applied sciences

MDPI

Article

Knowledge Graph-Based Structural Safety Risk Diagnosis and Control of Prestressed Concrete Bridges

Chunyang Hu [1,2] and Zhe Sun [1,2,*]

[1] College of Architecture and Civil Engineering, Beijing University of Technology, Beijing 100124, China; huchunyang@emails.bjut.edu.cn
[2] Chongqing Research Institute of Beijing University of Technology, Chongqing 401121, China
* Correspondence: zhesun@bjut.edu.cn

Abstract

Reliable structural safety risk diagnosis and control of prestressed concrete bridges is essential for safe operation. Unfortunately, current practice relies heavily on inspectors' engineering knowledge, field experience, and subjective judgments. In addition, existing general-purpose large language models (LLMs) often underperform in bridge defect diagnosis because of missing domain terminology and hallucinated technical statements. Therefore, there is a need to establish a trustworthy and explainable method for structural safety risk diagnosis and control. This study develops a domain knowledge-graph-enhanced framework, the prestressed concrete bridge defect knowledge-graph-enhanced LLM (PCBDK-LLM), to support defect diagnosis and treatment recommendations for prestressed concrete bridges. First, a prestressed concrete bridge defect knowledge graph is constructed using a hybrid approach that combines direct text-driven extraction from standards, peer-reviewed literature, and inspection reports with an ontology-based schema and consistency axioms. Then, the authors propose a retrieval module (REM) that performs ontology-aware chunking and hybrid similarity search to ground a locally deployed dialogue model (DeepSeek-R1) on verified domain knowledge. Eight real rehabilitation cases (eight bridges) are used to evaluate model recommendations against a reference solution documented in the project materials. Results indicate that the proposed PCBDK-LLM generates treatment suggestions that are more consistent with the reference plan than the baseline LLM and the ablation variants.

Keywords: prestressed concrete bridges; defect diagnosis; knowledge graph; ontology; retrieval-augmented generation; local large language model; decision support

Check for updates

Academic Editor: Jose María Alvarez Rodríguez

Received: 1 February 2026
Revised: 2 March 2026
Accepted: 5 March 2026
Published: 6 March 2026

Copyright: © 2026 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.

1. Introduction

Prestressed concrete bridges make up a large share of highway and urban bridge inventories because they provide high load-bearing capacity, long spans, and good serviceability. Many of these bridges are now entering a stage of accelerated deterioration due to increasing traffic demand, aggressive environments, and aging materials. Common defect modes include prestress loss, cracking and spalling, corrosion or rupture of tendons, duct grouting defects, and deterioration of anchorages and bearings. If such defects are not diagnosed in time or if treatments are improperly selected, they can reduce structural reliability and service life, and in extreme cases lead to sudden performance degradation. Consequently, bridge agencies require diagnostic conclusions that are not only timely but

also traceable to evidence and engineering standards, especially for prestressed members where internal conditions are difficult to observe directly.

In practice, defect diagnosis and treatment decisions for prestressed concrete bridges rely on several sources. These include inspection notes, nondestructive testing results, structural health monitoring signals, design drawings, construction records, and maintenance histories. Because these materials are stored across different systems and written in different formats, engineers must translate fragmented observations into a consistent assessment (defect type, location, severity, and required follow-up tests) and then select repair measures that match defect mechanisms, severity levels, and site constraints while complying with specifications. A decision-support tool is therefore valuable if it can connect field observations to codified knowledge and produce recommendations that are both actionable and traceable to evidence.

Large language models (LLMs) have recently shown strong capabilities in natural language understanding, summarization, and interactive question answering, and they are increasingly being adopted in engineering workflows [1–3]. For bridge maintenance, an LLM-based assistant could help engineers locate relevant clauses from standards, interpret inspection narratives, and draft maintenance suggestions in professional language. However, general-purpose LLMs are not designed for high-stakes structural decision-making and can generate plausible but incorrect statements, omit critical constraints, or misuse domain terminology [4–6]. Mitigation strategies such as prompt engineering, retrieval-augmented generation, output clustering, and constraint-based post-checks have been explored [7–9], but their effectiveness depends on whether the context is domain-consistent and whether the reasoning is bounded by engineering rules.

Knowledge graphs provide a complementary foundation for trustworthy decision support by representing domain knowledge as structured entities, relations, and attributes. In the context of prestressed concrete bridges, a knowledge graph can encode components, defect types, severity levels, diagnostic tests, causal mechanisms, and treatment measures, forming a machine-interpretable network that links observations to recommended actions. When combined with an ontology, the knowledge graph can also support consistency checking through axioms and constraints, for example, restricting certain treatments to specific severity levels or requiring that a defect diagnosis be supported by appropriate tests. However, knowledge graphs alone do not provide a flexible natural language interface and cannot easily accommodate engineers' narrative descriptions. Conversely, LLMs offer a powerful interface but require reliable grounding. This motivates a hybrid framework in which an LLM is used for interaction while a knowledge graph- and ontology-based layer provides authoritative, structured context and constraints [10–12].

Retrieval-augmented generation (RAG) grounds an LLM by retrieving relevant knowledge chunks before generation. Yet, a generic RAG pipeline can still retrieve passages that are semantically similar but technically incompatible, especially when different bridge defects share overlapping surface terms. To address this gap, the authors propose the Prestressed Concrete Bridge Defect Knowledge Graph-Enhanced LLM (PCBDK-LLM), which integrates a domain knowledge graph, an ontology-aware retrieval module (REM), and a locally deployed dialogue model (DeepSeek-R1) for privacy-preserving deployment. REM does not claim a new similarity metric; instead, it operationalizes domain constraints by tagging chunks with ontology labels (component, defect type, severity, and method), filtering candidates using these tags, and then performing a two-stage hybrid retrieval (cosine similarity for candidate selection followed by Euclidean-distance re-ranking).

The knowledge graph in PCBDK-LLM is constructed through two complementary strategies. First, a standard-driven text extraction route captures structured knowledge from authoritative specifications (e.g., JTG H11-2021 [13]), peer-reviewed literature, and

inspection/maintenance reports, including defect definitions, diagnostic thresholds, recommended tests, and treatment options. Second, an ontology-based route formalizes top-down classes, relations, properties, and axioms and then populates instances, enabling rule-based consistency checks and constraint-aware retrieval. For reproducibility, the case analysis reports the structured case inputs used for all compared models and explicitly defines the similarity metric and timing protocol.

The main contributions of this paper are as follows: (1) a hybrid knowledge graph construction strategy that combines standard-driven knowledge extraction with ontology-based formalization for prestressed concrete bridge defects; (2) an ontology-aware retrieval module (REM) that uses tag-based filtering and hybrid retrieval to improve domain-consistent grounding and reduce hallucination risk; (3) a privacy-preserving end-to-end deployment using a locally deployed dialogue model to support scenarios involving sensitive infrastructure data; and (4) an end-to-end case study on the eight bridges that evaluates treatment recommendations against a documented reference rehabilitation plan using a clearly defined semantic similarity metric. The remainder of this paper is organized as follows: Section 2 presents the overall framework, Section 3 details the knowledge graph and ontology, Section 4 introduces the knowledge-graph-enhanced LLM design, Section 5 reports the case analysis, and Section 6 concludes the paper.

2. Framework of Prestressed Concrete Bridge Defects Knowledge Large Language Model

PCBDK-LLM integrates a local prestressed concrete bridge defect knowledge graph, the REM retrieval module, and a locally deployed dialogue model. As shown in Figure 1, the knowledge graph provides structured defect classification, diagnosis criteria, and treatment measures. REM retrieves and filters the most relevant knowledge fragments according to ontology tags and similarity scores, and then provides them as grounded context to the dialogue model. Local deployment (DeepSeek-R1) enables instruction-following interaction while protecting project data and proprietary inspection records. The system is designed to (1) classify bridge defects, (2) recommend treatment options consistent with standards, (3) run on a lightweight local setup.

2.1. Overall Architecture and Data Flow

Figure 1 summarizes PCBDK-LLM as an end-to-end decision-support framework that combines a domain knowledge layer (knowledge graph + ontology + standards corpus), an ontology-aware retrieval and grounding layer (REM), and a locally deployed dialogue model for natural-language interaction. The framework is designed around two complementary phases: an offline knowledge preparation phase that organizes authoritative engineering knowledge into a retrievable and constraint-aware form, and an online inference phase that answers engineers' questions by retrieving defect-consistent evidence and generating an actionable diagnosis and treatment suggestion.

Offline knowledge preparation focuses on building (i) a prestressed concrete bridge defect knowledge graph that connects components, defect types, severity levels, diagnostic tests, and treatment measures, (ii) an ontology that defines the semantics of these concepts and provides axioms/constraints for consistency, and (iii) a document corpus containing standards clauses, technical manuals, and project records. This phase also produces tagged knowledge chunks and vector indexes that can be queried efficiently during online inference.

Online inference starts from a user query (or a structured inspection summary) and follows a consistent workflow: (1) interpret the query and map key phrases to ontology labels such as component, defect type, and severity; (2) retrieve candidate evidence from

the knowledge corpus using REM; (3) assemble a grounded context package that includes the most relevant clauses, criteria, and recommended actions; (4) generate a response with the local dialogue model under a controlled prompt template; and (5) apply lightweight rule checks to ensure that the proposed treatment is compatible with defect severity and required tests. This workflow is intentionally designed to provide traceable evidence, reduce hallucination risk, and keep response time within practical engineering requirements.

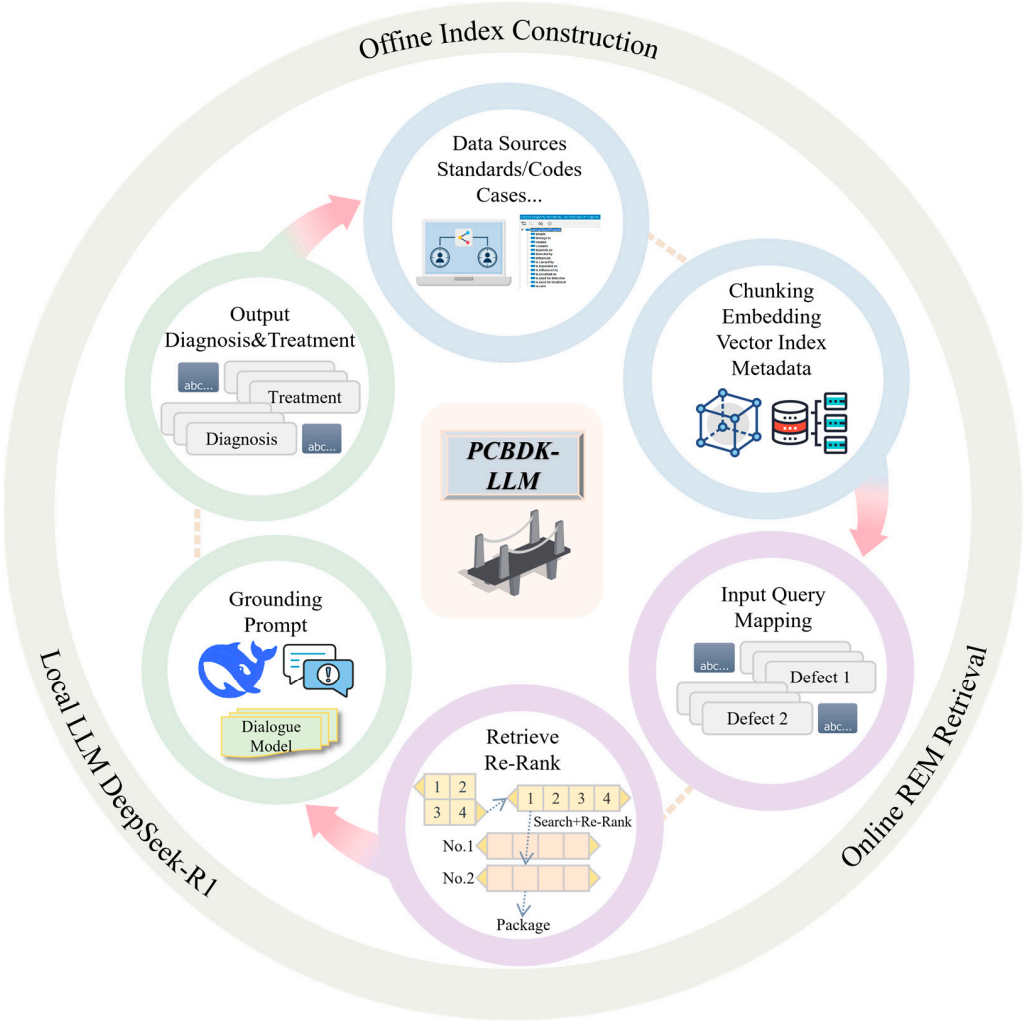


Figure 1. A Framework of PCBDK-LLM.

2.2. Knowledge Layer: Defect Knowledge Graph, Ontology, and Document Corpus

The knowledge layer serves as the authoritative backbone of PCBDK-LLM. It provides three tightly coupled assets: a defect knowledge graph, an ontology specification, and a document corpus. The knowledge graph organizes domain facts and procedural knowledge into machine-interpretable structures, enabling explicit links between what is observed in the field and what is recommended by standards and engineering practice. The ontology defines the vocabulary and constraints that make these links unambiguous and checkable.

In this work, the core entity types include bridge components (e.g., girder, tendon/duct, anchorage, bearing), defect types (e.g., cracking, spalling, corrosion, grout defects, prestress loss), severity levels (e.g., mild, moderate, severe, urgent), diagnostic tests and indicators (e.g., visual inspection, rebound hammer, ultrasonic/impact-echo, corrosion potential, grouting fullness, stress/strain monitoring), and treatment measures (e.g., crack sealing, patch repair, grouting repair, tendon protection or replacement, an-

chorage strengthening). Relations connect these entities through engineering meaningful semantics such as "component-has-defect", "defect-has-severity", "defect-requires-test", and "defect-recommended-treatment".

The document corpus includes standards clauses and technical guidance (for example, maintenance and inspection specifications) as well as project records. Each document is segmented into knowledge chunks, and each chunk is tagged with ontology labels so that retrieval can be restricted to defect-consistent evidence. These tags are essential because different defects can share overlapping surface terms; without tags, a similarity-only retriever may return text that appears relevant linguistically but is incompatible with the inspected component or severity.

2.3. Retrieval and Grounding Layer: Ontology-Aware REM

A standard RAG pipeline retrieves text chunks mainly by semantic similarity. While effective in general domains, this approach can be unsafe in engineering scenarios because semantically similar passages may correspond to different components, different defect mechanisms, or different severity levels. To address this issue, PCBDK-LLM introduces REM as an ontology-aware retrieval module that explicitly enforces defect consistency before generation.

REM is composed of three steps. First, the query (or structured case input) is parsed to extract ontology labels, including component, defect type, severity, and optional constraints such as environmental exposure or traffic limitation. Second, REM filters candidate chunks by matching these labels to the chunk tags, thereby eliminating technically incompatible evidence early. Third, REM performs a two-stage hybrid retrieval: cosine similarity is used to quickly select a top-k candidate set in the embedding space, and Euclidean distance is then used for re-ranking within this filtered set. The study emphasizes that REM does not claim novelty in the mathematical similarity primitives; rather, its contribution is a domain-constrained retrieval workflow that improves the relevance and safety of the grounding context.

The final output of REM is a compact grounding package that contains: (i) diagnosis criteria relevant to the defect and component, (ii) recommended tests or follow-up checks required by standards, and (iii) treatment options and their applicability conditions. This package is passed to the dialogue model together with an instruction template that requires the model to cite evidence items (e.g., clause identifiers or knowledge IDs) and to state assumptions explicitly. By separating retrieval (evidence selection) from generation (language output), the framework improves transparency and allows engineers to verify the basis of recommendations.

2.4. Dialogue Layer and Local Deployment

The dialogue layer provides a natural-language interface for engineers and inspectors. In PCBDK-LLM, the dialogue model is deployed locally to protect sensitive infrastructure information, including inspection records, monitoring data, and proprietary rehabilitation plans. Local deployment also enables stable performance in field environments where cloud access may be restricted.

During interaction, the user can provide either (a) an inspection narrative and supporting measurements, or (b) a structured input template derived from inspection forms. The dialogue layer converts the input into a controlled prompt that includes the REM grounding package, requests an output in a standardized engineering format, and restricts the response to the retrieved evidence. The model is instructed to avoid overconfident claims when evidence is insufficient and to recommend additional tests when required by standards.

2.5. Outputs, Traceability, and Practical Use

The framework is designed to produce outputs that are directly usable in maintenance workflows. The response template requests the model to output: (1) defect classification and severity interpretation; (2) key evidence supporting the diagnosis (grounded in retrieved clauses/knowledge items); (3) recommended diagnostic tests or monitoring actions if needed; (4) treatment options with applicability conditions and implementation notes; and (5) cautions and limitations. This structure ensures that the recommendation is not a single opaque paragraph but a verifiable decision-support artifact.

From a usability perspective, the framework supports two typical interaction modes. In the first mode, an engineer asks targeted questions such as "What treatments are recommended for grout defects in ducts at moderate severity?" and receives a grounded answer with evidence. In the second mode, the user provides a case summary, and the system returns a diagnosis and a treatment plan aligned with standards and domain constraints. In both modes, PCBDK-LLM is intended to assist, not replace, professional judgment; the final decision remains with qualified engineers and must consider site-specific constraints.

Finally, the end-to-end pipeline is designed to meet practical time requirements. REM reduces the context length by filtering candidates before re-ranking, and local inference avoids network latency. Together, these design choices allow PCBDK-LLM to provide traceable recommendations within a time window that is compatible with routine inspection and maintenance planning.

3. Knowledge Graph Construction for Structure Safety Risk Diagnosis and Control of Prestressed Concrete Bridges

3.1. Standard-Driven Knowledge Extraction and Text-Chunks Construction

To support trustworthy diagnosis and treatment recommendations, the knowledge graph is constructed from authoritative engineering sources and then converted into retrievable text chunks for REM. In this study, the authors follow a standard-driven extraction workflow: (i) collect and curate standards clauses, technical manuals, and peer-reviewed literature; (ii) segment documents into semantically complete units; (iii) extract structured fields and normalize terminology; and (iv) store the resulting entities, relations, and attributes with explicit provenance. This workflow emphasizes traceability: each knowledge graph entry can be traced back to a specific standard clause or technical reference so that engineers can verify why a recommendation is produced.

The extracted knowledge is organized into four tightly coupled modules that align with routine bridge management workflows. First, a defect classification system is built to unify naming across inspection records. The study adopts a hierarchical taxonomy (component level-defect category-sub-type-symptom/manifestation) so that defects such as tendon corrosion, duct grouting defects, and anchorage deterioration can be distinguished even when inspection narratives use colloquial descriptions. For each leaf node, the study defines aliases and normalization rules to reduce ambiguity (e.g., 'grout void', 'insufficient grouting', and 'poor grouting compactness' are mapped to a unified concept).

Second, diagnostic standards and methods are encoded by extracting condition criteria, applicable tests, and severity indicators. For each defect type, the knowledge graph stores the recommended diagnosis pathway (visual inspection-special testing-monitoring/early warning) together with threshold values and interpretation notes. This enables the system to answer not only 'what is the likely defect', but also 'what evidence is needed to confirm it' and 'what additional test should be performed when evidence is insufficient'. Table 1 summarizes typical diagnosis methods used as core nodes in the diagnostic subgraph.

Table 1. Typical diagnostic methods for prestressed concrete bridge defects and the corresponding evidence types.

Diagnosis Item	Testing Tools	Operation Points	Code Basis
Crack Width and Length Measurement	Crack Width Gauge and Tape Measure	Measure at the widest part and both ends of the crack, record the crack direction and distribution area	JTG H11-2021
Concrete Spalling and Reinforcement Exposure Inspection	Vernier Caliper and Camera	Measure spalling area, exposed reinforcement length and corrosion degree, take images for record	Article 5.4.2 of JTG H11-2021
Anchor and Steel Strand Visual Inspection	Flashlight and Small Hammer	Inspect anchor deformation and rust, tap the anchor lightly with a small hammer to judge compactness	Article 5.4.5 of JTG H11-2021

Third, treatment schemes are formalized as structured entries with applicability conditions. Each treatment node records the recommended method, required materials/equipment, implementation steps, safety notes, and constraints such as 'only applicable for moderate severity' or 'requires confirmation of grouting fullness before tendon-related intervention'. Treatments are linked to (a) defect mechanisms (cause priority), (b) severity grades (hierarchical treatment), and (c) site constraints (traffic organization, accessibility, and environmental exposure), allowing the assistant to generate recommendations that are actionable rather than generic.

Fourth, an engineering case base is introduced to complement standard clauses with practice-oriented context. Each case records the project background, observed symptoms, test results, final diagnosis, selected treatment, and post-treatment verification. Cases are indexed by component and defect type, and are used by REM as high-value evidence to provide implementation details and common pitfalls. Table 2 shows an example entry for steel strand corrosion, illustrating how a real case is encoded into the same structured template used by the knowledge graph.

Table 2. Example of a structured knowledge entry for steel strand corrosion treatment (case-based).

Defect Severity	Treatment Technology	Construction Points	Quality Control Standard
Mild Corrosion	Duct Grouting Repair	**CP.1**-Clean anchors and duct inlets **CP.2**-Adopt vacuum-assisted grouting technology to inject high-performance cement grout **CP.3**-Cure for $\geq$7 days	Grouting compactness $\geq$ 95%, steel strand corrosion stops developing
Moderate Corrosion	Local Steel Strand Repair and Grouting Reinforcement	**CP.1**-Chisel concrete in the corroded area to expose steel strands **CP.2**-Wrap carbon fiber cloth after derusting **CP.3**-Restore concrete protective layer and grout	Steel strand stress loss rate after repair $\leq$ 10%
Severe Corrosion	Steel Strand Replacement and External Prestressing Reinforcement	**CP.1**-Erect temporary supports **CP.2**-Remove old steel strands, replace with new ones and tension **CP.3**-Add external prestressing strands for auxiliary load-bearing	New steel strand stress meets design value, external strand tension control stress $\pm$ 5%

3.2. Ontology-Based Formalization and Rule Constraints

While standard-driven extraction provides rich content, an ontology is required to formalize semantics and enforce consistency. The study therefore constructs an ontology that defines the core classes (Component, Defect, Severity, DiagnosticTest, Indicator, TreatmentMeasure, and Case) and their object/data properties. The ontology plays two roles in PCBDK-LLM: it (i) provides stable labels for REM to perform ontology-aware filtering, and (ii) enables rule/axiom checks to reduce technically incompatible outputs. Table 3 lists the subdivisions of the core concepts used in the ontology.

Table 3. Core ontology concepts and representative subclasses used for prestressed concrete bridge defect knowledge.

Core Concepts	Including Categories
Bridge Structural Components	Main beams, piers, abutments, bearings, deck pavement, expansion joints, guardrails, foundations, etc.
Defect Types	Cracks, spalling, corrosion, deformation, leakage, looseness, damage, blockage, etc.
Defect Causes	Material aging, overloading, environmental erosion, construction defects, insufficient maintenance, unreasonable design, natural disasters, etc.
Detection Methods	Visual inspection, ultrasonic testing, rebound testing, radar testing, strain gauge testing, infrared thermographic testing, penetration testing, etc.
Disposal Measures	Repair, reinforcement, replacement, cleaning and dredging, anti-corrosion treatment, monitoring and observation, etc.
Defect Levels	Slight, Moderate, Severe, Critical
Detection Equipment	Ultrasonic detector, rebound tester, radar detector, infrared thermal imager, strain gauge, etc.
Influencing Factors	Traffic volume, temperature change, humidity, geological conditions, service life, etc.

In terms of class hierarchy, treatment measures are organized by purpose and mechanism (e.g., sealing/patching, protective coating, grouting repair, tendon protection or replacement, anchorage strengthening, and load restriction/monitoring), and are further refined by applicability conditions.

At the relation level, the study defines key object properties with explicit domain and range constraints, such as hasDefect (Component-Defect), hasSeverity (Defect-Severity), requiresTest (Defect-DiagnosticTest), supportedByIndicator (Defect-Indicator), and recommendedTreatment (Defect-TreatmentMeasure). These relations enable multi-hop reasoning. For example, a query about 'duct grouting defect' can retrieve not only treatments but also the tests required to confirm grouting fullness and the indicators used to grade severity.

At the axiom and rule level, the study encodes consistency constraints that reflect engineering logic. Examples include: (i) certain treatments are disallowed for mild severity (e.g., tendon replacement should not be suggested without severe indicators); (ii) a treatment that depends on internal tendon state requires at least one confirming test result; and (iii) mutually exclusive defect types (e.g., 'intact grout' vs. 'voided grout') cannot co-exist for the same duct segment. These constraints are used in two places: during REM filtering (by restricting candidate chunks to those matching component/defect/severity tags) and during response post-checking (by flagging missing tests or incompatible severity-treatment pairs).

Finally, to connect the ontology to retrieval-augmented generation, each knowledge graph entry is linearized into a fixed text template before embedding and indexing. A typical chunk follows the structure: component-defect-severity-diagnosis criteria-required tests-recommended treatment-implementation notes-reference. This template ensures that retrieved evidence is both semantically consistent (through tags) and information-complete (through fields), thereby improving the stability of downstream generation. The ontology

and knowledge graph are versioned so that new standards clauses, new defect patterns, and newly accumulated cases can be appended without breaking existing labels and constraints.

4. Knowledge-Graph-Enhanced Large Language Model for Defect Diagnosis and Treatment of Prestressed Concrete Bridges

To support high-stakes maintenance decisions for prestressed concrete bridges, PCBDK-LLM combines a domain knowledge graph and ontology with retrieval-augmented generation (RAG) and a locally deployed dialogue model. The core goal is to provide recommendations that are not only fluent, but also grounded in verifiable evidence such as standards clauses, diagnostic criteria, and documented engineering cases. Figure 2 summarizes the end-to-end workflow: (1) interpret the user query or structured inspection input, (2) retrieve defect-consistent evidence through REM, and (3) generate a structured diagnosis-and-treatment response under explicit constraints and traceability requirements. Compared with a generic RAG pipeline, the framework emphasizes ontology-aware grounding and auditability. Evidence is retrieved with component/defect/severity tags, assembled into a compact context package with provenance, and then provided to the local model through a controlled prompt template. When the available evidence is insufficient, the system is instructed to recommend additional tests or monitoring actions rather than to speculate, consistent with common reliability practices in infrastructure assessment.

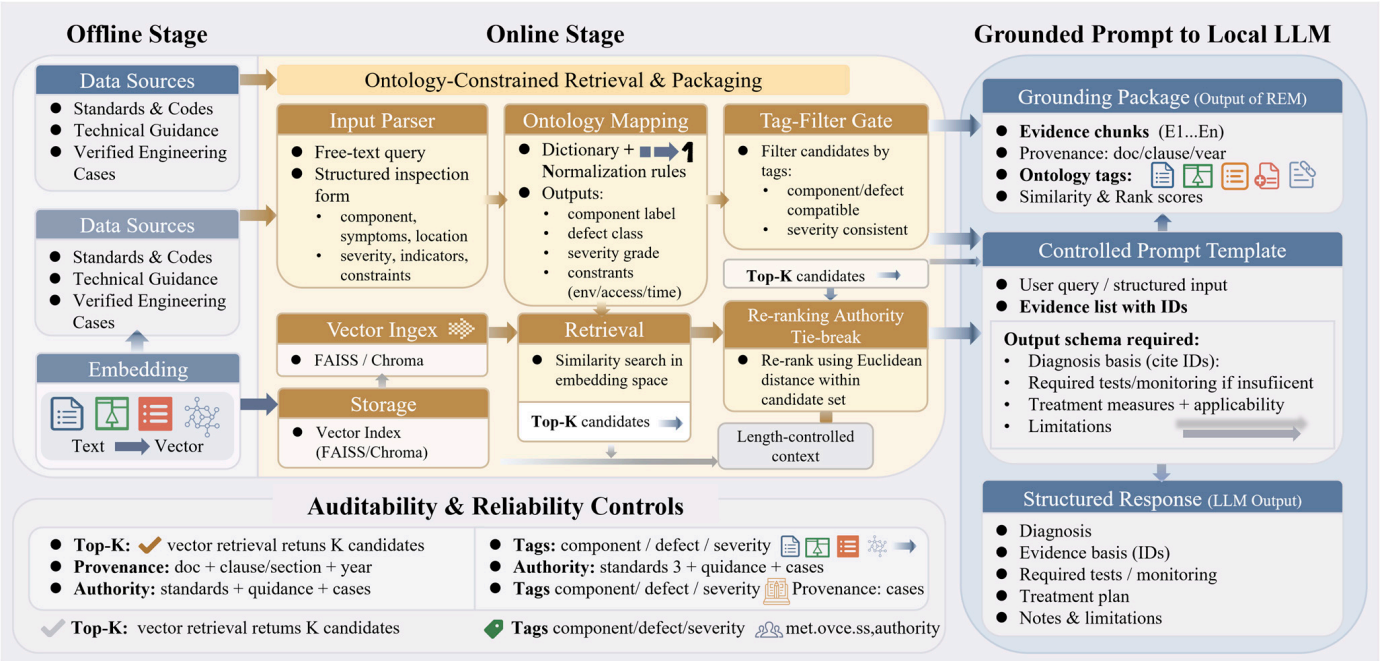


Figure 2. End-to-end workflow of REM-grounded generation in PCBDK-LLM.

4.1. RAG Pipeline Design and Prompting Strategy

RAG enhances reliability by retrieving external knowledge and conditioning the language model on that evidence prior to generation [7]. In PCBDK-LLM, retrieval is performed over a hybrid knowledge base composed of ontology-linearized knowledge-graph entries extracted from standards and technical guidance (Section 3) together with a case base that records defect manifestations, test results, and verified treatments. Each item is stored as a compact chunk with explicit provenance, including the document source, clause or section identifier, and publication year, so that engineers can audit the basis of the generated recommendations.

The online inference pipeline supports two practical input modes. In the free-text mode, inspectors can ask targeted questions such as "What tests are required to confirm duct grouting defects?" In the structured mode, users provide a standardized inspection summary that specifies the component, observed symptoms, location, severity indicators, and relevant constraints. The structured mode is recommended for case-level diagnosis because it reduces ambiguity and ensures that all systems being compared receive the same information during evaluation.

Given an input, the system first performs lightweight query interpretation by mapping key terms to the ontology label space, including component, defect type, severity, and optional constraints. These labels are then passed to REM to retrieve a small set of defect-consistent evidence chunks. The final prompt to the dialogue model includes the user query or structured input, the retrieved evidence chunks with identifiers and provenance, and an output schema that requires the model to provide the diagnosis basis, recommend tests when needed, propose treatment measures with applicability notes, and state explicit limitations. This prompt design follows established mitigation strategies for hallucination in high-stakes domains by restricting generation to retrieved context and encouraging transparent assumptions [11]

To keep responses practical and readable, the evidence context is length-controlled. REM removes duplicates, prioritizes authoritative standards over lower-level sources when conflicts arise, and selects a diverse set of chunks that jointly cover diagnosis criteria, test requirements, and treatment applicability. This design limits redundant context, reduces prompt overload, and improves the stability of multi-turn interactions.

4.2. Ontology-Aware Retrieval Module (REM)

REM serves as the primary grounding component that connects the knowledge layer with the dialogue model. Its core purpose is to prevent technically incompatible retrieval, which can occur when semantically similar passages refer to different components, defect mechanisms, or severity levels. To address this risk, REM combines ontology-tag filtering with efficient vector retrieval and re-ranking.

In the offline stage, each knowledge entry is linearized into a fixed template that captures the component, defect, severity, diagnosis criteria, required tests, recommended treatment, implementation notes, and reference information, and the resulting text is embedded into a vector index. Along with embeddings, each chunk stores ontology tags such as component type, defect class, and severity grade, as well as provenance metadata. In the online stage, REM parses the user input and maps terms to ontology labels using an ontology dictionary and normalization rules. It then applies ontology constraints to restrict candidates to compatible tags, so that severe-level interventions are not retrieved for mild defects unless severe indicators are present. After filtering, REM performs similarity-based retrieval in embedding space to obtain a top candidate set. It subsequently re-ranks candidates by using Euclidean distances within the candidate set to improve robustness to embedding magnitude variability, and it uses provenance authority as a tie-breaker when distances are comparable [14–16]. Finally, REM assembles a compact and diverse evidence set and attaches identifiers to support traceability.

Cosine similarity and Euclidean distance are standard primitives. REM's contribution lies in the domain-constrained workflow that combines ontology-aware filtering with hybrid retrieval and evidence packaging. This design directly addresses the reviewer concern that off-the-shelf RAG may return linguistically relevant but technically incompatible context, which is particularly risky for prestressed concrete bridge diagnosis.

4.3. Integration with a Locally Deployed Dialogue Model

PCBDK-LLM integrates REM with a locally deployed dialogue model, implemented with DeepSeek-R1, to support privacy-preserving deployment in infrastructure agencies. Local deployment avoids transmitting sensitive inspection records and proprietary maintenance plans to external servers, and it supports stable operation in environments with limited connectivity.

During inference, REM produces a grounding package consisting of evidence chunks together with their provenance. The dialogue model is instructed to summarize the defect diagnosis using engineering terminology, cite the evidence items supporting key claims, recommend follow-up tests when evidence is insufficient, and provide a treatment plan that is compatible with defect severity and implementation constraints. Outputs are formatted in a structured manner that includes diagnosis, evidence basis, required tests or monitoring, treatment options, and notes or limitations, which improves readability and reduces the likelihood that critical steps are omitted.

In multi-turn interaction, the system maintains conversational context while rerunning REM when new defect information or test results are introduced. This ensures that subsequent answers remain grounded in defect-consistent evidence rather than relying on long-context speculation. In addition, embeddings and retrieved candidates are cached to reduce repeated retrieval latency when users iteratively refine case descriptions.

The full PCBDK-LLM extends REM-based evidence selection by adding domain knowledge constraints and a structured prompting scheme to generate traceable outputs that remain consistent with engineering practice. In particular, PCBDK-LLM standardizes terminology according to the ontology to ensure component-defect compatibility, invokes additional safety measures when severity indicators warrant them, and compels the model to ground key claims in retrieved evidence by citing corresponding identifiers while organizing the response into explicit sections for diagnosis, required tests, and treatment actions. This setting represents the complete proposed framework.

4.4. Auditability, Reliability Controls, and Failure Handling

Because bridge maintenance is safety-critical, the framework incorporates lightweight reliability controls to enhance auditability. PCBDK-LLM logs the identifiers of retrieved evidence chunks, similarity scores, and provenance, allowing engineers to trace each recommendation to supporting sources. It also applies a small set of ontology-based compatibility checks to the generated output, focusing on severity–treatment consistency and test prerequisites. When a conflict is detected, such as a high-risk intervention being suggested without confirmatory tests, the system flags the issue and recommends additional verification rather than presenting the suggestion as a final decision.

The response template also includes an explicit limitations section. When retrieved evidence does not cover a required decision condition, the model is instructed to state what information is missing and propose a safe next step, such as additional inspection, nondestructive testing, or monitoring, instead of filling gaps with unsupported statements. These design choices aim to deliver a practical assistant that improves efficiency while preserving the conservative decision culture typical of structural engineering practice.

5. Case Analysis

5.1. Construction of Bridge Defects Knowledge Graph

The study constructed a prestressed concrete bridge defect knowledge graph to reflect real engineering decision-making. The sources include Chinese standards and specifications (e.g., JTG H11-2021), peer-reviewed studies, and inspection/maintenance reports. From these materials, the study extracted structured fields such as component, manifesta-

tion, diagnostic criteria or thresholds, severity grade, recommended tests, and treatment options, and then unified terminology across sources. Each item is linked to its provenance (document name and clause/section when available) to support auditability. All entries are converted into a fixed chunk template and tagged with ontology labels (component, defect, severity, test, treatment) so that REM can perform defect-consistent retrieval. Standards are segmented by clause, and case reports are split into description, evidence, treatment, and verification to reduce irrelevant retrieval. Ontology tagging uses a dictionary-based matcher with alias normalization, and severity tags are assigned when thresholds are explicitly provided.

5.2. Implementation Details of REM

REM is an ontology-aware retrieval and grounding layer built on the vector index. For each query, it maps the input into ontology labels, filters candidates by component, defect category, and severity when available, and then retrieves evidence using a hybrid strategy. Cosine similarity is used to select a top-k candidate pool, and Euclidean distance is used for re-ranking within the filtered set. Filtering is conservative, requiring component and defect-family alignment and preferring evidence that matches severity indicators when present. REM also prioritizes chunks that specify prerequisite tests for interventions that depend on internal tendon conditions. When candidates are similar, REM prefers higher-authority sources such as standards and aims to cover diagnosis criteria, test requirements, and treatment applicability together. The final context package contains non-redundant evidence chunks with source identifiers and brief tag summaries for traceable generation.

5.3. Real Case Analysis Based on PCBDK-LLM

The study deployed DeepSeek-R1 locally and integrated it with REM and the pre-stressed concrete bridge defect knowledge graph to form PCBDK-LLM. To evaluate practicality under realistic constraints (no external network access), the study analyzed eight bridges, including the Panlong Bridge (as shown in Figure 3). The study designed four ablation variants under the same evaluation protocol and the same benchmark large language model (DeepSeek-R1 based on Ollama). All models and experiments received the same structured case description (Table 4) prompt and were required to output (i) possible defect mechanisms/causes, (ii) key verification items or tests (if needed), and (iii) a feasible repair plan including main construction steps and safety precautions. The study compared each model's generated rehabilitation suggestion against the reference rehabilitation plan documented in the project materials. To obtain a consistent quantitative proxy for semantic agreement, Similarity in Table 5 is reported as BERTScore F1 multiplied by 100% [17]. BERTScore is used because it measures semantic overlap beyond exact wording and is therefore more suitable than purely lexical metrics for engineering narratives that may use different phrasing for the same action. To focus the evaluation on actionable content, the similarity computation is applied to the treatment/rehabilitation portion of each answer (construction steps and measures) rather than to introductory explanations. The runtime environment and key configuration are summarized in Table 6. The ablation experiment scheme is as follows:

(1) LLM-only: The model receives only the case description and generates the diagnosis and treatment plan without any external evidence retrieval or post hoc rule filtering.
(2) Generic RAG: Using only RAG and LLM to generate the diagnosis and treatment plan.
(3) REM-only: This variant retains the RAG pipeline but replaces generic retrieval with a retrieval enhancement module (REM). REM applies rule- and metadata-aware filtering and reranking before evidence injection, e.g., restricting candidates by component/defect type/severity compatibility and prioritizing higher-authority documents

(e.g., standards/guidelines) over lower-authority sources. Importantly, REM-only only changes how evidence is selected and ordered; it does not introduce additional reasoning constraints beyond the shared output format requirements.

(4) PCBDK-LLM: The full method builds upon REM-enhanced evidence selection and further incorporates domain knowledge constraints and structured prompting to produce traceable, engineering-consistent outputs. Specifically, PCBDK-LLM enforces ontology-consistent terms (component-defect type alignment), activates severity-triggered safety actions when applicable, and requires evidence-grounded outputs (e.g., citing retrieved evidence identifiers) with structured fields for diagnosis, tests, and treatment actions. This variant reflects the complete proposed framework.

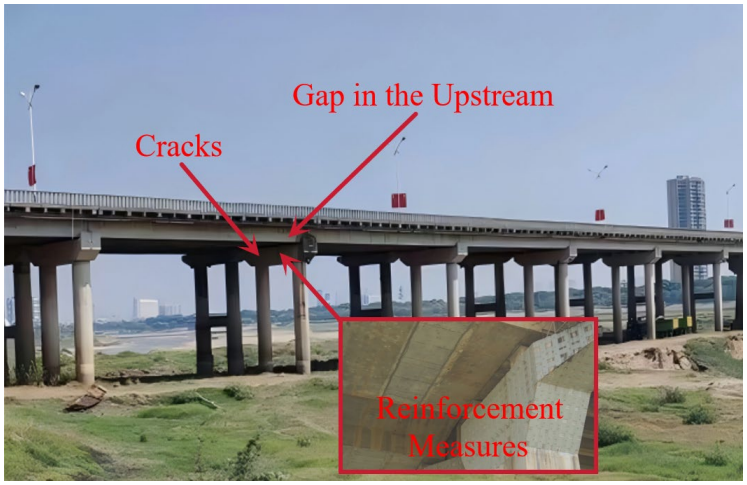


Figure 3. The Studied Bridge (Panlong Bridge).

Table 4. Case input information used for the bridge evaluation (Cases 01–08).

Case 01 Panlong Bridge	**Content Provided to Models**
Bridge type and layout	Prestressed concrete bridge with T-beams/box girders/hollow slabs; total length 1230 m; width 27.5 m.
Observed defect 1	Overall downward movement of two beams and slabs; downstream expansion joint closed; upstream expansion joint gap about 12 cm.
Observed defect 2	Longitudinal crack about 1 cm wide at the transition pier between variable-section box girder and prefabricated T-beam.
Project context	Bridge needs lifting and resetting; maintain traffic safety and restore structural alignment.
Brief output	defect diagnosis (likely causes) and a feasible treatment/rehabilitation plan with key construction steps.
Case 02 Varina-Enon Bridge Interstate 295 (I-295)	Content provided to models
Bridge type and layout	Two parallel 28-span post-tensioned bridges (constructed 1990).
Observed defect 1	Voids in grouted tendons discovered using a borescope probing vent tubes/end caps; voids were attributed to bleeding/segregation of water-cement grout.
Observed defect 2	Case-study summary reports ~45% of tendons had voids, and ~55% were vacuum grouted to fill voids; a tendon failure was identified during inspection (durability risk).
Project context	Durability-driven response: contracts awarded to grout voids (2003–2004), followed by emergency actions after a later tendon failure; access is constrained (tendon/duct inspection through vents/openings).
Brief output	Diagnosis of likely causes (grout bleed/segregation, venting/grouting practice), and a restoration plan emphasizing re-grouting/vacuum grouting, corrosion protection, and verification/acceptance testing (e.g., targeted openings/borescope checks, follow-up nondestructive evaluation (NDE)).

Table 4. *Cont.*

Case 03 Lee Roy Selmon Crosstown Expressway (SR 618) segmental bridges	Content provided to models
Bridge type and layout	Post-tensioned concrete segmental bridge structures on the State Road 618 (SR 618) corridor (controlled-access toll facility; corridor length reported as 14.168 miles).
Observed defect 1	Epoxy grout "pourback" at anchorages experienced cracking and spalling (documented as a field problem on newly constructed segmental bridges).
Observed defect 2	Cracked/spalled pourbacks increase risk of water ingress into anchorage protection details, raising tendon durability concerns if not promptly repaired. (Mechanism discussed in the pourback investigation context.)
Project context	Safety/constructability driven: anchorage protection is critical; repairs typically must be performed under traffic management constraints (urban elevated structures).
Brief output	Immediate safety measures + detailed assessment of anchorage/pourback condition, then repair strategy (remove unsound material, re-cast/repair mortar/pourback system), with strengthening options if needed (carbon fiber reinforced polymer (CFRP)/steel plating locally) and verification/monitoring steps.
Case 04 A689 River Eden Bridge (CNDR)	Content provided to models
Bridge type and layout	Approx. 158 m long; 4-girder composite steel deck supported on a central concrete pier and abutments; central pier has British Standard (BS5400) [18] pot bearings (3 free, 1 fixed).
Observed defect 1	Bearings showed wear; inspection identified defects including over-rotation/misalignment, and exceeded transverse translation capacities.
Observed defect 2	Additional issues around bearing region included cracking on bearing plinth beams and deformation/distortion of bearing components—consistent with abnormal movement demand and loss of proper bearing functionality.
Project context	Required a bearing replacement + bridge jacking scheme with temporary works and controlled movement monitoring; access constraints included site access preparation.
Brief output	Diagnosis of movement/bearing distress and a feasible plan emphasizing bearing replacement, jacking/temporary support scheme, staged traffic handling, and monitoring (lift control + post-replacement displacement checks).
Case 05 US 59 Bridges over Grand Lake (Southbound)	**Content provided to models**
Bridge type and layout	Post-tensioned concrete box girder, 25 spans, total bridge length 3043.8 ft (twin parallel structures; this case focuses on the companion structure).
Observed defect 1	Expansion joint seals leaking at both abutments; debris noted on neoprene seal and a hole allowed water entry (and animal ingress) into the girder interior.
Observed defect 2	Evidence of leakage-related distress included efflorescence along joints, leakage at segment joints, and contamination inside the box girder near expansion joints (durability concern for end-zone components).
Project context	Quick, traffic-sensitive maintenance: stop water ingress, improve joint performance, and prevent progressive end-zone deterioration while maintaining service.
Brief output	Diagnosis of leakage sources and a feasible plan emphasizing expansion joint repair/replacement, drainage/housekeeping improvements, end-zone concrete repair where needed, and protective sealing/coatings, with key construction steps.
Case 06 Dalton 194/093 Bridge	Content provided to models
Bridge type and layout	Small highway bridge (inspection-report case) with bituminous wearing surface over bridge deck; report documents geometry/condition details for maintenance planning.
Observed defect 1	Wearing surface reported rutted with potholes and wide cracks (indicative of fatigue/alligator-type distress and rutting under traffic).
Observed defect 2	Surface cracking/openings present pathways for water infiltration; inspection notes moisture/leakage indicators consistent with seepage-related deterioration risks.
Project context	Maintenance should minimize closures; phased work is preferred, with attention to whether waterproofing is needed and whether underlying deck condition requires intervention.
Brief output	Diagnosis (pavement distress mechanisms + infiltration risk) and plan emphasizing milling & overlay, waterproof layer, drainage detailing, and a check of deck condition (localized repairs if required) with staged traffic control.

Table 4. *Cont.*

Case 07 West Seattle High-Rise Bridge	Content provided to models
Bridge type and layout	Cantilevered segmental box-girder bridge rising ~140 ft, spanning 1340 ft across three spans (major urban link).
Observed defect 1	Routine inspection identified cracking near post-tensioning regions; the case study notes significant cracking issues discovered during inspection.
Observed defect 2	The case study explicitly references extensive diagonal cracking, consistent with web/shear-type crack patterns and the need to track progression (growth over time was a major concern).
Project context	Risk control became urgent (major traffic volumes; long-term cracking history and monitoring). Actions required detailed assessment and staged mitigation compatible with maintaining mobility where possible.
Brief output	Immediate risk control (e.g., restrictions/closures), structural assessment, shear strengthening options (e.g., CFRP U-wrap/steel), crack treatment (seal/inject as appropriate), and monitoring plan (sensors + inspection checkpoints).
Case 08 Channel #5 Bridge	Content provided to models
Bridge type and layout	Overseas Highway bridge group; Channel #5 is Florida Department of Transportation (FDOT) structure 900,098, built 1982, constructed as a precast segmental concrete box girder alternative. Reported bridge length 4580 ft.
Observed defect 1	Coastal/chloride exposure environment: documented need for repairs to substructure components typical of long-term marine exposure (corrosion-driven concrete distress).
Observed defect 2	FDOT project scope explicitly includes repairing columns underneath the bridge (consistent with chloride-related deterioration/spalling repair needs).
Project context	Active bridge repair project in Monroe County with lane-closure management; durability rehabilitation is prioritized under marine exposure conditions.
Brief output	Diagnosis (chloride ingress to corrosion to spalling) and a feasible plan: remove unsound concrete, rebar cleaning/passivation, patch repair, anti-chloride protective coating, and evaluation of cathodic protection + monitoring/inspection schedule.

Table 5. Comparison between PCBDK-LLM and different ablation experiments on the case bridges.

BERTScore-F1/Case	LLM-Only (DeepSeek-R1)	Generic RAG	REM-Only	PCBDK-LLM
Case 01	62.5%	85.1%	89.3%	93.5%
Case 02	60.9%	83.8%	88.1%	92.6%
Case 03	64.0%	86.5%	90.2%	94.2%
Case 04	61.8%	84.9%	89.0%	93.3%
Case 05	63.2%	85.4%	89.7%	93.8%
Case 06	59.7%	82.9%	87.6%	92.0%
Case 07	65.1%	87.1%	90.8%	94.6%
Case 08	62.1%	84.3%	88.7%	93.0%
Average	62.4%	85.0%	89.2%	93.4%

Table 6. Runtime environment and configuration used for all compared models (local execution).

Item	Setting
GPU	NVIDIA GeForce RTX 4060
LLM (local)	DeepSeek-R1 (Ollama)
Embedding model	all-MiniLM-L6-v2
Vector store	Chroma (local)
Decoding/generation parameters	Default settings (no additional tuning)
Timing protocol	End-to-end wall-clock time from query submission to final answer (includes retrieval + generation)
Temperature	0.8
Top-k	40
Repeat penalty	1.1
Num_ctx	2048
Max_new_tokens/num_predict	−1

5.4. Discussion

Compared with the standalone LLM baseline (62.4%), Generic RAG substantially increases semantic agreement (85.0%, +22.6 points), suggesting that injecting retrieved evidence reduces missing steps and improves alignment with reference rehabilitation procedures. REM-only further improves performance (89.2%, +4.2 points), implying that the quality of retrieved evidence matters: metadata filtering and authority-aware reranking likely reduce irrelevant or low-authority passages that can distract generation. Finally, PCBDK-LLM achieves the best result (93.4%, +4.2 points over REM-only), showing that beyond improved retrieval, domain constraints and structured prompting help produce more complete and engineering-consistent recommendations.

Computational overhead is introduced by REM due to ontology label parsing, tag-based filtering, and re-ranking. These steps are lightweight compared with local LLM inference and scale primarily with the corpus size and the chosen top-k for re-ranking. In this study, latency is measured using an end-to-end wall-clock protocol (Table 6) for consistency; however, detailed per-module latency comparisons between variants are not reported here and will be included in future work together with scalability analyses as the knowledge base grows.

Although BERTScore provides a reproducible proxy for semantic agreement, it does not replace expert judgment. Future work will therefore incorporate structured expert scoring and error taxonomy analysis (e.g., missing prerequisite tests, severity–treatment incompatibility, and constructability omissions) to better characterize practical reliability.

6. Conclusions

This study presents PCBDK-LLM, a knowledge-graph-enhanced LLM framework for defect diagnosis and treatment recommendations in prestressed concrete bridges. The proposed framework aims to support practical maintenance decision-making by combining (i) a prestressed bridge defect knowledge graph and ontology; (ii) an ontology-aware retrieval module (REM) for evidence grounding; and (iii) a locally deployed dialogue model for privacy-preserving interaction. Rather than relying solely on the generative capability of an LLM, PCBDK-LLM organizes authoritative engineering knowledge into a retrievable and checkable form and then constrains generation with defect-consistent evidence. Across eight bridge cases, PCBDK-LLM achieves higher semantic agreement with reference rehabilitation plans than LLM-only, Generic RAG, and REM-only baselines (Table 5).

A key contribution is the hybrid knowledge construction strategy that couples standard-driven extraction with ontology-based formalization. Knowledge from specifications, technical guidance, and engineering reports is segmented into traceable chunks and linked to structured entities and relations, while ontology classes and constraints provide stable semantics for retrieval and basic compatibility checks. Building on this foundation, REM improves grounding reliability through a domain-constrained workflow: it extracts ontology labels from user input, filters candidates by component/defect/severity tags, performs efficient similarity-based retrieval followed by re-ranking, and assembles a compact evidence package with provenance. The novelty of REM lies in this ontology-aware retrieval and evidence packaging procedure, not in introducing new similarity primitives.

Future work will therefore focus on (i) expanding the case base and benchmark set; (ii) introducing multimodal inputs (images, nondestructive testing signals, and monitoring time series) and improving the mapping from measurements to severity indicators; (iii) strengthening constraint checking through richer ontology axioms and external consistency checks; (iv) conducting formal expert evaluations with transparent scoring criteria; and (v) investigating generalization to other bridge types (e.g., steel bridges) and multi-

lingual standards integration by extending and aligning the ontology/knowledge graph across domains and languages. Overall, PCBDK-LLM offers a conservative and traceable path for integrating LLMs into bridge maintenance workflows, with the intention to assist engineers rather than replace professional judgment.

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

Funding: This research was funded by Beijing Natural Science Foundation (Grant No. QY25375), National Natural Science Foundation of China (Grant No. 52408145), and Natural Science Foundation of Chongqing (Grant No. CSTB2023NSCQ-MSX1013).

Data Availability Statement: Data will be made available on request.

Acknowledgments: The study is supported by Beijing Natural Science Foundation (Project No. QY25375) and the National Natural Science Foundation of China (Project No. 52408145), and the Natural Science Foundation of Chongqing (Project No. CSTB2023NSCQ-MSX1013). The supports are gratefully acknowledged.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*; Curran Associates: Red Hook, NY, USA, 2020; pp. 1877–1901.
2. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*; Curran Associates: Red Hook, NY, USA, 2022; pp. 27730–27744.
3. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2022**, arXiv:2201.11903. [CrossRef]
4. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Chen, D.; Dai, W.; et al. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* **2023**, *55*, 248. [CrossRef]
5. Maynez, J.; Narayan, S.; Bohnet, B.; McDonald, R. On Faithfulness and Factuality in Abstractive Summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 5–10 July 2020; pp. 1906–1919. [CrossRef]
6. Manakul, P.; Liusie, A.; Gales, M.J.F. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), Singapore, 6–10 December 2023; pp. 9004–9017. [CrossRef]
7. 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. In Advances in Neural Information Processing Systems 33 (NeurIPS 2020). *arXiv* **2020**, arXiv:2005.11401. [CrossRef]
8. Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; Chang, M.-W. REALM: Retrieval-Augmented Language Model Pre-Training. In Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR 119, Vienna, Austria, 12–18 July 2020; pp. 3929–3938.
9. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv* **2022**, arXiv:2203.11171. [CrossRef]
10. Hogan, A.; Blomqvist, E.; Cochez, M.; D'Amato, C.; de Melo, G.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge Graphs. *ACM Comput. Surv.* **2021**, *54*, 1–37. [CrossRef]
11. Gruber, T.R. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis.* **1993**, *5*, 199–220. [CrossRef]
12. W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation. 11 December 2012. Available online: https://www.w3.org/TR/owl2-overview/ (accessed on 29 January 2026).

13. Ministry of Transport of the People's Republic of China. *Specifications for Maintenance of Highway Bridges and Culverts (JTG H11-2021)*; China Communications Press: Beijing, China, 2021. (In Chinese)
14. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992. [CrossRef]
15. Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.-T. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Punta Cana, Dominican Republic, 8–12 November 2020; pp. 6769–6781. [CrossRef]
16. Johnson, J.; Douze, M.; Jégou, H. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data* **2021**, *7*, 535–547. [CrossRef]
17. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the International Conference on Learning Representations (ICLR). *arXiv* **2019**, arXiv:1904.09675. [CrossRef]
18. *BS 5400-9.1:1983*; Steel, Concrete and Composite Bridges. Bridge Bearings. Code of Practice for Design of Bridge Bearings. BSI: London, UK, 1983.

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.