

## Article

# Intelligent Checking Method for Construction Schemes via Fusion of Knowledge Graph and Large Language Models

Hao Li <sup>1,2</sup>, Rongzheng Yang <sup>1,2,\*</sup>, Shuangshuang Xu <sup>1,2</sup>, Yao Xiao <sup>1,2</sup> and Hongyu Zhao <sup>3</sup>

<sup>1</sup> CCCC Second Harbour Engineering Co., Ltd., Wuhan 430040, China; lihao21@ccccltd.cn (H.L.); 2023000354@ccccltd.cn (S.X.); 2021010168@ccccltd.cn (Y.X.)

<sup>2</sup> Key Laboratory of Large-Span Bridge Construction Technology, Wuhan 430040, China

<sup>3</sup> School of Civil Engineering, Chongqing University, Chongqing 400045, China; 20211601069@cqu.edu.cn

\* Correspondence: 2023008911@ccccltd.cn; Tel.: +86-1519-168-9110

**Abstract:** In the construction industry, the professional evaluation of construction schemes represents a crucial link in ensuring the safety, quality and economic efficiency of the construction process. However, due to the large number and diversity of construction schemes, traditional expert review methods are limited in terms of timeliness and comprehensiveness. This leads to an increasingly urgent requirement for intelligent check of construction schemes. This paper proposes an intelligent compliance checking method for construction schemes that integrates knowledge graphs and large language model (LLM). Firstly, a method for constructing a multi-dimensional, multi-granular knowledge graph for construction standards is introduced, which serves as the foundation for domain-specific knowledge support to the LLM. Subsequently, a parsing module based on text classification and entity extraction models is proposed to automatically parse construction schemes and construct pathways for querying the knowledge graph of construction standards. Finally, an LLM is leveraged to achieve an intelligent compliance check. The experimental results demonstrate that the proposed method can effectively integrate domain knowledge to guide the LLM in checking construction schemes, with an accuracy rate of up to 72%. Concurrently, the well-designed prompt template and the comprehensiveness of the knowledge graph facilitate the stimulation of the LLM's reasoning ability. This work contributes to exploring the application of LLMs and knowledge graphs in the vertical domain of text compliance checking. Future work will focus on optimizing the integration of LLMs and domain knowledge to further improve the accuracy and practicality of the intelligent checking system.

**Keywords:** natural language processing; construction scheme; intelligent check; large language model; knowledge graph



**Citation:** Li, H.; Yang, R.; Xu, S.; Xiao, Y.; Zhao, H. Intelligent Checking Method for Construction Schemes via Fusion of Knowledge Graph and Large Language Models. *Buildings* **2024**, *14*, 2502. <https://doi.org/10.3390/buildings14082502>

Academic Editor: Agnieszka Leśniak

Received: 13 May 2024

Revised: 25 July 2024

Accepted: 5 August 2024

Published: 13 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A construction scheme is a comprehensive and advanced construction guideline for a specific construction component or engineering phase, aimed at ensuring the safety, quality and feasibility of the construction process. The construction scheme undergoes multiple rounds of review, including initial, first and second reviews, to ensure compliance with current construction standards. The traditional checking method is typically expert-based. It suffers from low timeliness, high cost, and high dependence on the experts' personal experience and knowledge. It is difficult to ensure a comprehensive and error-free review of the construction scheme. Furthermore, it cannot meet the current requirements for rapid, accurate and cost-effective construction scheme reviews in the construction engineering field [1]. Therefore, exploring intelligent checking methods for construction schemes will reduce the complexity of checking process, improve the efficiency and comprehensiveness of checking process and promote the intelligent development of the construction field.

Many researchers have proposed various digital technology-based auxiliary checking methods in the construction field, including rule-based natural language processing methods and machine learning-based processing methods. However, the sophisticated linguistic comprehension demanded by the checking of construction schemes renders these methods somewhat constrained in the application scope. While rule-based natural language processing methods are effective in specific content checking, their utility is constrained by the pre-defined checking rules and lack of flexibility and portability. Machine learning-based processing methods, although avoiding the rule-making process, require a substantial quantity of high-quality data for model training. Moreover, they are constrained in their ability to understand and process text due to their inability to effectively capture semantic and contextual information.

In 2022, the large language model (LLM) made significant achievements in natural language processing. The LLM uses techniques such as pre-training [2], instruction fine-tuning strategies [3] and human feedback reinforcement learning [4] to train billions of parameters at a massive scale in a vast corpus [5]. This results in a model with stronger language comprehension and generation capabilities, extensive knowledge of the world and the ability to perform complex reasoning task. Currently, LLMs are widely used in general tasks such as intelligent question answering, code generation, text summarization and information extraction.

However, the direct application of LLMs in the compliance checking of construction schemes in vertical domains may encounter two major challenges: (1) The construction scheme checking requires a large amount of construction standard knowledge. However, an LLM is trained on general domain knowledge such as web pages, books and forums and is relatively poor at learning domain knowledge. Therefore, the checking results may contain content that is illusory and factually incorrect [6]. (2) Construction standards are frequently updated, and the frequent fine-tuning of the LLM to update the model's knowledge will greatly increase the cost of training and application. Additionally, ensuring stability during training is challenging.

Some scholars have pointed out that incorporating vertical domain knowledge into the LLM reasoning process can lead to more accurate content generation [7,8]. This approach can alleviate the issues of the LLM's illusion and lack of timeliness to some extent, without increasing the computational overhead. Given that a knowledge graph is a graph model for representing knowledge as a network of associations [9], it is evident that it offers significant advantages in terms of semantic relationship expression, data retrieval efficiency and the flexibility of knowledge increment. In light of the aforementioned advantages, this paper proposes an intelligent checking method for construction schemes through fusing a knowledge graph and LLM. The proposed method provides external domain knowledge for an LLM through a knowledge graph, making the LLM behave closer to experienced human experts, thereby enhancing the reasoning capability of the LLM. This not only markedly enhances the efficiency and precision of construction scheme checking but also reduces labor costs and potential risks during the construction phase. While this study is situated within the context of construction scheme checking in the construction field, its technical approach is generalizable and can be extended to similar tasks in other fields.

## 2. Related Work

### 2.1. Intelligent Checking in Engineering

As informatization and intelligence permeate into the field of construction engineering, automated compliance checking methods and systems have gained widespread research attention. In 2009, Eastman et al. [10] conducted research on compliance checking and identified four phases: (1) the interpretation and logical representation of rules; (2) modeling preparation; (3) rule implementation; and (4) rule checking report. Based on this work, Zhou et al. [11] delved into the automation of compliance checking and improved the overall process. The basic process framework for automated compliance checking

comprises three stages: information extraction and interpretation, information matching and compliance reasoning.

In the information extraction stage, the initial step is to extract checking information from construction standards and subsequently transform them into computer processable and executable checking rules. Early information extraction methods were mainly rule-based methods [12,13]. These methods rely on the manual creation of matching rules to extract entities and relationships from text. These rules are defined based on the syntactic and semantic features of the text [14]. Although these methods are effective for extracting specific standards content, they have poor portability and are difficult to maintain. With the development of natural language processing technology, machine learning-based methods have gradually emerged, such as support vector machines (SVMs) [15], hidden Markov models (HMMs) [16] and K-nearest neighbors (KNNs) [17]. However, the feature engineering of these methods requires a lot of manual participation, and the effectiveness of information extraction is closely related to the quality of the manually constructed features. In recent years, deep learning-based methods have become the mainstream method for information extraction, where large-scale datasets are used for training to automatically learn richer and deeper features. For example, deep learning models such as LSTM-CRF [18], Bi-LSTM CRF [19] and LSTM-NN [20] have been applied to the automatic extraction of in-depth information from construction standards and have achieved good results. Zhang [18] proposed the BILSTM-CRF model to extract semantic and syntactic information elements from regulatory documents in the construction field. Wang [21] developed a relation extraction model based on Attention-CNN to extract the content of provisions associated with specific entities from building safety regulations. Although deep learning-based methods have greatly improved the automation of information extraction from construction standards, the construction of sufficient and high-quality databases is essential for obtaining reliable deep learning models.

In the interpretation stage, Zhang et al. [22] proposed a method for representing standards based on first-order predicate logic. The method generates checking rules by semantically tagging and combining main logical clauses, activation condition logical clauses and compliance checking result logical clauses. With the development of semantic webs, ontology has attracted scholars' attention due to its advantages in knowledge sharing and logic-based representation. Ontology has been adopted in the standard representation. Rasin et al. [23] pointed out that ontology models can clearly express domain knowledge by specifying domain concepts and their relationships. Zhong et al. [24] utilized ontology to construct a knowledge representation model in the engineering domain. The model includes engineering frequency constraints, attribute constraints and selection constraints. Addressing the difficulty of existing models in representing implicit information, nested logic and complex relationships contained in complex clauses in standard texts, Yang et al. [25] proposed a standard representation model based on knowledge graphs. This model presents four knowledge patterns for interpreting, reconstructing, organizing and implementing semantic information across three dimensions: clause-level, sentence-level and text-level.

The objective of the information matching stage is to align the checking rules of the standards with the construction scheme text to be checked. Early matching approaches require manually developing mapping rules to convert the text to be checked and the checking rules into the same representation, with the objective of identifying the association between the two. Pauwels et al. [26] developed a set of mapping rules to map the standard terms to BIM terms for content alignment. Dimyadi et al. [27] developed coding rules to translate BIM information and standards into a specific regulatory knowledge query language for content alignment and reasoning. Nevertheless, these methods are inherently labor-intensive and difficult to maintain. Furthermore, the methods are unable to align pairs with disparate expressions but same meanings, as they lack the capacity to comprehend the semantic information of the text. Semantic vector technology offers a solution to this problem. Given that similar content should be positioned closer together in vector space, similarity calculations can be used to quantify the degree of similarity between texts [28].

Zheng et al. [29] employed unsupervised learning to obtain vector representations of entity words in a public dataset for content alignment. Zhou [30] proposed a method for matching semantic information based on deep learning and a network graph model to align BIM information with regulatory concepts, thereby supporting fully automated energy compliance checks.

Compliance reasoning is the process of applying logical reasoning to content-aligned text in order to ascertain its compliance with established checking rules. Currently, compliance reasoning is mainly implemented through semantic reasoning engines. An appropriate semantic reasoning engine is selected on the basis of the specific logical rule languages, including the Jess reasoning engine [31], the Jena reasoning engine [32] and the B-prolog reasoning engine [33]. Zhang et al. [22] developed an inference system by combining natural language processing technology and logic programming methods. The system employs the B-Prolog platform to automatically check compliance by comparing the logical rules extracted from the specification with the logical facts extracted from BIM. Xu et al. [34] augmented the capacity of SPARQL functions to represent BIM spatial rules and put forth a spatial logic reasoning mechanism for underground public facilities. However, these compliance reasoning methods are based on proprietary frameworks or hard-coded rule-based representations, lacking a comprehensive framework adaptable to rule modeling across diverse domains. Additionally, they are characterized by elevated maintenance costs and constrained flexibility, and their efficacy in qualitative auditing tasks remains a subject for improvement. Table 1 lists some representative studies on compliance checking.

**Table 1.** Studies on compliance checking.

Phase	Method	Advantages	Disadvantages	Relevant Literature
Information Extraction	Rule-based methods	High extraction accuracy and fast efficiency in specific domains	Dependence on manually constructed matching templates and poor portability	[12,13]
	Machine learning-based methods	Alleviates high human costs, avoids cumbersome rule-making and has a strong generalization ability	Requires significant human involvement in complex feature engineering, closely related to quality	[15–17]
	Deep learning-based methods	Automatically learns underlying semantic features and is more flexible when handling large-scale data	Relies on learning from large amounts of high-quality annotated data	[18–20]
Information Interpretation	Logic representation-based methods	Formal grammatical structure and consistent patterns with natural language	Limitations in interpreting and representing complex code clauses and high cost of logic construction	[22]
	Semantic network-based methods	Provides a structured framework for representing concepts, instances and their relationships and facilitates knowledge sharing	Inability to represent generative rules, requires expert involvement for complex rules or query formulation and difficult to create and maintain	[23,24]
	Knowledge graph-based methods	Stronger semantic relationship expression, efficient data retrieval and flexible knowledge association; supports querying and reasoning	Requires manual effort for designing the schema of knowledge graphs	[25]
Information Matching	Rule-based methods	High and fast application effect in quantitative checking tasks	High human cost, difficult to maintain and unable to understand semantic information	[26,27]
	Semantic vector-based methods	Strong versatility and eliminates human cost	Limited by the effect of mapping text to vector space	[29,30]
Compliance checking	Semantic reasoning engine-based methods	Good execution, accurate and fast	Poor versatility, high maintenance cost and low flexibility	[22,34]

## 2.2. Large Language Model Applications

The development of LLMs, such as ChatGPT, has led to the creation of self-developed models by various companies for different application domains. For instance, Huawei has developed the PanGu large model for industrial intelligence [35], Baidu has developed the ERNIE Bot Word for multimedia creation [36], and Tsinghua University has developed the ChatGLM-6B for deployment on consumer GPUs [37]. While an LLM can deeply understand human prompts and perform well in natural language tasks with its powerful text generation capabilities [38], its lack of learning on large-scale specific expertise limits the application of LLMs in the vertical domain.

Therefore, scholars have proposed two methods for enhancing the performance of LLMs in vertical-domain applications: (1) Domain Data + Fine-tuning; (2) Knowledge Base + LLM. The first method involves applying different domain-specific data to fine-tune pre-trained LLMs, such as P-tuning [39] and P-tuning v2 [40], equipping LLM with corresponding domain-specific expertise. Wang et al. [41] performed the fine-tuning of an LLM based on P-Tuning v2 to enhance its capabilities in specialized domains. However, Xu et al. [42] pointed out that fine-tuning may potentially impair the model's inference abilities, leading to catastrophic forgetting. Additionally, the fine-tuning process requires significant computational resources and lengthy training time.

Different from directly injecting knowledge into LLMs through data fine-tuning, the second method adopts a plug-and-play mode. This method typically uses a specialized knowledge base as supplementary data to inject domain knowledge into the LLM inference. This alleviates issues such as lack of authenticity and accuracy in generating results when facing domain-specific problems. Tan et al. [43] combined prompt engineering with local knowledge bases to improve the performance of LLMs in the field of construction engineering. Zhang et al. [44] proposed a knowledge question-answering system based on knowledge graphs and LLMs. The system utilizes a domain-specific knowledge graph to improve the generation results of the LLMs, ensuring accuracy in addressing domain-specific inquiries. To address the limitations of LLMs in knowledge integration and hallucination generation, Wen et al. [45] utilized the logical knowledge implied in knowledge graphs to prompt the LLM. This approach achieved transparent and reliable collaborative inferring through the combination of implicit and explicit knowledge.

## 2.3. Overview of Related Work

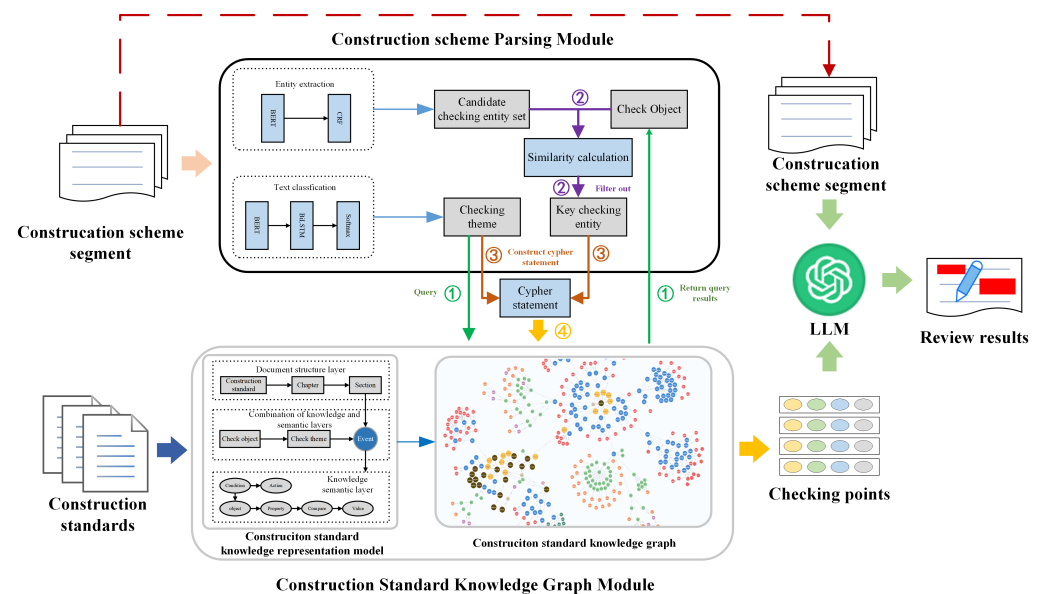
According to the above literature, researchers have reached a general consensus on the steps for achieving compliance checking. However, in terms of research focus, the majority of studies concentrate on the information extraction of construction standards and the interpretation of checking rules. Only a few studies have conducted a comprehensive evaluation of compliance checking tasks. A consensus on the most appropriate methodology for conducting compliance checking has yet to emerge within the construction field. From an automation perspective, research on information extraction and information matching has gradually shifted from human-involved methods to semi-automatic and automated deep learning methods. However, the compliance reasoning methods rely extensively on manually developed rules for maintenance and reasoning. Further studies are required to combine deep learning methods with automated compliance reasoning tasks. With regard to the subject of the checking, the extant research is primarily concerned with BIM models or contract risks, with relatively little attention devoted to the checking of construction schemes. The content of construction schemes is diverse, the semantics are complex, and the distribution of content that aligns with the rules of the standard is relatively scattered. This leads to the need for a reasoning model with robust semantic comprehension capabilities to comprehensively check construction schemes. To address the above problems, this study aimed to propose an automated compliance checking method for construction schemes. This study employs a knowledge graph to construct a standard checking-rule dataset and utilizes a deep learning-based parsing module to automatically align construction schemes with checking rules. In conclusion, the method employs an LLM to achieve intelligent checking.



### 3. Methods

The general framework of the intelligent checking method for construction schemes, integrating a knowledge graph and LLM, is shown in Figure 1. This framework comprises three modules: the construction standard knowledge graph module, the construction scheme parsing module and the LLM inference module.

1. **Construction Standard Knowledge Graph Module:** A hierarchical analysis of the construction standard concepts is conducted from the perspectives of document structure and semantic information. A multi-level representation model for construction standard knowledge, called “Document-Module-Knowledge”, is proposed. This model aims to reveal and organize construction standard knowledge based on multi-dimension and multi-granularity aspects, providing a knowledge foundation for subsequent checking.
2. **Construction Construction Scheme Parsing Module:** The BERT-BiLSTM model is employed for text classification to predict the query type of construction scheme text statements. The BERT model extracts candidate checking entities from construction scheme text statements, and a key checking entity is filtered based on the predicted query type. This process helps to accurately identify crucial standard checking points.
3. **LLM Check Inference Module:** It constructs cipher query statements based on the query type and the key checking entity of the construction scheme text statements. It retrieves construction standard checking points from the standard knowledge graph and combines them with the construction scheme text statements through prompting engineering. The resulting text is then introduced into the LLM for check analysis and result generation.



**Figure 1.** Technology route of intelligent checking method.

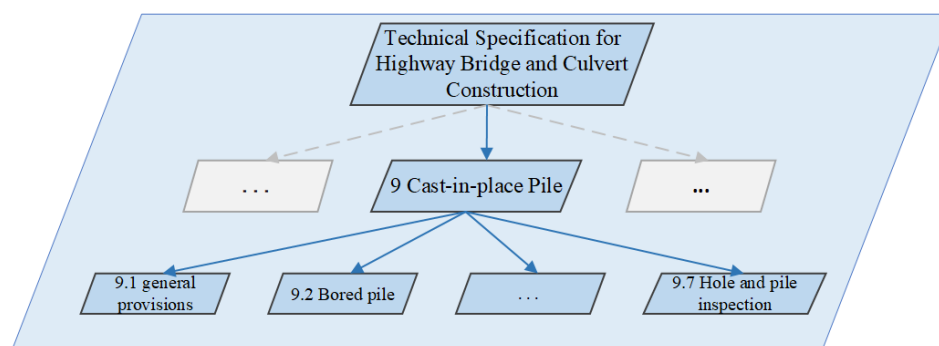
#### 3.1. Construction Standard Knowledge Graph Module

Construction standards are characterized by strong professionalism, rigorous structure and complex sentence patterns. Using a generic annotation semantic approach to construct knowledge graphs is difficult to fully represent the semantic information in standard clauses. Moreover, it overlooks document structure-related information, thus failing to provide precise support for querying standard clauses. To address these limitations, a hierarchical model is adopted for construction standards. This model involves standardizing, constraining and integrating construction standards from both the document structure layer and the knowledge semantic layer. The goal is to achieve the sharing, association and application of modules containing construction standard content.

### 3.1.1. Document Structure Layer

At the document structure layer, construction standards are systematically organized based on the hierarchical structure tree to provide a macroscopic view of the overall knowledge framework. Hierarchical relationships are introduced to effectively promote the association and integration of structural knowledge among construction standards. This aims to meet the demand for comprehensive, rapid retrieval, acquisition and associated searches of construction standards.

Construction standards exhibit clear hierarchical relationships in content structure, typically organized in the form of “chapters”, “sections” and “clauses”. Considering that “clauses” represent the smallest units containing specific knowledge content, only “chapters” and “sections” of construction standards are selected as hierarchical entities to construct the skeleton of the knowledge graph. In taking the “Technical Specifications for Highway Bridge and Culvert Construction” as an example, the knowledge hierarchy structure of its ninth chapter is illustrated in Figure 2.



**Figure 2.** Example of document structure layer (part).

Relationships at the document structure level are mainly hierarchical relationships, consisting of “superordinate” and “subordinate” relationships. In this paper, we define the “superordinate” relationship as the “containment” relationship, describing the containment relationship of higher-level concepts to lower-level concepts.

Furthermore, the document structure layer attributes contain fundamental information about construction standards. They are designed to help users efficiently retrieval and query standards. The primary attributes include the following: Name, Number, Level, Applicable engineering types, Implementation date and Implementation status. Table 2 presents the definitions of these attributes.

**Table 2.** Definitions of document structure layer attributes.

Attribute Type	Definition	Example
Name	Standard formulated and released by the country and industry	Technical Specifications for Construction of Highway Bridges and Culverts [46]
Number	Number of standards	JTG/T 3650-2020
Level	Level of standards	National level, industry level...
Application scope	Engineering type to which the standard applies	Highway and bridge construction
Implementation date	Effective date of the standard	1 October 2020
Status	Whether the standard is currently in force	effective or abolished

### 3.1.2. Knowledge Semantic Layer

The knowledge semantic layer involves the knowledge-based processing of the “clauses” content, aiming to reveal semantic entities and their semantic relationships within the clauses from a semantic perspective. This process leads to the formation of structured check points,

facilitating rapid comprehension by auditors and aiding computer understanding and retrieval. Currently, the clauses of construction standards are classified into three types: attribute constraint type, construction constraint type and reference constraint type.

Attribute constraints are intended to constrain compliance with structure component attributes or construction process indicators. This type of clause contains clear types of entities and relationships, making it easy to extract knowledge and represent structured.

Construction constraints are regulations that govern the construction process to ensure the quality and safety of construction. They consist of alternating conditional and behavioral phrases. It is difficult to represent the construction constraints in a complete and standardized way as knowledge triples, which leads to missing or misinterpreted semantics.

Reference constraints refers to clauses that are not described in detail in this standard, but require reference to an external standard. The key to these clauses lies in the clarity of the subject and the reference object.

Based on the above analysis, the knowledge semantic layer includes a total of 6 types of semantic entities and 4 types of semantic relationships. The definitions of each type of entity and relationship are shown in Table 3 and Table 4, respectively.

**Table 3.** Definitions of knowledge semantic layer entities.

Entity Type	Definition
Object	Objects subject to the clauses, including building components, equipment and materials
Attribute	Attribute requiring compliance check
Value	Attribute value
Comparative word	Comparative relationships between attributes and attributes and attributes and attribute values
Condition	Conditional information of constraint clauses
Action	Compliance behavior adopted under the condition

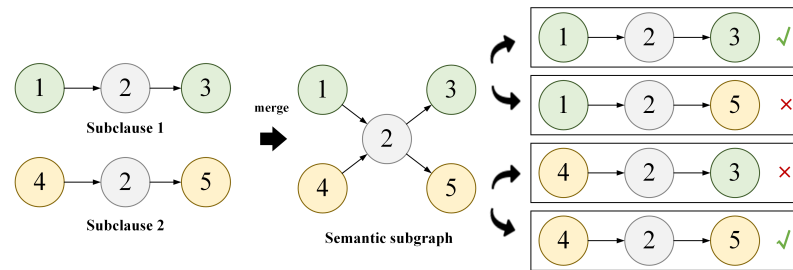
**Table 4.** Definitions of knowledge semantic layer relationships.

Relationship Type	Relationship Expression
has attribute of	[Object]—[has attribute of ]—>[Attribute]
has comparative word	[Attribute]—[has comparative word]—>[Comparative word]
has attribute value of	[Comparative word]—[has attribute value of]—>[Value]
constrains	[Condition]—[constrains]—>[Action]

### 3.1.3. Combination of Knowledge and Semantic Layers

Each standard clause consists of at least one sub-clause. At the knowledge semantic layer, each sub-clause is first transformed into a triad. During the transformation process, the same triples from different sub-clauses are merged to achieve the transformation from standard clause to semantic subgraph. However, as shown in Figure 3, the semantic information represented by a subgraph may be broader than the original semantics of the clause.





**Figure 3.** All possible semantic paths contained in the subgraph of the clause.

To accurately represent the semantic paths in the text, the entity node ‘Event’ is introduced [19]. This node is connected to the end nodes of the semantic paths of each clause via directed edges. This ensures that the details of the semantic paths are accurately captured in the attribute triples of the “Event” node, thereby imposing precise constraints on the semantic subgraph. Figure 4 illustrates the information encapsulated in the attribute triplets of “Event” node. Additionally, the “Event” node’s other end is linked to the corresponding “Section” node, thus facilitating the integration between the document structure layer and the knowledge semantic layer.

Event		
<id>	1717	
id	event96	
level	9.5.6	
nodes	con56,con59,act63,event96	

**Figure 4.** Attributes of Event Nodes.

While the current hierarchical structure of “Document-Module-Knowledge” enhances comprehension and retrieval, it may impede the connection between different sub-clauses or standards. In practical check tasks, it is often necessary to combine information from multiple standards to check construction scheme content and obtain accurate results. To overcome the limitations of the existing layered structure, two types of nodes, “Check Point” and “Check Object” are created on the “Event” node. These nodes directly access associated sub-clauses related to the auditing object and subject, enhancing the check process.

- Check Point: Describes the types of checking themes of the clause, such as “Installation of Steel Casing” or “Drilling Rig Operation”. This node is connected to either “Event” nodes or “Check Object” nodes.
- Check Object: Refers to the subdividing of the check object on the checking theme. For example, the theme of “Drilling Rig Operation” can be divided into “percussion drilling”, “rotary drilling” and so on, according to the selection of drilling rigs. The node is connected to a “checkpoint” at one end and an “event” at the other.

### 3.2. Construction Scheme Parsing Module

#### 3.2.1. Theme Classification of Construction Scheme Text Statement

In the process of classifying the construction scheme, the experts first perform a preliminary analysis of the construction scheme text statement to identify the checking theme. Subsequently, they compare the construction scheme text statement to the relative checking points of construction standards to obtain the checking result. In the overall

process, determining the checking theme of the construction scheme text statement is crucial in establishing appropriate standards.

This study adopted the BERT-BiLSTM model to classify construction scheme text statements. The model mainly consists of three layers: the BERT layer, the BiLSTM encoding layer and the text classification layer.

The BERT layer efficiently learns the semantic features of word-level, sentence-level and inter-sentence relations in the text, and obtains character-level representation vectors.

BiLSTM is used to perform deep contextual semantic mining on output from the BERT to obtain the text feature vectors containing contextual semantic information.

In the text classification layer, the feature outputs from BiLSTM are first mapped to the dimensionality of the category labels using a fully connected layer. Subsequently, a softmax layer is employed to normalize these outputs, enabling the classification of auditing text.

### 3.2.2. Entity Extraction of Construction Scheme Text Statement

The entity identification task focuses on identifying candidate checking objects from the construction scheme text statements, such as equipment, materials, construction processes, etc., directly linked to the relevant checking points in the construction standards knowledge graph. Since no nested entity exists in the construction scheme text statements, entity identification is modeled as a sequence annotation task. BIMS labeling rules are used to mark the construction scheme text statements with tags. The BERT+CRF network model is employed for model training. The trained model can effectively extract the above candidate checking entities.

### 3.2.3. Key Entity Screening

Multiple candidate checking entities may be obtained in the entity identification task. Filtering out the key checking entity relevant to the checking theme helps to enhance retrieval efficiency and ensure the precision of the retrieved standard.

The process of filtering out a key checking entity is as follows. (1) Map the query theme of the construction scheme text statement to a 'Check Point' node on the knowledge graph to obtain a set of associated 'Check Object' nodes. (2) Calculate the similarity between each candidate entity and all nodes in the set, and take the highest score as the association score between each candidate entity and the checking theme. (3) Select the candidate entity with the highest association score as the key checking entity under the checking theme. The semantic similarity calculation formula for entities  $a$  and  $b$  is as follows:

$$\text{sim}(a, b) = \frac{A \cdot B}{|A| \times |B|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

where  $A, B$  represent the word frequency vectors of  $a, b$  respectively;  $n$  represents the number of words obtained by  $a, b$  in the semantic space after word segmentation; and  $\text{sim}(a, b)$  represent the similarity between entities  $a, b$ . The larger the value of  $\text{sim}(a, b)$ , the higher the semantic similarity between them.

## 3.3. LLM Check Inference Module

### 3.3.1. Checking Point Retrieval

Based on the results of text classification and entity recognition, cipher statements are constructed to query the relevant standard checking points from the knowledge graph. For instance, the given construction scheme fragment is 'The density of the slurry should be between 1.03 and 1.10 g/cm<sup>3</sup>, the viscosity should be maintained between 17 and 20 Pa·s, and the colloid ratio should not be less than 98%'. After parsing, the checking theme is 'standard for slurry after hole cleaning' and the check entity object is 'slurry'. The final constructed cipher statement is displayed in Figure 5. The query result of the cipher statement in the knowledge graph is presented in Figure 6.

```

MATCH (n:Check point{name:'standard for slurry after hole cleaning'})$-$[r1:r{relation:'inclusion'}]$-$>(m:Check
Object{name:'slurry'})$-$[r2]$-$>(e:Event)
WITH e MATCH path = (e)<$-$[*]$-$>(target) WHERE ALL(node IN nodes(path) WHERE node.id IN split(e.nodes, ','))
RETURN path

```

Figure 5. Cypher statement.

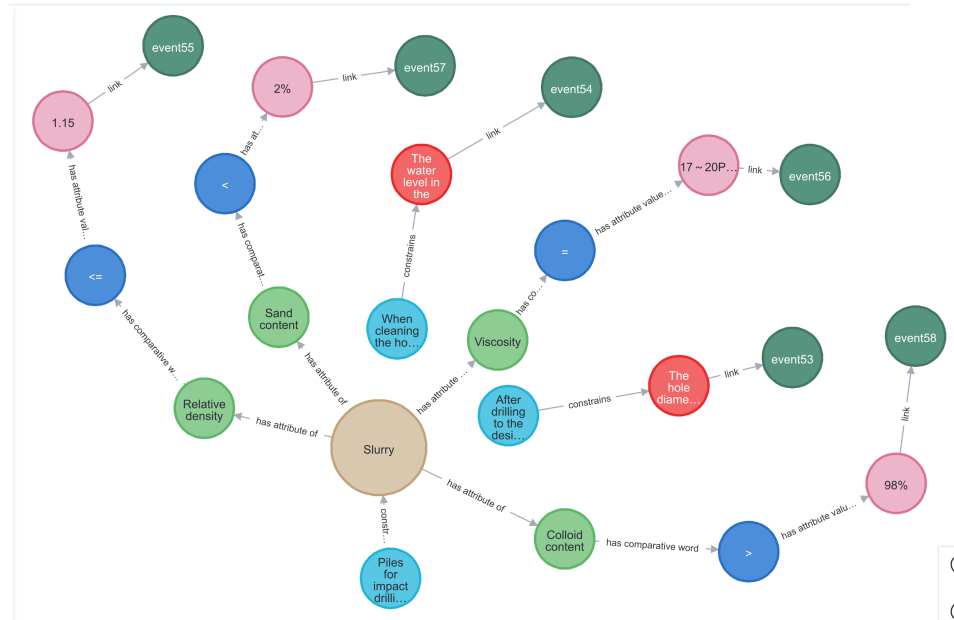


Figure 6. Standard checking points.

### 3.3.2. Prompt Engineering

User-written prompts are the core of prompt engineering for LLMs, guiding the LLM to follow the logic and thought process of the prompt for problem analysis and answer generation. Wei et al. [47] indicate that high-quality prompts should have clear task descriptions, accurate language expression and appropriate example prompts. As shown in Figure 7, we designed a prompt template including the following three components:

<b>Setting role parameters:</b>	Suppose you are an expert in the field of bridges and are able to assess and point out problems with the content of the construction scheme.
<b>Prompting domain knowledge:</b>	Please refer to the following elements of the bridge construction standard: {}
<p><b>Providing task description:</b></p> <p>Given the following content of the construction scheme to be checked [...], make a step-by-step checking judgement:</p> <p>(1) Answer whether the content of the bridge construction standard is related to the content of the construction scheme. If it is not related, then stop the review; if it is related, then go to the next step of the review;</p> <p>(2) Please show the review process step by step:</p> <ol style="list-style-type: none"> <li>① the current review of the construction scheme provisions;</li> <li>② reference to the provisions of the bridge construction standard;</li> <li>③ the review and assessment of the construction scheme from attribute value, unit symbols and flow process, and the review results include: compliance, non-compliance and unable to judge;</li> <li>④ The checking points of non-compliance and unable to judge in the construction plan text are recorded in the cause of the problem.</li> </ol>	

Figure 7. Prompt template.

1. Setting role parameters: Defining the identity parameters of the LLM, which guide and constrain the model's thinking scope and language style.
2. Prompting domain knowledge: This aims to address gaps in the LLM's comprehension of specialized domain knowledge.
3. Providing task description: Providing clear and concise task descriptions to guide the LLM in completing subsequent tasks.

#### 4. Experiments

##### 4.1. Text Parsing Performance

###### 4.1.1. Data Preparation

Data cleaning was performed on 16 special construction schemes of bored piles, including removing illegal characters, duplicate text and filtering, and 544 construction scheme text statements were obtained. These text statements were classified into seven types: steel casing selection, steel casing installation, drilling rig setup, drilling operation, standard for slurry after hole cleaning, mud construction requirements and submerged concrete placement. The types of entities in the text statements include equipment, material and workmanship.

###### 4.1.2. Evaluation Metrics

The precision rate  $P$  the recall rate  $R$  and the harmonic mean  $F1$  value are used to evaluate the classification and extraction performances. The calculation formulas are as follows:

$$P = \frac{TP}{TP + FP} \times 100\%$$

$$R = \frac{TP}{TP + FN} \times 100\%$$

$$F1 = 2 * \frac{P * R}{P + R} \times 100\%$$

where  $TP$  is the number of positive samples predicted to be positive.  $FP$  is the number of negative samples predicted to be positive.  $FN$  is the number of negative samples predicted to be negative.

###### 4.1.3. Scheme Theme Classification Performance

In the text classification task, the proposed model was compared with TextRCNN and BERT on the same dataset. The experimental results are shown in Table 5.

**Table 5.** Experimental results of text classification.

Type	TextRCNN			BERT			BERT-BiLSTM (Ours)		
	R/%	P/%	F1/%	R/%	P/%	F1/%	R/%	P/%	F1/%
Steel casing selection	100	92.59	96.15	100	92.59	96.15	100	100	100
Steel casing installation	100	97.06	98.51	93.94	96.88	95.38	96.00	96.00	96.00
Drilling rig setup	96.55	96.55	96.55	96.55	93.33	94.92	100	94.44	97.14
Drilling operation	94.92	96.55	95.73	96.61	98.28	97.44	93.33	100	96.55
standard for slurry after hole cleaning	97.50	100	98.73	95.00	97.44	96.20	100	100	100
Mud construction requirements	76.19	94.12	94.21	76.19	88.89	82.05	100	76.92	86.96
Submerged concrete placement	100	90	94.74	92.59	83.33	87.72	93.75	100	96.77
F1			95.73			94.02			96.79

In the same bridge construction scheme, the statements' content varies greatly between different modules, while the same module's content in different construction schemes is similar. Therefore, the classification task is relatively simple, and all models achieve great performance. The proposed model outperforms the baseline models on overall performance. This reflects the fact that the BiLSTM model further improves the feature extraction capability based on the semantic features extracted by BERT. This enables the

model to capture the subtle differences between different modules in the bridge construction scheme more accurately, thus improving the text classification performance.

#### 4.1.4. Checking Entity Extraction Performance

In the entity extraction task, mainstream models including BERT and BERT+BiLSTM+CRF were chosen as baseline models for comparative experiments with the model proposed in this paper. The obtained results are presented in Table 6.

**Table 6.** Experimental results of entity extraction.

No	Model	R/%	P/%	F1/%
1	BERT	83.44	86.78	85.06
2	BERT+BiLSTM+CRF	82.39	86.76	84.52
3	BERT+CRF (ours)	84.62	87.42	86.00

Our model outperforms other baseline models in performance metrics, with improvements of 0.94% and 1.48%, respectively, in the constructed dataset. Model 3 introduces CRF on the basis of Model 1, which has an improvement in the extraction performance. This result shows that CRF constructs a global undirected transition probability graph, effectively considering the constraint relationships between preceding and succeeding token labels to improve the extraction performance. However, the addition of the BiLSTM layer to Model 2 resulted in a decrease in extraction performance. This may be due to the fact that the BERT model already has powerful text feature representation capabilities, and the introduction of the BiLSTM layer did not provide any additional informational features. Instead, it increases the complexity of the model, leading to overfitting issues.

### 4.2. Construction Scheme Compliance Checking Performance

#### 4.2.1. Data Preparation

This study constructed a knowledge graph of bridge standards by selecting relevant chapters on bored piles from three bridge standard regulations: ‘Technical Specifications for Construction of Highway Bridges and Culverts’, ‘Quality Inspection and Evaluation Standards for Highway Engineering Section 1 Civil Engineering’[48] and ‘Safety Technical Specifications for Highway Engineering Construction’[49].

In the checking task, a total of 30 text statements were collated from the bored pile construction scheme. These statements constitute positive cases, as they all meet the requirements set out in the standards. Negative cases were composed by manually setting errors in the positive cases, resulting in a total of 20 cases. These text statements were divided into two categories: quantitative descriptions and qualitative descriptions, as shown in Table 7. The checking point of quantitative statements include attribute value constraints and unit symbols. The checking point of qualitative statements include key construction technology and processes.

The accuracy rate is employed as an indicator to evaluate the results of the checking task. Since the checking content of quantitative and qualitative statements is different, the evaluation criteria for the two are also different. For quantitative statements, the checking result is only deemed to be correct if all quantitative attributes have been correctly evaluated. For qualitative statements, the evaluation is considered correct if the construction technology and process are correctly evaluated in accordance with the checking rules, and if the content that is outside the scope of the checking rules is marked as ‘unable to judge’.



**Table 7.** Text statement samples.

Statement Type	Content	Checking Points
Quantitative Statement	<p>泥浆PH值控制在8~10，比重<math>\leq 1.10 \text{ g/cm}^3</math>，黏度宜为17~20 s，胶体率在98%以上，含沙率小于2%。(original)</p> <p>The PH value of the slurry should be controlled at 8–10, the specific gravity should be <math>\leq 1.10 \text{ g/cm}^3</math>, the viscosity should be controlled at 17–20 pa·s, the colloid rate should be higher than 98%, and the sand content should be lower than 2%. (translation)</p>	<ul style="list-style-type: none"> <li>• PH = 8~10;</li> <li>• specific gravity <math>\leq 1.10 \text{ g/cm}^3</math>;</li> <li>• viscosity = 17~20 pa·s;</li> <li>• colloid rate <math>\geq 98\%</math>;</li> <li>• sand content <math>&lt; 2\%</math>.</li> </ul>
Qualitative Statement	<p>若有异常变化，首先提高孔内泥浆水头，加大泥浆比重，提起钻头，防止卡钻、埋钻等现象。(original)</p> <p>If there is an abnormal change, first increase the slurry head in the hole, increase the slurry density, and lift the drill bit to prevent jamming and burying the drill bit. (translation)</p>	<ul style="list-style-type: none"> <li>• increase the slurry head in the hole;</li> <li>• increase the slurry density;</li> <li>• lift the drill bit.</li> </ul>

#### 4.2.2. Comparison of Different LLMs

To compare the performances of different LLMs on the compliance checking task, three user-accessible LLMs were selected, ChatGPT-3.5, ERNIE Bot and ChatGLM-6B. The prompt template from Section 3.3.2 was utilized to ensure better comprehension and the completion of the compliance checking task by each LLM. It should be noted that the objective of this study was to examine the performance of the method based on different LLM bases in the compliance checking task, rather than to identify the optimal LLM for inference.

The performances of the LLMs in the compliance checking of construction schemes are presented in Table 8.

**Table 8.** Checking results of LLMs.

Type	Quantitative Statements (31)	Qualitative Statements (19)	P/%	ALL	AVG
ChatGPT-3.5	64.52% (20)	84.21% (16)	72.00% (36)		75.37%
ChatGLM-6B	58.06% (18)	89.47% (17)	70.00% (35)		73.77%
ERNIE Bot	41.94% (13)	47.37% (9)	44.00% (22)		44.66%

ChatGPT-3.5 demonstrates the best overall and average accuracy, followed by ChatGLM-6B, which performs slightly lower than ChatGPT-3.5 but higher than ERNIE Bot. In the check task of quantitative statements, ChatGPT-3.5 performs the best. In the compliance checking task of qualitative statements, the performances of ChatGLM-6B and ChatGPT-3.5 were found to be comparable. These results reveal significant differences between the different LLMs in the compliance checking task.

Through an in-depth analysis of error samples, the following reasons for these results are summarized.

In the checking task of quantitative statements, the LLMs are required to align with the specified attributes from the text statements of the construction scheme and the checking rules to check the compliance of the numerical range. However, there are different

synonyms for attribute names. Due to the differences in semantic understanding between different LLMs, the model may erroneously treat some synonymous terms, such as ‘specific density’ and ‘relative density’, as different attribute names, which can lead to biases in the checking results. Furthermore, the above LLMs also have limitations in logical calculations in specific domains. For example, the standard regulation may specify that “the diameter-to-thickness ratio of steel casing should not exceed 120”. However, in construction scheme text statements, usually, only the numerical values of the diameter and thickness of the selected steel casing are provided. The LLMs lack the logical knowledge to calculate the diameter-to-thickness ratio, resulting in being unable to complete the check.

In the compliance checking task of qualitative statements, the discrepancies in the performances of the LLMs mainly stem from their differences in information matching and semantic understanding capabilities. In the checking process, the knowledge graph provides all checking points related to the construction scheme text statement, but most of them are redundant. This results in two issues. Firstly, when the LLMs apply the mismatched standard rules to the compliance checking of the construction scheme text statements, the results are biased. Secondly, the LLMs may excessively learn redundant standards knowledge, and treat redundant standard checking rules as critical checking rules for the construction scheme statements. However, since the construction scheme statements do not involve the content of redundant checking rules, the models generate redundant checking results.

#### 4.2.3. Comparison of Different Prompt Templates

In order to explore the impact of prompting engineering on LLMs for the compliance checking task, a new prompt template was designed and called template B, shown in Figure 8. Compared with the prompt template in this paper (named Template A), template B removes the role parameters and does not specify the task execution steps in detail. A total of ten text statements were randomly selected from positive and negative cases to form the experiment cases. The base model for the compliance checking task was ChatGPT-3.5. The number of review points and the number of correct results for the two types of statements under the two prompt templates were calculated, and the experimental results are presented in Table 9.

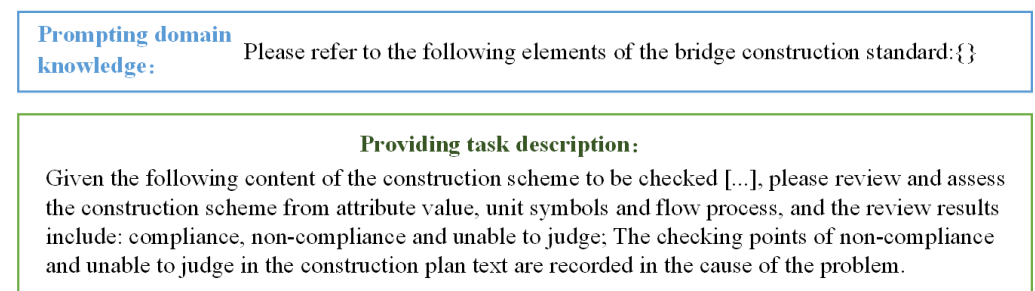


Figure 8. Prompt template B.

Table 9. Check results for different templates.

Type	Quantitative Statements		Qualitative Statements	
	Number of Checking Points (29)	Number of Correct Results	Number of Checking Points (17)	Number of Correct Results
Template A	36	22	23	10
Template B	30	26	19	11

From Table 7, there are 29 checking points in the quantitative statements and 17 checking points in the qualitative statements. However, the numbers of checking points for quantitative and qualitative checking under template A are 30 and 19, respectively, and the numbers of checking points under template B are 36 and 23, which exceed the expected

number of checking points. This indicates that there is excessive checking. However, the number of checking points under template A is significantly lower than that under template B. This reflects that the reasonable division of tasks in the prompt template can more effectively guide LLMs to focus on the execution of tasks. Furthermore, the checking result receives a higher rate of accuracy under template A than under template B. The reason for this is that the step-by-step guidance of template A enables the LLM to initially extract the checking points, then match the corresponding checking rules based on the checking points and finally perform the checking operation, thereby effectively ensuring the correctness of the checking result. As shown in Figure 9, the checking process under template B resulted in an improper match between the checking point and the checking rule, leading to an incorrect checking result.

<p><b>Statement:</b> Steel casing is installed by direct excavation with difficulties supplemented by vibrating hammers for sinking in place. The excavation diameter is 10 cm larger than the outer diameter of the steel casing, and the top surface of the steel casing is 0.1 m above the original ground level, with a maximum center offset of 50 mm, and a vertical tilt of no more than 1/100.</p> <p><b>Checking rules:</b> ① Top surface elevation should be 1.0 to 2.0 m above the water surface . ② Top surface elevation should be 0.3 m above the ground.③ The deviation of the plane position of the center of the casing and the center of the pile should not exceed 50mm. ④ The vertical inclination should not exceed 1/150 ⑤ ...</p>	
<p><b>Template A (ours)</b></p> <p>(1) <b>Relevance Check:</b> ...relevant...</p> <p>(2) <b>Compliance Check:</b></p> <p>① -- <b>Current Construction Plan Text Under Review:</b> The top surface of the casing is 0.1m above the original ground level.</p> <p>-- <b>Relevant Standard Rule:</b> Top surface elevation should be 0.3 m above the ground.</p> <p>-- <b>Audit Judgement:</b> The plan states 0.1 m above ground, which is less than the specified minimum of 1.0 m above the original ground level as per standard. <b>**Not compliant**</b> due to insufficient elevation.</p> <p>② -- <b>Current Construction Plan Text Under Review:</b></p> <p>...</p>	<p><b>Template B</b></p> <p><b>Checking Result:</b></p> <p>-- The top surface of the casing is 0.1m above the original ground level: The standard states that 'the top elevation should meet the requirements of water surface +1.0 to 2.0m or ground surface +0.3m.' It is impossible to determine whether the standard requirements are met.</p> <p>...</p>

**Figure 9.** An excerpt of checking results under different templates.

#### 4.2.4. Robustness Experiment

During the compliance checking process, errors in the text parsing module may lead to the incorrect classification of the text statements, which will result in the LLM model being fed irrelevant standards knowledge. It is necessary to investigate the robustness of our method in the case of errors in the text parsing module.

In the experiment, 10 text statements that are not related to the checking themes were selected to simulate the misclassification of the text parsing module.

The total number of mismatches between the standard rules and the text statements successfully identified by the LLM in the checking task as employed as an evaluation indicator. The results are shown in Table 10.

**Table 10.** Robustness experiment results.

Type		Checking Results	
First stage	irrelevant (8)	relevant (2)	
Second stage	—	unable to judge (2)	check (0)

Of the 10 test samples, the LLM correctly identified 8 as irrelevant and incorrectly identified 2 as relevant. Figure 10 illustrates the checking process of the LLM on the incorrectly judged samples. It can be seen that in the first step of the relevance check, the LLM identified a relevance between the construction scheme text and the standard checking

rules, as both seemed to be relevant to the pile foundation construction process. However, in reality, the standard code rules are about 'standard for slurry after hole cleaning'. In the second step of the compliance checking process of the two misjudged samples, the LLM correctly identified that the standard rules introduced could not be used to check these samples, thereby avoiding the checking results of the illusionary facts. This demonstrates that the two-stage checking process adopted in this study can effectively eliminate the false positive impact of errors in the text parsing module on the checking task. Therefore, our method is robust and ensures the accuracy and reliability of the checking results.

**Statement:** After the core mold is placed and clean water is replaced, 65cm pile concrete pouring will be performed. The pile concrete is underwater C30 concrete, with a concrete volume of 236m<sup>3</sup>, and will be poured to a height of 3.5m from the pile top.

**Checking rules:** ① Slurry sand content should be less than 2%. ② Relative density of slurry should be  $\leq 1.15$ . ③ Slurry viscosity should be 17--20 Pa·s. ④ Slurry colloid rate should be greater than 98%. ⑤ Must maintain borehole water level to prevent cave-ins. ⑥...

**Relevance Check:** The provided construction plan text is **relevant** to the bridge construction standard specifications since it pertains to the overall process of pile construction, which includes steps where slurry and borehole verification are critical.

**Compliance Check:** Since the standard specifications focus on slurry properties and borehole verification, which are not directly referenced in the construction plan text, a direct compliance check on these parameters is not possible. **Unable to Judge.**

**Figure 10.** An excerpt of the check result of the LLM with mismatch between the statement and checking rules.

## 5. Discussion

### 5.1. Reasoning Ability of LLM

In the framework of our method, the LLM is the key reasoning engine for achieving compliance reasoning in construction schemes. As evidenced by the experiment in Section 4.2.2, the performances of different open-source LLMs varied in the compliance checking task. The question of how to select the foundation of the reasoning engine is worthy of further consideration. In the process of scheme compliance checking task, the LLM actually performs an operation that imitates logical reasoning, rather than having the capacity to truly 'understand' and 'think' as humans. Specifically, the mechanism by which the LMM checks construction schemes mainly relies on three key abilities: information retrieval, semantic understanding and computing ability. These abilities enable LLMs to compare the semantic similarity or attribute value range between the scheme statement and the standard checking rules. In addition to the differences in these intrinsic abilities of LLMs, two external factors affect the accuracy of LLM's construction scheme checking tasks: (1) the number of checking rules introduced; (2) the similarity between the checking rules and the scheme statement.

With regard to the influence of the number of rules introduced, the LLM exhibits a tendency to overlearn redundant standard rules, which increases the probability of selecting redundant rules for checking, thus affecting the accuracy of the checking results. For similarity problems, the content of construction schemes may differ from the standard due to the idiosyncrasies of writing habits among individuals, even when they adhere to the same standards. This may manifest as elaboration, omission or rewriting. Consequently, when the similarity between the construction schemes and the standard rules is low, the LLM may hardly perform analytical thinking about the construction scheme based on the standard rules, limiting the effective stimulation of its reasoning ability.

Inspired by ensemble learning, multiple LLMs can be integrated into an integrated base, enabling parallel reasoning and collective decision making for the same checking

task. The results of the reasoning process are aggregated, and the option with the greatest number of votes is returned. The voting rate is then used as an indicator of confidence.

Although LLMs are not yet capable of performing all the same cognitive and reasoning tasks as humans, they are nevertheless possible to simulate the logical reasoning process based on data and appropriate guidance models. The experiment in Section 4.2.3 showed that well-designed prompt templates can guide the model to follow the expected process and stimulate the model's reasoning ability. For the same checking task, the LLM often generates diverse results. By adding example learning examples to the template and having a well-defined format for the output, the LLM can be effectively prompted to transfer its existing thinking ability to a specific professional domain through few sample learning or sample-free learning, prompting the model to generate more reliable and professional responses.

### 5.2. Construct Comprehensive Knowledge Base

The LLM+ external knowledge approach, as implemented in this work, does not require the fine-tuning of the LLM, but the comprehensiveness and detail of the knowledge graph will significantly influence the upper limit of our method. In this study, a two-layer model architecture was employed for the construction of the bridge standard knowledge graph. The two-layer architectural model, comprising the document structure layer and the knowledge semantics layer, was designed to achieve a balance between the macro-module organization of the standard and micro-knowledge analysis. The document structure layer is concerned with the modular structure analysis of the bridge standard, ensuring the efficient retrieval of the content of the standard chapters, while the knowledge semantic layer achieves the precise positioning of the checking rules for specific themes or checking objects through fine-grained knowledge organization. Concurrently, the structured fine-grained knowledge is readily comprehensible and processable by computers. However, due to the considerable number of standards pertaining to construction, a multitude of concepts and terms emerge. Therefore, when defining the knowledge graph framework, the selected concepts should be shared and reused. This ensures that, when new standards are added, only new knowledge entities need to be added to the existing framework, thus enabling the knowledge base to be updated dynamically. In the checking process, the abundance of the knowledge graph also brings the problem of redundant checking rules during the query process. When querying for a specific checking theme or object, the knowledge graph provides a large number of relevant checking rules from multiple sources. However, for the check of specific scheme text, only a few checking rules may be needed. The large amount of unused rules increases the burden of matching information in the LLM. As shown in the experiment in Section 4.2.4, the LLM selects the mismatch rules to check the statement, resulting in incorrect results. In light of the above issues, subsequent research urgently needs to develop efficient rule-ordering algorithms to solve the challenges of rule priority and redundant information. Through optimizing the query and matching method of the knowledge graph, it is expected to improve the accuracy of the LLM checking process and reduce the occurrence of erroneous checking results.

### 5.3. Application in Practice

In practical applications, our method demonstrates robust integration and automation capabilities, facilitating the timely identification of non-compliant content in construction schemes and enhancing the efficiency of construction scheme check. (1) The method offers greater flexibility in the compliance checking of construction schemes. The method primarily incorporates domain knowledge bases, text analysis modules and LLM bases. Each module can be updated or customized according to the specific requirements without affecting the functions of other modules, which makes it highly flexible and integrative. In addition, for different checking tasks, LLM knowledge enhancement can be achieved by defining a knowledge base specific to the domain and fine-tuning appropriately to complete the corresponding checking tasks. (2) The proposed method represents a significant advancement in the automation and reliability of construction scheme compliance



checking. The reviewer only needs to input the construction scheme text, and the proposed method can automatically complete the entire process from text parsing, to the retrieval and matching of checking rules, to compliance checking, greatly saving time and resources. In addition, each checking result is accompanied by comprehensive checking rules and a rationale, which improves the transparency of the checking process and interpretability of the results. When there is doubt about the checking results, the system supports the traceability to the relevant checking rules, ensuring the rigor and credibility of the checking process.

## 6. Conclusions

This paper proposes a method that integrates a knowledge graph and LLM for the intelligent checking of construction schemes, aiming to provide an automated and reliable solution for the compliance checking of construction schemes. The method employs domain knowledge to enhance an LLM, compensating for the lack of professional domain knowledge in the LLM for construction scheme checking tasks, without the need to fine-tune the LLM. The applicability of this method is demonstrated through a case verification of actual construction schemes. Furthermore, a discussion on potential issues in practical application and corresponding insights was conducted. The main conclusions are as follows:

1. This study proposes a method for the compliance check of construction schemes that combines a knowledge graph and LLM. Aiming at the problem of the lack of vertical domain knowledge in LLMs, a hierarchical knowledge graph of construction standards is constructed. Through an text analysis model and the powerful semantic understanding and computational reasoning capabilities of the LLM, the processes of the automatic analysis of construction schemes, automatic query of standard specifications and automatic review reasoning are achieved, thereby achieving the compliance checking of construction schemes.
2. An in-depth examination was conducted to assess the influence of various factors, including the LLM base, prompt templates and parsing module errors, on the effectiveness of the method in the checking task. The experiment results demonstrate that there are notable discrepancies in the performances of different LLMs in the checking task. It can be concluded that semantic understanding and calculation capabilities are the key factors affecting the checking performance. The well-designed prompt template can more effectively guide the model to think logically, activate its potential ability and thus improve the quality of checks. In addition, our method demonstrates robustness through the use of a two-stage checking process, which effectively mitigates the impact of potential errors in the parsing module. It is also accurate in intelligent checking tasks. However, to meet the practical application requirements in real-world scenarios, it is necessary to further optimize the deep integration strategy between the LLM and domain expertise to enhance the model's correctness and interpretability.
3. In practical application, the method can be transferred to other fields by adapting the domain knowledge base to suit the specific requirements of the new field. This illustrates the high adaptability of our method, which provides a valuable avenue for further exploration in the application of LLMs in vertical domains.

There are still some limitations that need to be addressed. Firstly, the dataset used in this experiment is relatively small and does not cover all types of construction schemes. Further research can consider using a larger dataset to evaluate the accuracy and robustness of the method in complex and diverse scenarios. Secondly, the current knowledge graph of construction standards is constructed manually, which is a time-consuming and labor-intensive process. It is necessary for further research to explore methods for automatically constructing high-precision knowledge graphs to improve efficiency and reduce labor costs. Thirdly, the large number of checking rules obtained through knowledge graph retrieval needs to be further screened. A sorting algorithm must be developed to identify the most

critical checking rules from a large number of relative rules, thereby reducing the impact of redundant rules on the LLM.

**Author Contributions:** Study conception and design: H.L., R.Y., S.X. and Y.X.; data collection: R.Y., S.X. and Y.X.; analysis and interpretation of results: H.L., R.Y., S.X. and H.Z.; draft manuscript preparation: R.Y. and S.X. All authors reviewed the results and approved the final version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data used in this study are not publicly available due to confidentiality and privacy concerns. Access to the data can only be granted upon request and with the permission of the appropriate parties. Please contact the corresponding author for further information.

**Acknowledgments:** We thank the editor and anonymous reviewers who provided very helpful comments that improved this paper.

**Conflicts of Interest:** Authors Hao Li, Rongzheng Yang, Shuangshuang Xu and Yao Xiao were employed by the company CCCC Second Harbour Engineering Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Lin, J.R.; Guo, J.F. BIM-based automatic compliance checking. *J. Tsinghua Univ. (Sci. Technol.)* **2020**, *60*, 873–879.
2. Lu, J.W.; Guo, C.; Dai, X.Y.; Miu, Q.H.; Wang, X.X. The ChatGPT After: Opportunities and Challenges of Very Large Scale Pre-trained Models. *Acta Autom. Sin.* **2023**, *49*, 705–717.
3. Wei, J.; Bosma, M.; Zhao, V.Y.; Guu, K.; Yu, A.W. Finetuned Language Models are Zero-Shot Learners. *arXiv* **2021**, arXiv:2109.01652v3.
4. Christiano, P.; Leike, J.; Brown, T.; Martic, M.; Legg, S. Deep reinforcement learning from human preferences. *Adv. Neural Inf. Process. Syst.* **2017**, *49*, 4302–4310.
5. Li, C.T.; Han, X.; Jiang, R.H.; Yun, P.W.; Hu, P.F. Application and prospects of large models in materials science. *Chin. J. Eng.* **2024**, *46*, 290–305.
6. Qin, T.; Du, S.H.; Chang, Y.Y.; Wang, C.X. Key Technologies and Emerging Trends of ChatGPT. *J. Xi'an Jiaotong Univ.* **2024**, *58*, 1–12.
7. Pan, S.R.; Luo, L.H.; Wang, Y.F.; Chen, C.; Wang, J.P. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *arXiv* **2024**, arXiv:2306.08302v3.
8. Xu, Y.X.; Yang, Z.B.; Lin, Y.C.; Hu, J.L.; Dong, S.B. Interpretable Biomedical Reasoning via Deep Fusion of Knowledge Graph and Pre-trained Language Models. *Acta Sci. Nat. Univ. Pekin.* **2024**, *60*, 62–70.
9. Qiao, S.J.; Yang, G.P.; Yu, Y.; Han, N.; Tan, X. QA-KGNet: Language Model-driven Knowledge Graph Question-answering Model. *J. Softw.* **2023**, *34*, 4584–4600.
10. Eastman, C.; Lee, J.M.; Jeong, Y.S.; Lee, J.K. Automatic rule-based checking of building designs. *Autom. Constr.* **2009**, *18*, 1011–1033. [[CrossRef](#)]
11. Zhou, P.; El-Gohary, N. Ontology-based automated information extraction from building energy conservation codes. *Autom. Constr.* **2017**, *74*, 103–117. [[CrossRef](#)]
12. Salama, D.M.; El-Gohary, N.M. Semantic text classification for supporting automated compliance checking in construction. *J. Comput. Civ. Eng.* **2016**, *30*, 4014106. [[CrossRef](#)]
13. Lal, M.S.; Gunaydin, H.M. Computer representation of building codes for automated compliance checking. *Autom. Constr.* **2017**, *82*, 43–58.
14. Zhang, J.S.; El-Gohary, N.M. Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. *J. Comput. Civ. Eng.* **2016**, *30*, 4015014. [[CrossRef](#)]
15. Lee, J.; Yi, J.S. Predicting project's uncertainty risk in the bidding process by integrating unstructured text data and structured numerical data using text mining. *Appl. Sci.* **2017**, *7*, 1141. [[CrossRef](#)]
16. Hassan, F.u.; Le, T.; Lv, X. Addressing legal and contractual matters in construction using natural language processing: A critical review. *Constr. Eng. Manag.* **2021**, *147*, 03121004. [[CrossRef](#)]
17. Hassan, F.u.; Le, T. Automated requirements identification from construction contract documents using natural language processing. *J. Leg. Aff. Disput. Resolut. Eng. Constr.* **2020**, *12*, 04520009. [[CrossRef](#)]
18. Zhang, R.C.; El-Gohary, N. A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Autom. Constr.* **2021**, *132*, 103834. [[CrossRef](#)]
19. Song, J.; Lee, J.K.; Choi, J.; Kim, I. Deep learning-based extraction of predicate-argument structure (PAS) in building design rule sentences. *J. Comput. Des. Eng.* **2020**, *7*, 563–576. [[CrossRef](#)]
20. Pan, Y.; Zhang, L.M. BIM log mining: learning and predicting design commands. *Autom. Constr.* **2020**, *112*, 103107. [[CrossRef](#)]

21. Wang, X.Y.; El-Gohary, N. Deep learning-based relation extraction from construction safety regulations for automated field compliance checking. In Proceedings of Construction Research Congress 2022: Computer Applications, Automation, and Data Analytics, Arlington, VN, USA, 9–12 March 2022.
22. Zhang, J.; El-Gohary, N.M. Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Autom. Constr.* **2016**, *73*, 45–57. [\[CrossRef\]](#)
23. Raskin, V.; Hempelmann, C.; Triezenberg, K.E.; Nirenburg, S. Ontology in information security: A useful theoretical foundation and methodological tool. In Proceedings of the Workshop on New Security Paradigms, Cloudcroft, NM, USA, 10–13 September 2001; pp. 53–59.
24. Zhong, B.T.; Wu, H.T.; Xiang, R.; Guo, J.D. Automatic Information Extraction from Construction Quality Inspection Regulations: A Knowledge Pattern-Based Ontological Method. *J. Constr. Eng. Manag.* **2022**, *148*, 04021207. [\[CrossRef\]](#)
25. Yang, M.S.; Zhao, Q.; Zhu, L.; Meng, H.N.; Chen, K.H. Semi-automatic representation of design code based on knowledge graph for automated compliance checking. *Comput. Ind.* **2023**, *150*, 103945. [\[CrossRef\]](#)
26. Pauwels, P.; Van Deursen, D.; Verstraeten, R.; De Roo, J.; De Meyer, R.; Van de Walle, R.; Van Campenhout, J. A semantic rule checking environment for building performance checking. *Autom. Constr.* **2011**, *20*, 506–518. [\[CrossRef\]](#)
27. Dimyadi, J.; Pauwels, P.; Amor, R. Modelling and accessing regulatory knowledge for computer-assisted compliance audit. *J. Inf. Technol. Constr.* **2016**, *21*, 317–336.
28. Wang, S.K.; Zheng, C.M.; Su, X.; Tang, Y.Q. Construction contract risk identification based on knowledge-augmented language models. *Comput. Ind.* **2024**, *157*, 104082. [\[CrossRef\]](#)
29. Zheng, Z.; Zhou, Y.C.; Lu, X.Z.; Lin, J.R. Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. *Autom. Constr.* **2022**, *142*, 104524. [\[CrossRef\]](#)
30. Zhou, P.; El-Gohary, N. Semantic information alignment of BIMs to computerinterpretable regulations using ontologies and deep learning. *Adv. Eng. Inform.* **2021**, *48*, 101239. [\[CrossRef\]](#)
31. Zhong, B.T.; Ding, L.Y.; Luo, H.B.; Zhou, Y.; Hu, Y.Z.; Hu, H.M. Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Autom. Constr.* **2012**, *28*, 58–70. [\[CrossRef\]](#)
32. Jiang, L.; Shi, J.Y.; Wang, C.Y. Multi-ontology fusion and rule development to facilitate automated code compliance checking using BIM and rule-based reasoning. *Adv. Eng. Inform.* **2022**, *51*, 101449. [\[CrossRef\]](#)
33. Wu, J.; Zhang, J.S. Model validation using invariant signatures and logic-based inference for automated building code compliance checking. *J. Comput. Civ. Eng.* **2022**, *36*, 4022002.1–4022002.14. [\[CrossRef\]](#)
34. Xu, X.; Cai, H. Semantic approach to compliance checking of underground utilities. *Autom. Constr.* **2020**, *109*, 103006. [\[CrossRef\]](#)
35. Zeng, W.; Ren, X.Z.; Su, T.; Wang, H.; Liao, Y. PanGu-a: Large-scale autoregressive pretrained Chinese language models with auto-parallel computation. *arXiv* **2021**, arXiv:2104.12369.
36. Wang, S.H.; Sun, Y.; Xiang, Y.; Wu, Z.H.; Ding, S.Y. ERNIE 3.0 Titan: Exploring larger-scale knowledge enhanced pre-training for language understanding and generation. *arXiv* **2021**, arXiv:2112.12731.
37. Du, Z.X.; Qian, Y.J.; Liu, X.; Ding, M.; Qiu, J.Z. GLM: General language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 320–335.
38. Zhao, B.; Jin, W.Q.; Ser, J.D.; Yang, G. ChatAgri: Exploring potentials of ChatGPT on cross-linguistic agricultural text classification. *Neurocomputing* **2023**, *557*, 126708. [\[CrossRef\]](#)
39. Liu, X.; Zheng, Y.N.; Du, Z.X.; Ding, M.; Qian, Y.J. GPT understands, too. *arXiv* **2021**, arXiv:2103.10385.
40. Liu, X.; Ji, K.X.; Fu, Y.C.; Tian, W.L.; Du, Z.X. P-tuning w2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 61–68.
41. Wang, Y.H.; Bai, H.Y.; Meng, X.Y. An Intelligent Practical Exploration of Large Language Model in Library Reference Consulting Service. *Inf. Stud. Appl.* **2023**, *46*, 96–103.
42. Xu, H.; Zhu, T.Q.; Zhang, L.; Zhou, W.L.; Yu, P.S. Machine Unlearning: A Survey. *ACM Comput. Surv.* **2024**, *56*, 9. [\[CrossRef\]](#)
43. Tan, S.Z.; Zheng, Z.; Lu, X.Z. Exploring and Discussion on the Application of Large Language Models in Construction Engineering. *Ind. Constr.* **2023**, *53*, 162–169.
44. Zhang, H.Y.; Wang, X.; Han, L.F.; Li, Z.; Chen, Z.R. Research on Question Answering System on Joint of Knowledge Graph and Large Language Models. *J. Front. Comput. Sci. Technol.* **2023**, *17*, 2377–2388.
45. Wen, Y.L.; Wang, Z.F.; Sun, J.M. MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models. *arXiv* **2023**, arXiv:2308.09729.
46. JT/T 3650-2020; Technical Specifications for Construction of Highway Bridges and Culverts. China Communications Press: Beijing, China, 2020.
47. Wei, J.; Wang, X.Z.; Schuurmans, D.; Bosma, M.; Ichter, B. Chain-of-Thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.

48. *JTG F80/1-2023*; Quality Inspection and Evaluation Standards for Highway Engineering Section 1 Civil Engineering. China Communications Press: Beijing, China, 2023.
49. *JTG F90-2015*; Safety Technical Specifications for Highway Engineering Construction. China Communications Press: Beijing, China, 2015.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.