analysis and micro-level causal tracing, thereby providing quantifiable and traceable insights for risk assessment.

Formally, let the accident report corpus be $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$. For each report $d_i \in \mathcal{D}$, the extracted multi-dimensional safety elements constitute a set $\mathcal{E}_i = \{T_i, I_i, L_i, C_i\}$. To facilitate structured reasoning, we represent these associations as a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, where the vertex set $\mathcal{V}$ is mapped from the elements in $\mathcal{E}$, and $\mathcal{R}$ denotes the set of logical relations among them.

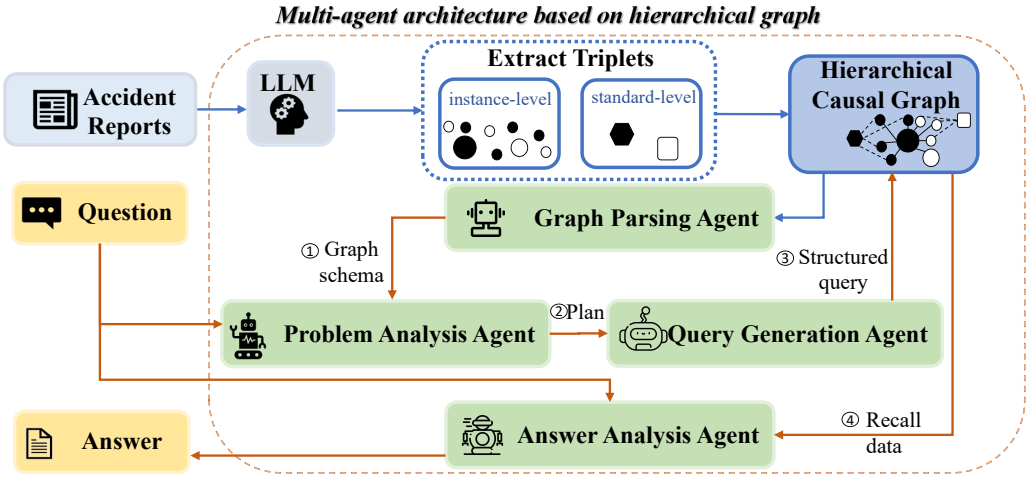

Figure 2. Architecture of the CausalAgent. This framework innovatively introduces the Hierarchical Causal Graph (HC-Graph) and the Multi-Agent Collaborative Reasoning (MACR) mechanism. HC-Graph integrates a fine-grained instance layer with a national standard causation layer to resolve semantic granularity conflicts. The MACR mechanism facilitates structured reasoning through the sequential collaboration of four specialized agents: Graph Parsing, Problem Analysis, Query Generation, and Reasoning Insight.

Given a natural language query Q, the CQA task is formulated as identifying a reasoning mapping Ψ that generates a comprehensive answer A based on the query and the knowledge graph $\mathcal{G}$:

$$A = \Psi(Q, \mathcal{G}) \tag{1}$$

where A encapsulates both quantifiable statistical insights and traceable causal evidence. The objective is to ensure that the mapping Ψ effectively retrieves and aggregates relevant information from the structured representation $\mathcal{G}$ to provide accurate and interpretable safety analysis.

3.2. Hierarchical Causal Graph

A fundamental challenge in constructing knowledge graphs for production safety lies in balancing the granular details of accident cases with the efficiency of systematic statistical analysis. Causal graphs directly extracted by Large Language Models (LLMs) frequently exhibit a granularity mismatch: fragmented instance-level descriptions impede large-scale aggregation, whereas overly broad classifications result in the loss of critical technical specifics. The HC-Graph implements the hierarchical architecture $\mathcal{G}$ to bridge this gap, semantically aligning accident data with national production safety standards while preserving essential instance-level details. The HC-Graph construction pipeline is shown in Figure 3, which maps case-specific attributes from unstructured reports to standardized categories to facilitate macro-level semantic aggregation while preserving micro-level technical details.

The construction of HC-Graph is guided by a differentiated modeling strategy over the dimensions of the element set $\mathcal{E}$. Specifically, dimensions such as *Industry* and *AccidentType* are governed by clear, mutually exclusive national standards, namely GB/T 4754-2017 [37] and GB 6441-2025 [38]. These dimensions therefore admit direct modeling as nodes in the standard layer $\mathcal{V}_{std}$. In contrast, the *Causation* and *Location* dimensions are inherently more heterogeneous and require hierarchical abstraction in order to reconcile fine-grained case evidence with macro-level statistical analysis. To this end, we introduce explicit

mapping functions to bridge the semantic gap between instance-level observations and standardized representations:

$$\phi_c : \mathcal{C}_{\text{inst}} \to \mathcal{C}_{\text{std}}, \quad \phi_l : \mathcal{L}_{\text{inst}} \to \mathcal{P} \tag{2}$$

where $\mathcal{C}_{\text{inst}} \subset \mathcal{V}_{inst}$ denotes raw causal instances (e.g., "Valve A Leakage"), $\mathcal{C}_{\text{std}} \subset \mathcal{V}_{std}$ denotes standardized causal categories (e.g., "Equipment Failure" under GB/T 13861 [32]), and $\mathcal{P}$ denotes provincial-level administrative units. In the graph schema, these mappings are instantiated through the *classified_as* and *located_in* relations, respectively.

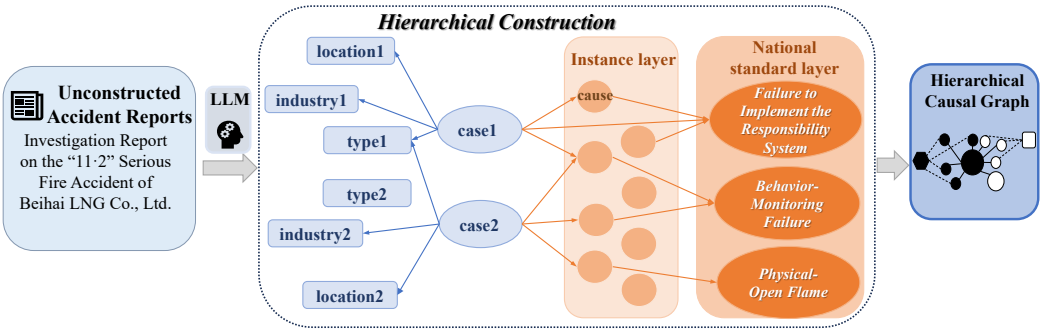

Figure 3. Overview of the HC-Graph construction pipeline. Case-specific attributes from unstructured reports are mapped to standardized categories to facilitate macro-level semantic aggregation while preserving micro-level technical details.

At the structural level, HC-Graph organizes the relations in $\mathcal{R}$ into a complete accident logic chain. The *AccidentCase* node serves as the central hub, linking contextual attributes such as industry and location through occurrence–attribution relations, while connecting internal accident mechanisms through *has_cause* relations. Within the causal subgraph, the *triggers* relation explicitly models the sequential evolution between causal factors, thereby capturing the dynamic progression of accident formation. By further incorporating *SpecificConsequence* nodes, the graph represents the full trajectory from root causes to eventual severity outcomes. A comparative illustration of the instance-level causal graph, standard-level causal graph, and the proposed HC-Graph is presented in Figure 4. This hierarchical and relational organization enables the reasoning function Ψ to traverse seamlessly between macro-level patterns and micro-level evidence, thereby providing a principled structural foundation for Multi-Agent Collaborative Reasoning (MACR). Distinct from logical decision structures such as Ordered Binary Decision Diagrams (OBDDs) or Linear Secret Sharing Schemes (LSSSs) used in cryptographic access control, HC-Graph is a directed multi-relational knowledge representation. Its hierarchical nature is designed for semantic abstraction (e.g., mapping specific 'Valve Leakage' to 'Equipment Failure') rather than Boolean logic evaluation or permission management. More implementation details of HC-Graph construction are provided in Appendix B.

3.3. Multi-Agent Collaborative Reasoning

Transforming natural language queries into executable graph operations presents a significant challenge, even with a well-constructed causal graph. The single-step generation paradigm couples graph comprehension, intent parsing, and code synthesis. This often leads to execution errors caused by language model hallucinations or the misinterpretation of graph structures. To address this challenge, we propose the MACR framework. This framework decomposes the reasoning process into specialized and constrained subtasks, ensuring that each analytical stage is strictly grounded in the structural characteristics of the knowledge graph. As illustrated in Figure 5, the MACR framework distributes cognitive responsibilities across four specialized agents to guarantee the reliability and

interpretability of the results: The Graph Parsing Agent (Section 3.3.1) extracts the metadata of the HC-Graph to provide a reliable structural reference. The Problem Analysis Agent (Section 3.3.2) filters out unsupported queries and decomposes complex questions into solvable logical steps. The Query Generation Agent (Section 3.3.3) converts these steps into executable graph query code and retrieves the data. Finally, the Reasoning Insight Agent (Section 3.3.4) parses the raw execution results and synthesizes a coherent natural language response with an explicit deductive process. By decoupling graph comprehension from code generation, MACR robustly executes multi-step analytical queries and ensures that every insight is traceable.

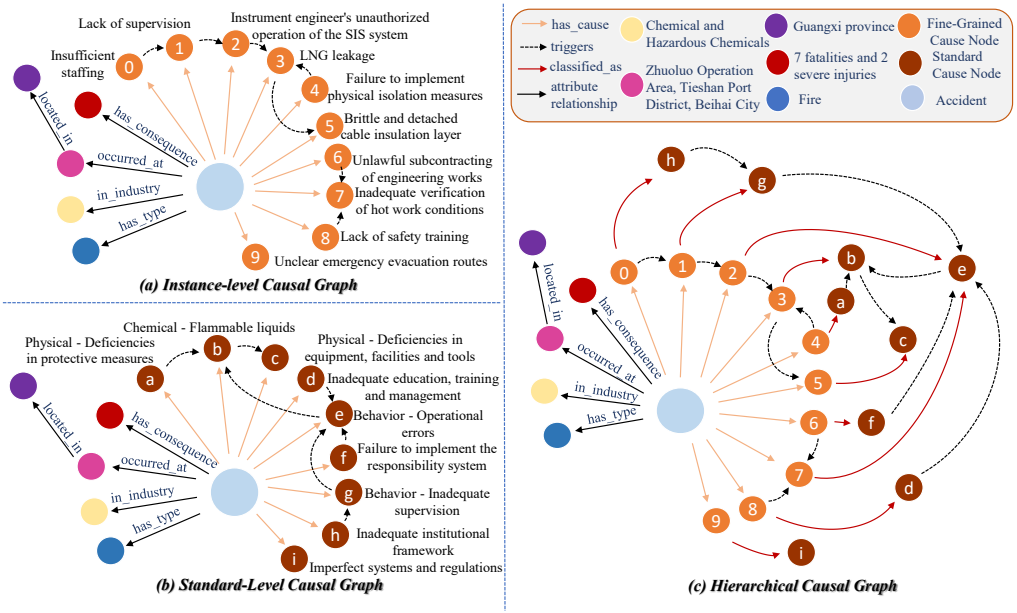

Figure 4. Comparative analysis of (**a**) instance-level, (**b**) standard-level, and (**c**) the proposed HC-Graph. The HC-Graph utilizes the hierarchical mapping mechanism to enable both micro-level traceability and macro-level pattern extraction.

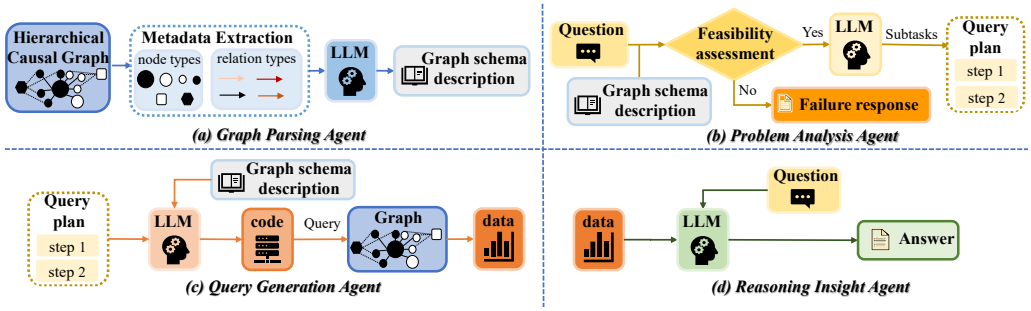

Figure 5. Architecture of the Multi-Agent Collaborative Reasoning (MACR) Workflow. (**a**) **Graph Parsing Agent**: Extracts the structural schema of the graph and transmits it to downstream modules. (**b**) **Problem Analysis Agent**: Evaluates query feasibility and plans the logical execution workflow based on the graph schema. (**c**) **Query Generation Agent**: Synthesizes executable queries according to the planned workflow and retrieves results. (**d**) **Reasoning Insight Agent:** Synthesizes the final answer through logical deduction over the discrete query results.

3.3.1. Graph Parsing Agent

The efficacy of downstream agents relies on a comprehensive understanding of the graph schema. We introduce the Graph Parsing Agent as the fundamental perception unit of the framework. During system initialization, this agent traverses the graph to extract metadata and synthesize a standardized textual description of the topology and attribute definitions. This structured semantic prompt equips subsequent modules with prior

knowledge regarding entity types, relation types, and data distribution. Consequently, it constrains all query tasks to defined data boundaries, thereby bolstering the interpretability and transferability of the automated reasoning process. For more details on Graph Parsing Agent, see Appendix A.1.

3.3.2. Problem Analysis Agent

Direct execution of natural language queries frequently fails due to misalignments between user intentions and the structural constraints of the knowledge graph. Furthermore, complex analytical questions often exceed the capacity of single-step reasoning. To address these issues, the Problem Analysis Agent functions as a semantic planner. It leverages the schema provided by the Graph Parsing Agent to optimize the query processing workflow through a two-phase strategy comprising feasibility assessment and task decomposition.

In the first phase, the agent conducts a feasibility assessment to verify whether the query falls within the analytical scope of the graph. The large language model cross-validates the entities in the user query against the predefined schema. If the requested information exceeds the graph coverage, the agent identifies the domain conflict and generates an explanatory response rather than attempting an invalid retrieval. For example, if a user queries "traffic accidents in California" but the graph only covers Chinese administrative regions, the agent will detect this mismatch and terminate the process to prevent hallucinated execution instructions.

In the second phase, for feasible queries, the agent proceeds with task decomposition. It maps the complex analytical goals of the user into a series of executable graph operations layer by layer. The agent identifies core constraints, locates corresponding node types, and plans a logical execution chain. For instance, given the query "What is the third most frequent national standard cause type in the logistics industry?", the agent locates the "Logistics" entity via the "Industry" node type. It then plans the traversal path: Industry $\rightarrow$ occurs_in $\rightarrow$ Accident $\rightarrow$ has_cause $\rightarrow$ Cause $\rightarrow$ classified_as $\rightarrow$ Standard Cause. Finally, it specifies the statistical operations for frequency counting and ranking. This mechanism converts high-level intentions into a rigorous execution blueprint for the subsequent Query Generation Agent. For more details on Problem Analysis Agent, see Appendix A.2.

3.3.3. Query Generation Agent

The logical execution plan generated by the Problem Analysis Agent cannot interact directly with the database. To bridge the semantic gap, the Query Generation Agent acts as the code synthesizer of the framework. Based on the planned workflow and the graph schema, this agent leverages the coding capabilities of large language models to dynamically construct executable graph queries. The generated code operates within a controlled local sandbox to perform subgraph traversal, path finding, and statistical aggregation directly on graph objects. This mechanism ensures the accuracy and reproducibility of the retrieval results, mitigating the precision loss inherent in traditional vector retrieval methods. For more details on Query Generation Agent, see Appendix A.3.

3.3.4. Reasoning Insight Agent

Although the Query Generation Agent accurately retrieves factual data, its output typically consists of discrete statistical results or isolated case summaries. These raw outputs lack narrative coherence and logical synthesis. To achieve expert-level analytical understanding, we introduce the Reasoning Insight Agent as the terminal component of the framework.

Through prompt engineering, this agent is designated as a domain expert in emergency management. It receives the original user query and the structured retrieval results as inputs. The agent is explicitly constrained to answer the query directly, cite the retrieved

data as evidence, and deeply analyze the underlying causal patterns. For instance, if a user requests a comparative analysis of accident patterns between Guangdong and Jiangsu provinces, the raw graph output might only yield statistical aggregates. The Reasoning Insight Agent synthesizes these discrete points into a comprehensive report. It highlights that Guangdong faces dynamic operational risks (e.g., transportation accidents caused by overspeeding), whereas Jiangsu encounters static structural hazards (e.g., construction accidents involving illegal demolition). Furthermore, the agent correlates these findings with regional economic profiles. Through graph-based reasoning, this agent transforms discrete data points into an integrated, semantically rich analytical narrative, providing actionable decision support. For more details on Reasoning Insight Agent, see Appendix A.4.

3.3.5. Agent Communication Mechanism

To ensure robust and interpretable collaborative execution, the four agents communicate via a sequential prompt-passing mechanism rather than low-level shared memory or message queues. Specifically, the output of each upstream agent is encapsulated into a structured context (e.g., JSON object) and directly injected into the prompt of the downstream agent as verified evidence. For example, the graph schema extracted by the Graph Parsing Agent serves as the foundational context for the Problem Analysis Agent, and the generated query plan is passed to the Query Generation Agent for executable code synthesis. This structured prompt-passing paradigm achieves strong modularity and decoupling among agents. It also makes the entire reasoning chain fully observable and traceable, avoiding information loss or semantic drift during multi-step causal inference.

4. Experiments

We conduct a systematic evaluation to verify the effectiveness, robustness, and limitations of the proposed CausalAgent for causal analysis of production safety accident reports.

This section is organized as follows. Section 4.1 describes the experimental setup, including the data sources, the construction of benchmark questions, and the configuration of the HC-Graph. Section 4.2 compares the proposed framework with mainstream baseline methods across three paradigms: vector-based RAG, Graph Search-based KG-RAG, and Semantic Parsing based KG-RAG. Through quantitative metrics and qualitative case studies, we evaluate the performance of each method on complex multi-step reasoning queries. Section 4.3 investigates the influence of various backbone Large Language Models (LLMs) on the overall efficacy of the proposed method. Section 4.4 reports ablation studies to quantify the contributions of the HC-Graph and the MACR mechanism. Finally, Section 4.5 analyzes failure cases to identify remaining challenges and areas for improvement.

4.1. Experimental Setting

This section details the dataset and benchmarks utilized for evaluation. We introduce the data source, the construction of the HC-Graph and question–answer pairs, and the evaluation metrics, which together provide a reproducible foundation for assessment. To facilitate the reproducibility of our research, we provide Supplementary Materials including the source code and processed datasets.

4.1.1. Data Source

To construct the causal knowledge graph, we utilize 1872 official production safety accident reports retrieved from the public announcement section of the Ministry of Emergency Management of the People's Republic of China [39]. These reports provide comprehensive information, including accident progression, emergency response measures, casualty statistics, economic losses, and officially identified causal factors.

4.1.2. HC-Graph Construction

We construct an Instance-level Causal Graph, a Standard-level Causal Graph, and the proposed HC-Graph for comparison. Following the strategy in Section 3.2, the HC-Graph processes the 1872 reports to extract multi-level semantic information. This includes temporal and spatial contexts, industry classifications, accident types, consequences, and standardized causal categories.

The extraction of instance-level causal factors and their subsequent mapping to national classification standards are fully automated via a pipeline powered by GPT-5-mini. This automation ensures scalability for large-scale accident report analysis. To mitigate hallucination risks and ensure factual consistency in high-stakes safety scenarios, we implemented a two-stage verification protocol.

We implemented a two-stage verification protocol to evaluate the reliability of our automated graph construction pipeline. Detailed test procedures, scoring criteria, quantitative results and score distributions are provided in Appendix E.

4.1.3. Question–Answer Pair Construction

To validate the effectiveness of the proposed CausalAgent, we develop a benchmark dataset consisting of 100 question–answer (QA) pairs based on the causal graph and original reports. The questions are categorized into three types to assess model performance across diverse scenarios. The first category focuses on macro-level standard compliance and commonality analysis (25 questions). These questions center on nationally standard-defined cause classifications to evaluate the model's ability to generalize specific incidents into systemic risk patterns. The second category emphasizes micro-level fact extraction and causal traceability (25 questions). These queries target instance-level details to assess the fine-grained retrieval capability of the model in pinpointing specific environmental or behavioral factors. The third category targets multi-dimensional aggregation and relational reasoning (50 questions). These queries involve cross-correlation analysis across industries, regions, and severity levels to examine the model's capacity for cross-node logical reasoning. The three task categories and corresponding illustrative question examples are summarized in Table 2.

Table 2. Taxonomy of Task Categories and Illustrative Query Examples in the Production Safety Accident Analysis.

Task Category	Illustrative Examples
Micro-level Fact Retrieval and Causal Traceability	How many accidents in the database were directly caused by "electric sparks"?
	What is the total number of accidents involving "valve failure"?
Macro-level Standard Attribution and Pattern Analysis	Identify the most frequent national standard cause type across the knowledge graph and provide its specific count.
	Within the construction industry, what is the most frequent cause identified according to national standards?
Multi-dimensional Aggregation and Relational Reasoning	For the industry with the highest accident frequency in Guangdong Province, identify the third most frequent national standard cause type.
	How many electric shock accidents have occurred in the energy and public utility sector of Jiangsu Province?

To ensure objectivity and domain relevance, the benchmark was constructed through a hybrid workflow that combines graph-based exemplar extraction, LLM-assisted question generation, and expert-grounded answer verification. Specifically, we first sampled representative accident instances, standardized causal entities, and their statistical distributions directly from the HC-Graph. These structured graph elements were then used as prompts to guide an LLM to generate candidate questions covering the three target categories, so that the benchmark would reflect both fine-grained case evidence and macro-level statistical patterns. Rather than directly adopting model-generated answers, we used the generated questions as evaluation items and established the reference answers through expert verification. Concretely, three domain experts in emergency management and production safety independently examined each question by jointly consulting the HC-Graph and the corresponding original accident reports, and then determined the final answer based on factual consistency, causal validity, and compliance with national safety standards. For cases involving disagreement, the final annotation was resolved through expert discussion and consensus. This process ensures that the benchmark answers are grounded in both the structured graph representation and the underlying accident records, thereby providing a reliable and authoritative testbed for evaluating causal reasoning performance. The prompts used in the question generation stage are provided in Appendix D.

4.1.4. Evaluation Metrics

To quantitatively evaluate the efficacy of the proposed CausalAgent in production safety accident analysis and emergency decision-support, we select three key metrics: Query Execution Rate (QER), Faithfulness, and Accuracy. These metrics capture the technical robustness of the system while aligning with the requirements of domain-specific analytical tasks.

Query Execution Rate (QER) quantifies the operational reliability of the system in processing complex logical queries. In practical accident investigation scenarios, the ability to respond to professional queries stably is a prerequisite for automation. This metric is defined as the proportion of queries that successfully retrieve non-empty graph data. A high QER indicates that the model can robustly transform natural language into executable database commands.

$$\text{QER} = \frac{\text{Number of queries returning non-empty results}}{\text{Total number of query requests}} \tag{3}$$

Faithfulness assesses the factual consistency between the analytical conclusions and the original accident reports. In production safety accident analysis, the system must avoid the fabrication of non-existent causal factors when reconstructing causal chains. This metric measures the extent to which the responses are strictly grounded in retrieved evidence. A high faithfulness score ensures that the analytical outputs are hallucination-free and feature complete traceability.

$$\text{Faithfulness} = \frac{\text{Number of generated statements supported by retrieved evidence}}{\text{Total number of statements in the answer}} \tag{4}$$

Accuracy measures the correctness of the final results by comparing outputs with ground-truth answers. A high Accuracy score demonstrates the system's potential to facilitate or automate manual verification processes.

$$\text{Accuracy} = \frac{\text{Number of outputs matching the ground-truth}}{\text{Total number of test queries}} \tag{5}$$

4.2. Method Comparison

To evaluate the causal reasoning performance of the proposed method, we conduct a comparative study with three representative baselines representing three distinct paradigms in graph-based question answering. These include: (1) Standard Vector-RAG, which utilizes dense retrieval reasoning; (2) Reasoning on Graphs (RoG) [24], which represents the Graph Search-based KG-RAG method; and (3) StructGPT [28], which represents the Semantic Parsing based KG-RAG method.

4.2.1. Quantitative Evaluation

Table 3 presents a unified comparison between CausalAgent and several representative baselines under two different backbone LLMs, namely Gemini-3-pro-preview and GPT-5.2. To eliminate randomness and enhance experimental reliability, all results in this section are averaged over three independent repeated trials, reporting both the mean and the standard deviation. The results show that CausalAgent consistently outperforms all competing methods by a substantial margin across both settings, demonstrating not only superior task effectiveness but also strong robustness to changes in the underlying reasoning engine.

Table 3. Quantitative comparison of CausalAgent and baseline methods on the production safety accident analysis task under different backbone LLMs. Notations S. and I. denote the utilization of the standard-level and instance-level knowledge graphs, respectively. Results across both Gemini-3-pro-preview and GPT-5.2 consistently show that CausalAgent achieves the best overall performance, demonstrating that its superiority mainly arises from the proposed framework design rather than the choice of a specific backbone model.

Method	QER	Faithfulness	Accuracy
Backbone LLM: Gemini-3-pro-preview			
Embedding based RAG [13] + S.	100.0 ± 0.0%	–	14.2 ± 1.5%
RoG [24] + S.	64.3 ± 2.8%	73.4 ± 2.1%	18.5 ± 2.4%
StructGPT [28] + S.	93.5 ± 1.1%	94.6 ± 0.8%	42.1 ± 1.8%
StructGPT [28] + I.	89.2 ± 1.4%	95.5 ± 0.7%	38.4 ± 1.9%
CausalAgent (Ours)	**100.0 ± 0.0%**	**92.7 ± 1.1%**	**87.3 ± 1.2%**
Backbone LLM: GPT-5.2			
Embedding based RAG [13] + S.	98.2 ± 0.6%	–	11.6 ± 1.8%
RoG [24] + S.	56.7 ± 3.2%	67.9 ± 2.8%	14.2 ± 2.6%
StructGPT [28] + S.	84.1 ± 1.9%	88.1 ± 1.4%	33.7 ± 2.2%
StructGPT [28] + I.	80.5 ± 2.1%	88.8 ± 1.2%	29.5 ± 2.1%
CausalAgent (Ours)	**96.4 ± 1.3%**	**86.5 ± 1.5%**	**72.8 ± 1.7%**

Under Gemini-3-pro-preview, the proposed framework achieves the best overall performance, reaching a 100.0 ± 0.0% Query Execution Rate (QER), 92.7 ± 1.1% faithfulness, and 87.3 ± 1.2% final accuracy. In comparison, the embedding-based RAG baseline attains a perfect QER but only 14.2 ± 1.5% accuracy, indicating that conventional similarity-driven retrieval can ensure execution stability but fails to support reliable causal reasoning in logic-intensive accident analysis. RoG and StructGPT benefit from the incorporation of structured knowledge and therefore achieve moderate improvements, with StructGPT+S. obtaining the strongest baseline accuracy of 42.1 ± 1.8%. However, their performance remains substantially below that of CausalAgent, suggesting that single-layer knowledge representations are insufficient for capturing the cross-level causal dependencies required in this task. By explicitly integrating macro-level safety standards with micro-level accident instances through a dual-layer causal graph, and by coordinating specialized agents,

CausalAgent produces significantly more accurate results with lower variance, reflecting its superior stability.

To further verify that the observed gains are attributable to the proposed framework design rather than the use of a particular backbone LLM, we conduct an additional controlled evaluation by replacing the default reasoning engine with GPT-5.2. As expected, all methods experience a noticeable performance decline under this alternative backbone; yet, the relative advantage of CausalAgent remains highly consistent. Specifically, CausalAgent still achieves the best overall results, with $96.4 \pm 1.3\%$ QER, $86.5 \pm 1.5\%$ faithfulness, and $72.8 \pm 1.7\%$ accuracy, whereas the strongest baseline reaches only $33.7 \pm 2.2\%$ accuracy. This finding provides strong evidence that the superiority of CausalAgent does not depend on a specific foundation model, but instead arises from the proposed framework-level innovations. In other words, while the absolute performance of all methods is influenced by backbone capability, the substantial and persistent margin achieved by CausalAgent across different LLMs confirms that its effectiveness primarily stems from the architectural advantages of the framework itself.

4.2.2. Qualitative Evaluation

To analyze the cognitive boundaries of various frameworks in processing complex safety queries, we present a challenging cross-regional comparative analysis task. This task focuses on comparing the industrial distribution of accidents and their underlying causal mechanisms between Guangdong and Jiangsu Provinces. This scenario necessitates not only macro-level statistical aggregation but also deep multi-hop reasoning across heterogeneous nodes. As illustrated in Table 4, the traditional vector-based retrieval method suffers from significant performance degradation when addressing such macro-statistical tasks. Its outputs exhibit statistical inaccuracies in industry distributions. For instance, it erroneously estimates the proportion of the transportation industry in Guangdong Province as 35% instead of the actual 84%. Furthermore, it exhibits severe semantic hallucinations in causal analysis by capturing only superficial keywords like "natural disasters" or "technical failures". It overlooks critical human and organizational factors, such as "violations of operating rules" in Guangdong and "management deficiencies" in Jiangsu. This confirms that shallow retrieval based on local similarity in vector spaces is insufficient for complex investigation tasks that require rigorous logical aggregation.

When knowledge graphs are introduced as external support, mainstream reasoning paradigms still exhibit notable limitations. Specifically, the information-retrieval-based RoG method shows fragility when faced with uneven information density. Due to its inability to match strongly associated causal paths for Jiangsu Province, the method suffers from catastrophic retrieval failure. When confronted with complex aggregation requirements, its incapacity to conduct cross-node attribute enumeration results in broken reasoning chains. Similarly, the semantic parsing-based StructGPT struggles to balance macro-statistics and micro-reasoning, regardless of the graph granularity. Specifically, StructGPT+S successfully retrieves statistical counts (e.g., 105 cases) by leveraging the standard schema but lacks specific causal details due to the coarse nature of the nodes. In parallel, StructGPT+I fails to execute valid queries entirely, as the model cannot map high-level analytical intents to fragmented, unorganized instance nodes. These results confirm that a single layer of granularity is insufficient for complex safety analysis, leading to either superficial statistics or retrieval failures.

Table 4. Qualitative evaluation on a Complex Analytical Query. Results indicate that the CausalAgent framework achieves optimal performance in logical aggregation and causal interpretability. Notations S. and I. denote the utilization of the standard-level and instance-level knowledge graphs, respectively. **Bold** text highlights accurate key information, while red text indicates errors or inference failures.

Question: Compare the industrial distribution characteristics and latent causal mechanisms of production safety accidents between Guangdong and Jiangsu Provinces based on the knowledge graph.	
Method	Response
Expert Ground Truth (Expert Answer)	Industry Distribution: Significant difference in sample size (Guangdong 125 cases vs. Jiangsu 16 cases). • Guangdong: **Transportation and logistics account for approximately 84%** (reflecting its export-oriented economic characteristics), other industries account for a small proportion. • Jiangsu: **Construction industry ranks first**, followed by chemicals and manufacturing, with a more dispersed distribution. Causal Mechanism: • Guangdong: The core causes are **human behavior/operational errors** and **insufficient education/training**. Risks are primarily attached to the "Flow". • Jiangsu: The core causes are **defects in systems/management**, followed by **physical equipment defects**. Risks are primarily attached to the "Field".
Embedding based RAG+S. [13]	Industry Distribution: • Guangdong: Maritime transportation (∼35%), construction (∼25%). • Jiangsu: Chemicals and heavy industry (∼40%). Causal Mechanism: • Guangdong: Natural disasters (45%) (e.g., typhoons), regulatory gaps (25%). • Jiangsu: Technical failures (50%).
Rog [24]+S.	Industry Distribution: • Guangdong: Maritime/Port operations (correctly identified dominant nodes, but lacks specific values). • Jiangsu: No nodes or cases related to Jiangsu Province were retrieved from the dataset. (Encountered retrieval failure). Causal Mechanism: • Guangdong: Unable to enumerate specific causes due to missing node attributes in the knowledge graph. • Jiangsu: No data available.
StructGPT [28]+S.	Industry Distribution: • Guangdong: Transportation and logistics (105 cases), construction (9 cases), chemicals (4 cases). (High concentration). • Jiangsu: Construction (6 cases), chemicals (3 cases), others dispersed. Causal Mechanism: • Analysis failed. The system reported that "deep causal mechanism" fields in the retrieved data are empty, preventing comparison.
StructGPT [28]+I.	Industry Distribution: • Guangdong: Transportation and logistics dominate (105 cases), followed by construction. • Jiangsu: Construction ranks first (6 cases), followed by chemicals (3 cases). Causal Mechanism: • Analysis failed. The system response indicated that relevant causal data is not included in the database, resulting in an inability to analyze.
StructGPT [28]+HC-graph.	Industry Distribution: • Guangdong: Transportation and logistics (105 cases), construction (9 cases), chemicals and hazardous goods (4 cases). • Jiangsu: Construction (6 cases), chemicals and hazardous goods (3 cases), equipment manufacturing (2 cases). Causal Mechanism: • Specific causal data unavailable. Inferred based on industrial structure: Guangdong's risk stems from high-density logistics (behavior/operation); Jiangsu's risk is related to manufacturing/construction.
CausalAgent (Ours)	Industry Distribution: **Guangdong 125 cases vs. Jiangsu 16 cases.** • Guangdong: **Transportation and logistics dominate overwhelmingly** (105 cases, accounting for approx. **84%**). • Jiangsu: **Construction industry has the highest proportion** (6 cases), followed by the chemical industry (3 cases). Causal Mechanism: • Guangdong: Primary causes are **violations of operations** and failure to follow regulations (e.g., "negligent lookout"). • Jiangsu: Root causes are deeply embedded in **procedural failures** and systemic management loopholes (e.g., "blind demolition", "illegal subcontracting").

In contrast, CausalAgent produces high-quality analytical reports that align with domain expert standards. In terms of industry distribution, the framework accurately generates and executes graph query operators through multi-agent collaboration, yielding precise statistical results (e.g., identifying 105 cases in Guangdong's transportation industry). Regarding causal mechanisms, the HC-Graph enables the model to delve beyond surface phenomena and pinpoint core risk disparities. It identifies "behavioral errors" in Guangdong due to high-density logistics and "management failures" in Jiangsu related to engineering construction. CausalAgent also provides granular case-specific details, such as "negligence in look-out" and "illegal subcontracting", as factual support. In summary, these results demonstrate that CausalAgent effectively suppresses hallucinations through structured constraints and multi-agent cognitive decoupling. It addresses the dual challenges of statistical accuracy and causal interpretability, providing reliable support for evidence-based safety decision-making.

4.3. Effect of Different Backbone LLMs

To comprehensively evaluate the effect of backbone LLM selection on the overall behavior of CausalAgent, we analyze both its computational overhead and task performance under a controlled experimental setting.

Table 5 summarizes the comparative results across four representative backbone models, Gemini-3-pro-preview [8], GPT-5.2 [12], Kimi-k2-250905 [40], and Qwen3-max [10]. The results show that the choice of backbone LLM exerts a substantial influence on both system efficiency and final analytical quality. Among all candidates, Gemini-3-pro-preview delivers the strongest overall performance, achieving the highest Query Execution Rate (94%) and the best analysis accuracy (87%), while maintaining a moderate per-query cost of 0.058 USD and an average latency of 11.8 s. This suggests that stronger semantic understanding and long-context reasoning capabilities are critical for accurately tracing causal paths over the dual-layer knowledge graph.

Table 5. Comprehensive analysis of operational cost, latency, and performance across different backbone LLMs.

Base LLM	Avg. Tokens	Avg. Cost (USD)	Latency (s)	QER	Accuracy
Gemini-3-pro-preview [8]	2840	0.058	11.8	94%	87%
GPT-5.2 [12]	4120	0.132	15.2	82%	54%
Qwen3-max [10]	2260	0.025	9.1	92%	51%
Kimikimi-k2-250905 [40]	3580	0.035	19.4	59%	25%

By contrast, Qwen3-max achieves a comparably high QER of 92% with the lowest average token consumption, cost, and latency among the evaluated models, indicating strong efficiency in executing the workflow. However, its accuracy remains limited at 51%, implying that successful query completion does not necessarily translate into reliable causal reasoning. A similar phenomenon can be observed for GPT-5.2, which sustains an execution rate of 82% but yields only 54% accuracy, despite incurring the highest token usage and operational cost. These results indicate that general instruction-following and tool-calling competence alone are insufficient for the accident analysis task, which requires deeper multi-hop causal inference over structurally complex knowledge representations.

In comparison, Kimi-k2-250905 exhibits the weakest overall results, with only 59% QER and 25% accuracy, together with the highest average latency. Although its monetary cost remains relatively moderate, its limited reasoning robustness makes it unsuitable for this specialized safety analysis setting. Overall, these findings indicate that the upper bound of CausalAgent is largely determined by the reasoning capacity of the backbone

LLM. While lightweight or lower-cost models may offer advantages in execution efficiency, only advanced models with strong long-sequence comprehension and causal reasoning abilities can fully exploit the structural benefits of the proposed HC-Graph and consistently generate rigorous, reliable accident causation analyses.

4.4. Ablation Study

To quantify the individual contributions of CausalAgent, we conduct ablation experiments focusing on the HC-Graph and the MACR. In each configuration, one core module is ablated while the remaining components are kept constant to isolate its functional role. This controlled design evaluates the impact of each module on reasoning accuracy and system robustness.

As illustrated in Table 6, when the MACR mechanism is replaced by a monolithic single-agent generator for end-to-end processing, the QER decreases from 100% to 94%, and the accuracy drops from 87% to 84%. This result indicates that while general LLMs possess basic logical transformation capabilities, their stability degrades when processing complex causal reasoning queries in the production safety domain. The MACR mechanism decomposes complex tasks into four sequential sub-tasks: graph parsing, intent analysis, query generation, and result verification. This task division allows each agent to focus on specialized cognitive functions. Consequently, this modular approach reduces potential syntax errors in complex query generation and enhances the logical rigor of the causal chain through multi-step inference, thereby improving final accuracy.

In contrast, removing the HC-Graph (w/o HC-Graph) exerts a more pronounced negative impact on performance. Although the QER remains at 100%, indicating that the generated queries are syntactically valid, the accuracy plummets to only 36%. This phenomenon highlights the severity of the semantic granularity challenge. Without the dual-layer architecture of the HC-Graph to link fine-grained accident instances with standardized national categories, the model encounters data sparsity or fails to perform effective causal induction. The HC-Graph provides the necessary domain context and structural constraints, serving as the cornerstone for accurate multi-hop reasoning.

In summary, the optimal performance of the full CausalAgent framework confirms that the HC-Graph and MACR mechanism are not merely a functional superposition. Instead, they exhibit a significant synergistic effect. The HC-Graph establishes a high-quality structured knowledge space, while the MACR mechanism provides targeted reasoning strategies to navigate this space precisely. Together, they enable the comprehensive analysis of complex accident causation mechanisms.

Table 6. Ablation study on the key components of CausalAgent.

Components	QER	Accuracy
CausalAgent (Ours)	100%	87%
w/o MACR	94%	84%
w/o HC-Graph	100%	36%

4.5. Failure Case Study

Despite the robust performance across multiple metrics, we conduct a systematic examination of failure cases to identify technical limitations and directions for future optimization. Table 7 illustrates two representative failure scenarios, highlighting requirements for refinement in numerical precision and high-order logical aggregation.

In the first case, regarding a specific frequency query for “valve-related accidents”, the system output was 44 incidents, slightly exceeding the ground-truth count of 42. Further investigation reveals that this discrepancy primarily arises from a failure in semantic disambiguation. During entity association retrieval, the keyword matching logic employed by the

Query Generation Agent was overly broad. Consequently, the agent failed to discern the contextual distinction between instances where a valve failure was the core accident cause and those where a valve was merely mentioned as environmental background. This erroneous inclusion of non-causal noise nodes highlights a persistent challenge in mitigating false-positive interference when modeling causal strength within complex industrial scenarios.

Table 7. Failure cases of CausalAgent. Incorrect answers are highlighted in red color.

Question	Ground Truth	Answer
How many accidents involve valve operation errors or failures?	42	44 accidents.
What is the second most frequent national standard cause?	This issue can be attributed to management factors, specifically the inadequacy of the education and training system.	Cannot determine due to insufficient data; only one cause category ("inadequate education and training management") with 1 case was found.

The second category of failure involves complex logical queries, particularly those involving ordinal constraints, such as "identify the second most frequent cause type". In these scenarios, the system exhibited logical discontinuities. It successfully identified a single high-frequency cause category, such as "inadequate management", but failed to perform accurate aggregation and ranking across multiple node categories. This failure stems from the inconsistent capacity of the multi-agent pipeline to handle long-distance logical dependencies when translating global natural language intents into structured query operators. Syntactic errors or information loss during the query generation process directly resulted in an incomplete reasoning chain.

These failure cases delineate the performance bottlenecks of CausalAgent in extreme reasoning scenarios. Experimental data suggests that substantial scope remains for improvement in retrieval precision, query generation robustness, and output stability. Future research will focus on optimizing the collaborative workflow by integrating dynamic semantic boundary verification and complex logical enhancement operators. These enhancements aim to achieve deterministic knowledge discovery and reasoning outputs in complex deep analysis tasks involving large-scale unstructured data.

5. Conclusions

This paper proposes CausalAgent, a hierarchical graph-enhanced multi-agent reasoning framework designed for the automated analysis of production safety accident reports. By integrating Large Language Models (LLMs) with a Hierarchical Causal Graph (HC-Graph), the framework effectively bridges the gap between unstructured textual evidence and structured, actionable safety intelligence. The proposed framework offers two key technical contributions. First, HC-Graph resolves the semantic granularity mismatch that commonly arises in accident analysis by aligning fine-grained instance-level evidence with standardized national safety taxonomies, thereby supporting both detailed causal tracing and macro-level statistical analysis. Second, the Multi-Agent Collaborative Reasoning (MACR) mechanism decomposes complex causal analysis into modular and coordinated sub-tasks, which improves reasoning robustness, interpretability, and controllability in safety-critical scenarios.

6. Limitations and Future Work

Despite its strong performance, CausalAgent still has several limitations that motivate future work. First, the current framework mainly relies on textual accident reports and does not yet incorporate multimodal evidence, such as images, diagrams, or sensor logs, which may provide additional causal cues. Extending the framework to multimodal causal reasoning is therefore an important direction for improving evidential completeness and

robustness. Second, although HC-Graph provides an effective structured representation, its construction and updating remain relatively static, limiting its adaptability to evolving safety standards and emerging accident patterns. Future work will explore more adaptive graph evolution mechanisms, including incremental knowledge updating and dynamic standard alignment. Third, the multi-agent reasoning pipeline introduces additional computational and storage overhead, which may constrain deployment in resource-limited environments. To address this issue, we plan to investigate lightweight graph storage, more efficient retrieval, and optimized agent orchestration strategies. Finally, practical deployment in production safety settings also raises concerns regarding data security, privacy protection, and access control, since accident reports often contain sensitive enterprise and personnel information. An important future direction is therefore to incorporate security-aware and privacy-preserving mechanisms to support trustworthy deployment in real-world industrial scenarios.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/a19050355/s1.

Author Contributions: Conceptualization, T.W., Z.Z. and S.H.; Methodology, T.W., Z.Z. and S.H.; Software, T.W.; Validation, T.W.; Formal analysis, T.W.; Investigation, T.W. and T.S.; Resources, T.S.; Writing—original draft, T.W. and T.S.; Writing—review—editing, T.S. and Z.Z.; Visualization, T.W. and T.S.; Supervision, Z.Z., S.H., H.H., Q.C. and H.Y.; Project administration, H.H., Q.C. and H.Y.; Funding acquisition, H.H., Q.C. and H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Guangdong Emergency Management Science and Technology Program (No. 2025YJKJ001).

Data Availability Statement: The original contributions presented in this study are included in the Supplementary Material. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Implementation Details of Agents

This appendix provides the implementation details of the core components in the Safety Graph Agent System, including the algorithmic logic for the Graph Parsing Agent and the prompt designs for the LLM-based agents.

Appendix A.1. Graph Parsing Agent

The Graph Parsing Agent extracts the schema from the NetworkX graph object without invoking LLMs. Algorithm A1 illustrates the extraction process.

Algorithm A1 Graph Schema Extraction Algorithm

Input: Directed Graph $G = (V, E)$ with attributes
Output: Schema Description String S

$S_{nodes} \leftarrow \emptyset$ ▷ Dictionary to store node types and attributes
$S_{edges} \leftarrow \emptyset$ ▷ Set to store edge relationships
Phase 1: Node Analysis
for each node $v \in V$ **do**
    $type \leftarrow$ GetType(v)
    if $type \notin S_{nodes}$ **then**
        $attrs \leftarrow$ GetAttributeKeys(v)
        $example \leftarrow$ SampleValue(v)
        $S_{nodes}[type] \leftarrow \{attrs, example\}$
    end if
end for

Algorithm A1 *Cont.*

Phase 2: Edge Analysis
for each edge $(u, v) \in E$ **do**
    $src\_type \leftarrow$ GetType(u)
    $tgt\_type \leftarrow$ GetType(v)
    $rel\_type \leftarrow$ GetEdgeType((u, v))
    S_{edges}.add("(src_type) -[rel_type]-> (tgt_type)")
end for
$S \leftarrow$ FormatString(S_{nodes}, S_{edges})
return S

Appendix A.2. Problem Analysis Agent

This agent analyzes the feasibility of the user's query and decomposes it into sub-tasks.

System Prompt: Question Analysis

You are a planning expert for a Safety Knowledge Graph. Please analyze the user's question based on the provided graph schema.
Graph Schema: {schema}
User Question: "{query}"
Please execute the following steps: 1. **Feasibility Check:** Does the data in the graph suffice to answer this question? (e.g., if specific names are anonymized, questions about names cannot be answered). 2. **Task Decomposition:** If the question is complex, break it down into multiple simple logical steps. 3. **Proxy Task:** If the original question cannot be answered directly, generate the closest possible proxy task.
Output Format (JSON): { "is_feasible": true/false, "reasoning": "...", "sub_tasks": ["Step 1: Find...", "Step 2: Calculate..."], "refined_query": "..." // Fill here if the query needs refinement, otherwise keep original }

Appendix A.3. Query Generation Agent

This agent translates the natural language plan into executable NetworkX (Python) code.

System Prompt: Query Generation

You are a NetworkX expert. Please write Python code to query the in-memory graph object `G`.
Context: 1. Variable `G` is a `networkx.DiGraph` object with data loaded. 2. Graph Schema: {schema}
Task: {task}
Code Requirements: 1. Must define a function `solve(G)`. 2. `solve(G)` must return the result (List, Dict, String, or Number). 3. Use `G.nodes(data=True)` or `G.successors(n)` for traversal. 4. **Do NOT** include any print statements; only return data. 5. **Do NOT** include markdown markers (e.g., "'python); output pure code only.
Logic Hints: - Time range search: Iterate 'AccidentCase' nodes and compare 'occurred_time'. - Cause analysis: Find 'StdCategory' nodes first, then traverse backwards (predecessors) to find specific 'CauseInstance'. - Statistics: Use Python's `len()` or `sum()`.
Code:

Appendix A.4. Reasoning Insight Agent

This agent synthesizes the raw execution results into a natural language response.

System Prompt: Answer Synthesis

You are a professional safety accident analyst. Please answer the user's question based on the query results.
User Question: "{query}" **Analysis Trace:** {trace} **System Raw Result:** {result}
Please generate the final response: 1. **Direct Answer:** Answer the question directly at the beginning. 2. **Data Support:** Cite the queried data as evidence. 3. **Explanation:** Summarize if the data volume is large; explain potential reasons if data is empty. 4. **Style:** Professional, objective, and fluent.
Final Answer:

Appendix B. HC-Graph Construction Algorithm

The construction process involves transforming unstructured accident reports into a structured knowledge graph compliant with the GB/T 13861-2022 standard. Algorithm A2 outlines the detailed procedure.

Algorithm A2 HC-Graph Construction

Input: Set of accident reports $D = \{d_1, d_2, ..., d_n\}$
Input: Standard Taxonomy T_{GB} (GB/T 13861-2022)
Output: Knowledge Graph $G = (V, E)$

Initialize empty graph G
Initialize lookup tables M_{loc} (Location Normalization), M_{cat} (Category Matching)
for each report $d_i \in D$ **do**
    Phase 1: Information Extraction (via LLM)
    $J_i \leftarrow$ LLM_EXTRACT(d_i, T_{GB}) ▷ Extract JSON containing metadata, causes, and relations
    if Extraction Fails **then continue**
    end if
    Phase 2: Entity Normalization
    $loc_{raw} \leftarrow J_i$.get("location")
    $loc_{std} \leftarrow$ NORMALIZELOCATION(loc_{raw}, M_{loc})
    $causes \leftarrow J_i$.get("causes")
    Phase 3: Graph Update
    $v_{case} \leftarrow$ CreateNode(d_i, type="AccidentCase")
    AddNodeToGraph(G, v_{case})
    ▷ Process Metadata Nodes
    for attr in {Industry, AccidentType, Consequence, Location} **do**
        $v_{attr} \leftarrow$ GetValue(J_i, attr)
        if v_{attr} is valid **then**
            AddNodeToGraph(G, v_{attr})
            AddEdge(G, $v_{case} \rightarrow v_{attr}$)
        end if
    end for
    ▷ Process Causal Entities
    for cause $c \in causes$ **do**
        $v_{inst} \leftarrow c$.instance
        $cat_{raw} \leftarrow c$.std_category
        $cat_{std} \leftarrow$ MATCHCATEGORY(cat_{raw}, T_{GB}) ▷ Fuzzy match & semantic alignment
        if cat_{std} is Null **then** $cat_{std} \leftarrow$ "Unclassified"
        end if
        AddNodeToGraph(G, v_{inst}, type="CauseInstance")
        AddNodeToGraph(G, cat_{std}, type="StdCategory")
        AddEdge(G, $v_{case} \rightarrow v_{inst}$, type="has_cause")
        AddEdge(G, $v_{inst} \rightarrow cat_{std}$, type="classified_as")
    end for

Algorithm A2 *Cont.*

▷ Process Causal Relations
for rel $r \in J_i$.relations **do**
AddEdge(G, $r.source \rightarrow r.target$, type="triggers")
end for
end for
return G

Appendix C. Reasoning Complexity Analysis

To characterize the computational boundary of CausalAgent, we analyze the theoretical upper bound of its reasoning complexity from the perspective of graph-based query execution. The overall reasoning process can be decomposed into two major stages: *path hopping*, which retrieves causally relevant subgraphs through multi-hop traversal, and *data aggregation*, which consolidates the retrieved evidence into the final analytical result.

Appendix C.1. Path Hopping Complexity

In HC-Graph, a typical causal query requires traversing multiple semantic layers. A representative reasoning path can be abstracted as *Province* $\rightarrow$ *Location* $\rightarrow$ *Accident Instance* $\rightarrow$ *Specific Cause* $\rightarrow$ *Standard Category*. Let D_i denote the average branching factor from nodes in layer i to nodes in the subsequent layer. For a query involving k hops, the worst-case upper bound of traversal complexity can be expressed as

$$T_{hop} = O\left(\prod_{i=1}^{k} D_i\right). \tag{A1}$$

In the proposed schema, several cross-layer mappings are functional, meaning that each source node corresponds to a unique target node. In particular, the mappings from an accident instance to its location, and from a specific cause to its standardized category, both have branching factor 1. As a result, the effective traversal complexity reduces to

$$T_{hop} = O(D_1 \cdot D_3), \tag{A2}$$

where D_1 reflects the accident density under a regional node and D_3 captures the average expansion from accident instances to specific causal factors. In practice, this complexity is further reduced by the *Problem Analysis Agent*, which anchors the query to a localized entry point and thereby constrains the traversal to a task-relevant subgraph rather than the full graph.

Appendix C.2. Data Aggregation Complexity

After path traversal, the retrieved evidence set contains M candidate result entries. The subsequent aggregation phase consists of three standard operations: (1) a linear scan for frequency accumulation, with complexity $O(M)$; (2) ranking or sorting of candidate categories, with complexity $O(M \log M)$; and (3) Top-N filtering, whose additional cost is negligible relative to sorting. Therefore, the overall upper bound of the aggregation stage is

$$T_{agg} = O(M \log M). \tag{A3}$$

Notably, due to the hierarchical constraints imposed by HC-Graph, the retrieved set size M is bounded by the relevant local subgraph rather than the total graph size. This property prevents the aggregation stage from becoming a bottleneck even as the global graph continues to grow.

Appendix C.3. Total Reasoning Upper Bound

By combining the traversal and aggregation stages, the total worst-case reasoning complexity of CausalAgent for a complex query can be written as

$$T_{total} = O(D_1 \cdot D_3 + M \log M). \tag{A4}$$

This result indicates that the computational cost of reasoning grows approximately log-linearly with the size of the retrieved evidence and only multiplicatively with a small number of effective branching factors in the localized traversal path. Consequently, the proposed framework remains computationally tractable for deep causal reasoning over large-scale production safety accident corpora.

Appendix D. LLM Prompt Template for Benchmark QA Generation

This appendix provides the instruction template used to guide the LLM in generating candidate question items for the benchmark construction process. The prompt was designed to encourage broad coverage over the HC-Graph while constraining generation to structured accident causation information, including accident types, industries, specific causes, standardized cause categories, locations, and their causal relations. The generated questions were subsequently reviewed and verified by domain experts, as described in the main text.

System Prompt:

You are a senior accident analysis expert in the field of production safety. Your task is to generate high-quality question items based on a structured accident causation dataset, which includes entities such as accident types, industries, specific causes, national-standard cause categories, locations, and their causal relationships. All generated questions must be strictly grounded in the structured causal graph and must satisfy the requirements of the following three categories.

Category 1: Macro-level Compliance and Pattern Analysis

Objective: Evaluate whether a model can abstract from individual accident cases to systemic risk patterns defined by national standards.

Requirements: 1. Focus on statistical distributions, common-cause patterns, and compliance with standardized safety classifications. 2. Questions should be answerable through frequency, proportion, ranking, or other aggregate statistics derived from the causal graph. 3. Standardized terminology should be used throughout.

Example Question Templates: (1) "What is the most common accident type in [Region/Industry]?" (2) "What is the most common national-standard cause among all accidents?" (3) "What percentage of accidents are associated with behavioral or operational errors?" (4) "Among [Accident Type] incidents, which cause category accounts for the largest proportion?"

Category 2: Micro-level Fact Extraction and Causal Traceability

Objective: Evaluate fine-grained retrieval ability over instance-level evidence and the capacity to trace complete causal chains.

Requirements: 1. Focus on instance-specific details, root-cause traceability, and the extraction of concrete environmental or behavioral factors. 2. Questions should require answers grounded in explicit causal chains or unique instance-level entities in the graph.

https://doi.org/10.3390/a19050355

Example Question Templates: (1) "What was the primary direct cause of the [Specific Incident]?" (2) "In this case, which unsafe behavior directly led to the occurrence of [Accident Type]?" (3) "Trace the complete causal chain of the [Typical Incident]." (4) "Which environmental factors contributed to the severity of [Accident]?"

Category 3: Multi-dimensional Aggregation and Relational Reasoning

Objective: Evaluate cross-node reasoning ability through multi-dimensional analysis over industry, region, severity, and causation.

Requirements: 1. Focus on correlation analysis, multi-dimensional comparison, and relational reasoning. 2. Questions should require cross-category comparison, association analysis, or synthesized conclusions derived from multiple graph components.

Example Question Templates: (1) "Which cause is most strongly associated with [Accident Type] incidents?" (2) "How do the primary causes of [Accident Type] differ between [Industry A] and [Industry B]?" (3) "What common contributing factors are observed in high-severity accidents across different regions?" (4) "How do the most common accident types differ between Province A and Province B?"

General Constraints: 1. All generated questions must be fully grounded in the structured causal graph; no fabricated or unverifiable content is allowed. 2. The total number of generated items is 100, distributed as 25 for Category 1, 25 for Category 2, and 50 for Category 3. 3. Questions should collectively cover both instance-level evidence and standard-level abstractions, and should reflect a diverse range of accident scenarios. 4. Output format for each item:

Q: [Question]

Appendix E. Verification of the Automated Pipeline

To ensure the reliability of the automated HC-Graph construction, we conducted a rigorous evaluation of the GPT-5-mini-based extraction and mapping process. This evaluation focused on factual integrity and alignment with national standards.

Appendix E.1. Semantic Fidelity Test

We conducted a semantic-level consistency check on 100 randomly sampled accident reports. To account for potential paraphrasing or synonym replacement by GPT-5-mini, we employed an embedding-based verification method rather than rigid string matching. Specifically, each extracted causal factor and its corresponding source context were converted into high-dimensional vectors using a pre-trained embedding model. A match was confirmed if the cosine similarity between the two vectors exceeded a threshold of 0.9, ensuring that the extracted information was semantically grounded in the original report.

Each report was evaluated on a 100-point scale, starting with a base score of 100, where a 5-point deduction was applied for every extracted factor that failed to meet the semantic similarity threshold. Our evaluation across the 100-report test set yielded an average fidelity score of 96.75. The distribution of results indicates high factual stability: 47 reports achieved a perfect score of 100, 41 reports scored 95 (representing one semantically unsupported factor), and 12 reports scored 90 (representing two semantically unsupported factors). This high level of fidelity confirms that GPT-5-mini accurately captures the factual essence of the source reports without introducing hallucinations, providing a robust foundation for subsequent reasoning.

https://doi.org/10.3390/a19050355

Appendix E.2. Expert Audit on Standard Mapping

To verify the semantic alignment between fine-grained instance causes and the national classification standard GB/T 13861-2022, a manual audit was performed on 100 randomly selected mapping pairs. Domain experts reviewed each pair to ensure that the automated categorization accurately reflected the hierarchical definitions of the national standard. This evaluation confirmed a mapping accuracy of 100%, demonstrating that the GPT-5-mini-based pipeline can reliably interpret complex safety standards and assign appropriate categories to instance-level data without manual intervention. These results collectively validate the factual integrity and reliability of the automated HC-Graph construction process.

References

1. Dhinakaran, D.; Kumar, N.J.; Brundha, A.R.; Subiksha, N.; Karpagam, T. Precision Maintenance with PARM and Augmented Reality for Asset Optimization. In *Navigating the Augmented and Virtual Frontiers in Engineering*; Dhinakaran, D., Priya, S., Pravin, A., Ganesh, D., Eds.; IGI Global: Hershey, PA, USA, 2024; pp. 21–48.
2. Cooke, D.L.; Rohleder, T.R. Learning from incidents: From normal accidents to high reliability. *Syst. Dyn. Rev.* **2006**, *22*, 213–239. [CrossRef]
3. Cao, Y.; Wang, X.; Yang, Z.; Wang, J.; Wang, H.; Liu, Z. Research in marine accidents: A bibliometric analysis, systematic review and future directions. *Ocean. Eng.* **2023**, *284*, 115048. [CrossRef]
4. Underwood, P.; Waterson, P. Systems thinking, the Swiss Cheese Model and accident analysis: A comparative systemic analysis of the Grayrigg train derailment using the ATSB, AcciMap and STAMP models. *Accid. Anal. Prev.* **2014**, *68*, 75–94. [CrossRef]
5. Cheng, M.Y.; Kusoemo, D.; Gosno, R.A. Text mining-based construction site accident classification using hybrid supervised machine learning. *Autom. Constr.* **2020**, *118*, 103265. [CrossRef]
6. Baker, H.; Hallowell, M.R.; Tixier, A.J.P. Automatically learning construction injury precursors from text. *Autom. Constr.* **2020**, *118*, 103145. [CrossRef]
7. Zhong, B.; Pan, X.; Love, P.E.; Sun, J.; Tao, C. Hazard analysis: A deep learning and text mining framework for accident prevention. *Adv. Eng. Inform.* **2020**, *46*, 101152. [CrossRef]
8. Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv* **2025**, arXiv:2507.06261. [CrossRef]
9. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The llama 3 herd of models. *arXiv* **2024**, arXiv:2407.21783. [CrossRef]
10. Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. Qwen3 technical report. *arXiv* **2025**, arXiv:2505.09388. [CrossRef]
11. Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. Deepseek-v3 technical report. *arXiv* **2024**, arXiv:2412.19437.
12. Singh, A.; Fry, A.; Perelman, A.; Tart, A.; Ganesh, A.; El-Kishky, A.; McLaughlin, A.; Low, A.; Ostrow, A.; Ananthram, A.; et al. Openai gpt-5 system card. *arXiv* **2025**, arXiv:2601.03267.
13. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
14. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In Proceedings of the The Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
15. Jiang, Z.; Xu, F.F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active retrieval augmented generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023.
16. Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3580–3599. [CrossRef]
17. Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R.O.; Larson, J. From local to global: A graph rag approach to query-focused summarization. *arXiv* **2024**, arXiv:2404.16130. [CrossRef]
18. Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; Huang, C. Lightrag: Simple and fast retrieval-augmented generation. *arXiv* **2024**, arXiv:2410.05779.
19. Jimenez Gutierrez, B.; Shu, Y.; Gu, Y.; Yasunaga, M.; Su, Y. Hipporag: Neurobiologically inspired long-term memory for large language models. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 59532–59569.
20. Huang, Y.; Yan, R.; Zhang, Z. Automated knowledge extraction from marine accident reports using large language models: Graph construction and evaluation. *Ocean. Coast. Manag.* **2026**, *272*, 108015. [CrossRef]

21. Zhang, J.; Zhang, W.; Wei, Q.; Zhong, H.; Miao, Y. GRAG-ProSafe QAS: Graph retrieval-augmented generation for process production safety intelligent question and answer system. *Expert Syst. Appl.* **2026**, *298*, 129947. [CrossRef]
22. Dhinakaran, D.; Edwin Raja, S.; Velselvi, R.; Purushotham, N. Intelligent IoT-Driven Advanced Predictive Maintenance System for Industrial Applications. *SN Comput. Sci.* **2025**, *6*, 151. [CrossRef]
23. Lan, Y.; He, G.; Jiang, J.; Jiang, J.; Zhao, W.X.; Wen, J.R. Complex knowledge base question answering: A survey. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 11196–11215. [CrossRef]
24. Luo, L.; Li, Y.F.; Haffari, G.; Pan, S. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In Proceedings of the The Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
25. Zhang, J.; Zhang, X.; Yu, J.; Tang, J.; Tang, J.; Li, C.; Chen, H. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022.
26. Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.M.; Shum, H.Y.; Guo, J. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv* **2023**, arXiv:2307.07697.
27. He, X.; Tian, Y.; Sun, Y.; Chawla, N.; Laurent, T.; LeCun, Y.; Bresson, X.; Hooi, B. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 132876–132907.
28. Jiang, J.; Zhou, K.; Dong, Z.; Ye, K.; Zhao, W.X.; Wen, J.R. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), Singapore, 6–10 December 2023.
29. Luo, H.; Haihong, E.; Tang, Z.; Peng, S.; Guo, Y.; Zhang, W.; Ma, C.; Dong, G.; Song, M.; Lin, W.; et al. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 11–16 August 2024.
30. Mandilara, I.; Androna, C.M.; Fotopoulou, E.; Zafeiropoulos, A.; Papavassiliou, S. Decoding the mystery: How can LLMs turn text into cypher in complex knowledge graphs? *IEEE Access* **2025**, *13*, 80981–81001. [CrossRef]
31. Smeros, P.; Emonet, V.; Wang, R.; Sima, A.C.; de Farias, T.M. SPARQL-LLM: Real-Time SPARQL Query Generation from Natural Language Questions. *arXiv* **2025**, arXiv:2512.14277.
32. *GB/T 13861-2022*; State Administration for Market Regulation; Standardization Administration of China. Classification and Code of Hazardous and Harmful Factors in Production Process. Standards Press of China: Beijing, China; National Standard of the People's Republic of China: Beijing, China, 2022.
33. Sarkar, S.; Vinay, S.; Djeddi, C.; Maiti, J. Text Mining-Based Association Rule Mining for Incident Analysis: A Case Study of a Steel Plant in India. In *Pattern Recognition and Artificial Intelligence*; Springer: Cham, Switzerland, 2021; pp. 257–273.
34. Li, Y.; Song, D.; Tian, Y.; Wang, H.; Zhou, C.; Zhang, S. A Framework of Knowledge Graph-Enhanced Large Language Model Based on Global Planning. *IEEE Trans. Knowl. Data Eng.* **2025**, *38*, 736–748. [CrossRef]
35. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. *arXiv* **2023**, arXiv:2308.00352.
36. Kuai, C.; Li, Z.; Rosen, B.; Paal, S.; Jafari, N.; Briaud, J.L.; Zhang, Y.; Hashash, Y.; Zhou, Y. Knowledge-Grounded Agentic Large Language Models for Multi-Hazard Understanding from Reconnaissance Reports. *arXiv* **2025**, arXiv:2511.14010.
37. *GB/T 4754-2017*; Industrial Classification for National Economic Activities. National Bureau of Statistics of China: Beijing, China, 2017. Available online: https://openstd.samr.gov.cn/bzgk/std/newGbInfo?hcno=A703F0E23DD165A5A1318679F312D158 (accessed on 8 March 2026) (In Chinese)
38. *GB 6441-2025*;State Administration for Market Regulation (SAMR); Standardization Administration of the People's Republic of China (SAC). Classification and Coding of Work Safety Accidents. National Standard of the People's Republic of China: Beijing, China; Standards Press of China: Beijing, China, 2025.
39. Ministry of Emergency Management of the People's Republic of China. Disaster and Accident Information. Available online: https://www.mem.gov.cn/xw/zhsgxx/ (accessed on 8 March 2026).
40. Kimi K2: Open Agentic Intelligence. Moonshot AI Technical Report. Available online: https://moonshotai.github.io/Kimi-K2/ (accessed on 8 March 2026).