Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Evidence-constrained railway fault question answering with adaptive graph-vector retrieval and conditional answerability control

Rongrong Fu
Yanshan University
Qiyun Su
Yanshan University
Yong Yang
Center of National Railway Intelligent Transportation System Engineer and Technology
Qizheng Hu
hqzcars@163.com

China Academy of Railway Sciences Corporation Limited

Article

Keywords: railway fault diagnosis, knowledge graph, retrieval-augmented generation, hybrid retrieval, domain lexicon, answerability and hallucination control

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11096221/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Evidence-constrained railway fault question answering with adaptive graph-vector retrieval and conditional answerability control

Rongrong Fu[1], Qiyun Su[1], Yong Yang[2] & Qizheng Hu[3,*]

[1] Hebei Key Laboratory of Measurement Technology and Instrumentation, Yanshan University, Qinhuangdao 066004, China

[2] Center of National Railway Intelligent Transportation System Engineer and Technology, Beijing 100081, China

[3] Signal & Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China

* Correspondence: Qizheng Hu (email: hqzcars@163.com).

Abstract

Railway fault question answering must tolerate colloquial maintenance language, preserve equipment relations, and refuse unsupported diagnostic claims. This study presents an evidence-constrained framework combining a railway-domain Entity Dictionary, adaptive graph-vector retrieval, and a conditionally invoked post-retrieval Answerability Guard. Sparse records are expanded into manually reviewed five-field knowledge units, and a maintenance-response-mediated graph preserves the association between observed responses and diagnostic hypotheses. A vector index supplies semantic entry points for incomplete or narrative queries, while typed anchoring, query-mode classification, graph-sufficiency scoring, vector-guided expansion, evidence fusion, quota-based compression, and evidence gating determine the response path. Four 50-question sets were evaluated; V1 and V3 supported development/testing, and V2 and V4 provided fixed-parameter validation. The Full system achieved a four-set macro score of 0.670, including 0.658 and 0.677 on V2 and V4. Pooled gains were 0.0556 for the Entity Dictionary and 0.0500 for the conditionally invoked Guard, with both bootstrap intervals excluding zero. Fusion Control had an unresolved quality gain of 0.0131 but reduced mean context length by 19.6% and end-to-end latency by 23.9%. Unoptimized Vector-only retrieval outperformed Unoptimized Graph-only overall, although Graph-only retained a small Scenario advantage. These results support complementary retrieval sources governed by explicit evidence and refusal controls.

Keywords: railway fault diagnosis; knowledge graph; retrieval-augmented generation; hybrid retrieval; domain lexicon; answerability and hallucination control

1. Introduction

The railway system, a critical component of national transportation infrastructure, is essential for maintaining transportation safety and operational efficiency. However, its inherent complexity and frequent equipment failures pose significant operational challenges. While routine failures disrupt daily operations, severe malfunctions can lead to safety incidents with potential casualties and property damage [1]. Consequently, the efficient identification and management of fault information has emerged as a crucial research focus for enhancing railway system reliability [2].

In the broader context of transport safety, Wang et al. [3] developed an elastic dry electrode for ECG-based real-time driver fatigue detection. The present study addresses railway equipment faults through the interpretation of maintenance records.

Railway signalling maintenance links an observed alarm to a recorded response, trigger condition, plausible direct cause, and component for inspection. These elements are distributed across abbreviated or incomplete field records. A useful QA system must therefore preserve the relation between an observation and the evidence that justifies its answer.

Knowledge graphs (KGs) provide explicit entities, directed relations, and inspectable provenance for QA [4, 5]. Railway studies have used KGs for maintenance organization, onboard-equipment diagnosis, and fault localization [6, 7]. Liu et al. [8] presented an alarm-log-based method for constructing a fault KG for subway onboard train-control equipment. Recent GraphRAG systems retrieve defect or bogie subgraphs for LLM-based diagnosis [9, 10]. Their value nevertheless depends on reliable entity linking and schema coverage: exact graph queries can miss records when technicians use colloquial descriptions or omit canonical equipment terms.

Dense retrieval complements graph lookup by mapping paraphrased questions and records into a shared vector space [11, 12]. However, semantic proximity does not prove that a retrieved alarm, cause, or action supports the question, and concatenating records can blur response-cause boundaries. LLMs and KGs are therefore complementary rather than interchangeable [13]. GraphRAG improves corpus-level reasoning [14], but global summaries alone do not resolve evidence authority, typed anchoring, or refusal in a small diagnostic knowledge base.

Chinese railway terms may be fragmented by general-purpose segmentation, causing graph queries to anchor on generic substrings. Lexicon-aware Chinese NER shows that word-level information can preserve multi-character entities without relying on one segmentation result

[15, 16]. Accordingly, registered alarms and components are retained as typed units before fuzzy or semantic expansion.

Retrieval does not guarantee faithful generation: models can produce claims unsupported by or contradictory to context [17, 18]. Research on unanswerable and selective QA shows that systems must recognize insufficient evidence, especially under domain shift [19, 20]. Although adaptive retrieval and self-reflection can improve factuality [21], safety-relevant refusal should remain traceable to explicit evidence conditions.

These observations motivate three controls. A typed Entity Dictionary anchors registered alarms and components and blocks incidental substring inheritance. Adaptive retrieval selects exact graph traversal when evidence is adequate, uses vectors to propose graph entry points when it is weak, and governs hybrid cases through fusion and quotas. A conditionally invoked Answerability Guard audits only questions matching the literal cues in Section 3.4.2; untriggered questions proceed to constrained generation only after the evidence-sufficiency gate. When invoked, the Guard returns supported, contradicted, insufficient, or unsupported task.

We ask whether typed anchoring improves answer quality, adaptive fusion improves retrieval quality or efficiency, and conditional answerability control reduces unsupported answers. These module tests are separated from the deliberately unoptimized Graph-only versus Vector-only source comparison.

The principal contributions are as follows:

1. Entity Dictionary. We introduce a typed railway-domain lexicon in which canonical alarm names and potentially faulty components serve as strong retrieval anchors. Auxiliary relation and compound terms assist segmentation but cannot independently authorize a graph entity. Longest non-overlapping matching and an unknown-concept gate prevent unsafe substring inheritance.

2. Adaptive fusion retrieval. We propose a fusion retrieval mechanism that combines query-mode classification, a graph-sufficiency score, vector-guided graph expansion, response-level evidence fusion, redundancy penalties, quotas, and field-preserving context compression. Vector retrieval can discover a graph entry point without being promoted automatically to verified diagnostic evidence.

3. Answerability and refusal control. We introduce a conditionally invoked post-retrieval Answerability Guard that separates evidence selection from proposition-level support

assessment. The implemented literal cue set covers selected yes/no expressions, absolute terms, numeric and unsupported-field requests, comparisons, and procedural or seasonal cues. Trigger-matched questions receive deterministic contradiction, insufficiency, and unsupported-task fallbacks; untriggered questions proceed to constrained generation only after the deterministic evidence-availability sentinel check is passed.

2. Related work

2.1. Railway fault knowledge graphs and graph-grounded diagnosis

KGs represent heterogeneous facts as entities, relations, and attributes and support both symbolic queries and graph-based learning [4]. In railway operation and maintenance, Zhang et al. [6] described a train-maintenance KG that integrates ontology design, information extraction, knowledge mapping, storage, and fusion. Liu et al. [7] constructed a KG for train-control onboard equipment from fault analysis reports and used graph-based QA to support maintenance decisions. These studies address fragmented maintenance knowledge and the limited interpretability of purely statistical diagnosis, but their retrieval depends mainly on predefined graph entities and relations.

Recent work has moved from graph construction toward graph-grounded LLM interaction. Miao et al. [9] proposed a GraphRAG framework for train-bogie fault diagnosis in which query representations retrieve structured graph evidence for an LLM. Zhu et al. [10] developed an evidence-verifiable interaction system for high-speed railway infrastructure using entity anchoring, multi-hop graph retrieval, and structured evidence generation. In a neighbouring industrial domain, Liu et al. [22] unified a domain KG and a text-grounded causal-event graph for automotive-chip reasoning and selected graph queries according to question type. Collectively, these studies show that graph structure can improve traceability and domain reasoning. The present work targets a narrower but unresolved design problem: how to route between exact graph evidence and semantic vector retrieval when canonical railway terms may be missing, and how to refuse a proposition that remains unsupported after retrieval.

Topology also affects evidence interpretation. A flat alarm-centred star can imply unrecorded combinations when responses, causes, and components are serialized independently. The proposed topology instead keeps each response as the local parent of its diagnostic hypotheses.

2.2. Domain lexicons and Chinese entity anchoring

Chinese technical terms do not necessarily align with the boundaries learned by a general-purpose segmenter. Zhang and Yang [15] incorporated lexicon-matched words into a lattice

LSTM and showed that word information can improve Chinese NER without committing to one error-prone word segmentation. FLAT later represented characters and lexicon spans in a flat lattice Transformer, improving both efficiency and recognition performance [16]. The common principle is that a lexicon can preserve multi-character entities that would otherwise be fragmented.

Our Entity Dictionary adapts this principle to retrieval control rather than open-vocabulary NER. Alarm and component strings authorize typed graph traversal, relation terms identify requested fields, and frequent compounds stabilize segmentation. A generic token inside an unknown compound is not treated as proof of a registered entity.

2.3. Combining structured and semantic retrieval

RAG was introduced as a way to combine language-model generation with external retrieved memory [11], while DPR demonstrated the practical effectiveness of dense semantic retrieval for open-domain QA [12]. Dense retrieval is tolerant of paraphrase, but its output is a ranked set of passages rather than an explicit relation proof. KGQA provides the complementary ability to follow typed relations and multi-hop paths, although complex questions remain difficult when entity linking or path construction fails [5].

Several systems combine textual and structured resources. PullNet iteratively retrieves from both a knowledge base and a text corpus to build a question-specific subgraph [23]. QA-GNN scores the relevance of KG nodes to a question context and jointly reasons over language and graph representations [24]. The LLM-KG roadmap distinguishes KG-enhanced LLMs, LLM-augmented KGs, and synergized systems in which both resources contribute [13]. GraphRAG constructs graph-based indices and summaries for corpus-level questions [14]. These approaches establish the value of heterogeneous retrieval, but they generally optimize relevance or reasoning performance. The mechanism proposed here adds an evidence-authority policy: exact graph paths are preferred for verification, vectors may guide graph expansion, pure-vector cases remain explicitly unverified, and the final context is compressed at the evidence-block level.

2.4. Grounded generation, answerability, and evaluation

Hallucination can arise even when a generated answer is fluent and topically appropriate [17]. SQuAD formalized the requirement that QA models recognize when a passage does not contain an answer [19], and selective QA under domain shift showed that abstention policies based only on a model's own confidence can be unreliable [20]. RAGTruth demonstrated that retrieved context does not prevent unsupported or contradictory claims [18]. Self-RAG

addressed related problems by learning when to retrieve and by generating reflection signals for relevance, support, and utility [21]. More recent abstention evaluation likewise finds that explicit prompting can help, but strong LLMs still struggle to withhold unsupported answers consistently [25].

These findings motivate a conditional proposition-level audit distinct from retrieval. For trigger-matched questions, the Guard tests whether context supports the claim's direction and strength; CONTRADICTED, INSUFFICIENT, and UNSUPPORTED_TASK lead to fixed fallback behavior.

Evaluation of open-ended answers presents a second reliability problem. Atomic-fact evaluation makes support judgments at the level of individual claims rather than treating an entire response as uniformly factual [26]. LLM-based evaluation can correlate with human judgments, but it may also exhibit model-related bias [27]. The evaluator used in this study therefore organizes reference answers into railway-specific fields, applies deterministic normalization and item matching, reports question-level paired differences, and scores hallucination and refusal cases with explicit stance rules. This design does not eliminate the need for future expert evaluation, but it makes the reported module comparisons reproducible from the saved standards, evidence, and predictions.

3. Methods

Fig. 1 summarizes four stages: LLM-assisted enrichment of sparse records; typed entity-relation extraction and construction of graph and vector resources; adaptive graph-vector retrieval; and evidence-constrained answer generation. The first two stages operate offline. Online processing maps each question to a compact, auditable evidence set before generation.

The graph is organized around observed maintenance responses. Each alarm links to a recorded response, which links to manually reviewed triggers, direct-cause hypotheses, and potentially faulty components. Vector retrieval identifies candidate alarms for incomplete or narrative questions; exact graph traversal then recovers structured evidence.

3.1. LLM-based data enrichment and rewriting

The corpus contained 2,345 sparse records across 55 alarm categories. Each supplied an observed alarm type and maintenance response; the missing trigger, direct cause, and potentially faulty component were added to form a five-field knowledge unit while preserving the observed fields as evidence.

3.1.1. Constrained generation of diagnostic fields

Let source record i be $r_i = (a_i, h_i)$, where a_i denotes the alarm type and h_i denotes the observed maintenance response. Conditioned jointly on a_i and h_i, an LLM generated three supplementary fields: the alarm trigger condition t_i, a direct diagnostic cause c_i, and one or more potentially faulty components p_i. The resulting representation was

$$A(r_i) = z_i = (a_i, h_i, t_i, c_i, p_i).$$

Generation was limited to missing fields and could not alter the alarm label, recorded operation, or introduce an unrelated procedure. Every retained field was manually reviewed, and unsupported content was corrected or removed before model training and knowledge-base construction (Supplementary Table S1).

3.1.2. Record concatenation and controlled rewriting

The five fields formed a fixed-order structured description. To reduce dependence on field position and recurrent phrasing, one natural-language rewrite was generated per record under four constraints: retain every entity, allow syntactic reordering, add no technical fact, and output only the rewritten description.

BAAI/bge-large-zh-v1.5, from the C-Pack family of Chinese text-embedding resources [28], encoded x_i and $x_i^{'}$ as 1,024-dimensional vectors. After L2 normalization, semantic consistency was measured by cosine similarity:

$$\mathrm{sim}(x_i, x_i^{'}) = e_i^T e_i^{'}, \qquad \text{accept if } \mathrm{sim}(x_i, x_i^{'}) \geq 0.80.$$

The similarity threshold screened semantic consistency but did not establish technical correctness, which remained subject to manual review. Of 2,345 records, 2,306 yielded usable rewrites: 2,299 passed automatically and seven truncated outputs were completed manually. The remaining 39 had no usable rewrite. The vector corpus therefore contained 4,651 entries (2,345 structured descriptions and 2,306 rewrites).

3.2. Structured knowledge extraction and construction of graph and vector knowledge bases

The second stage constructs complementary resources: an adapted language model extracts typed entities and directed relations for the graph, while structured descriptions and rewrites are embedded for semantic retrieval. The graph preserves diagnostic direction; the vector index provides entry points when canonical alarm names are absent.

3.2.1. Maintenance-response-mediated diagnostic graph

For each accepted record z_i, the handling-mediated graph contains five node types: Alarm (A), Maintenance response (H), Trigger condition (T), Direct cause (C), and Potentially faulty component (P). The edge set contains one relation from an alarm to a maintenance response and three relations originating from the response node:

$$G = (V, E), \qquad V = A \cup H \cup T \cup C \cup P,$$

$$E = \{(a_i, \text{has_response}, h_i), (h_i, \text{has_trigger}, t_i), (h_i, \text{suggests_cause}, c_i), (h_i, \text{suggests_component}, p_i)\}.$$

Unlike a flat alarm-centred star, the handling-mediated topology preserves which trigger, cause, and component hypothesis belongs to each response. These reviewed hypotheses do not estimate causal strength or intervention effects. Supplementary Fig. S1 illustrates the resulting difference in response-level grouping.

3.2.2. Entity and relation extraction

GLM-4-9B was adapted with LoRA [29, 30] to extract typed entities and triples. Records were split 8:2 by original source so each structured description and its rewrites remained in one partition. Training used Alpaca-style instructions with manually annotated outputs (Supplementary Table S2).

The LoRA rank was 8, alpha was 16, dropout was 0.05, the learning rate was 5×10^{-5}, the maximum sequence length was 1,024 tokens, the batch size was 1, gradients were accumulated over 32 steps, and training lasted 70 epochs.

3.2.3. Entity normalization and graph instantiation

Before insertion, extracted strings were compared with a manually defined alarm-and-component whitelist. Mappings with cosine similarity at least 0.90 were proposed for merging, while ambiguous cases were reviewed manually to limit duplicate or uncertain nodes.

Neo4j supports forward traversal from alarms and reverse traversal from components while preserving response-level associations. The graph contains 6,838 nodes, 12,849 relationships, and 2,345 alarm-response units, each serving as the smallest downstream evidence block.

3.2.4. Vector knowledge base

In parallel, BAAI/bge-large-zh-v1.5 [28] encoded the 4,651 structured and rewritten descriptions. The vectors were L2-normalized and stored in a FAISS inner-product index [31]; for normalized vectors, inner product is equivalent to cosine similarity. The vector knowledge base was not treated as an independent source of verified relations. It primarily identified

candidate alarms for incomplete, colloquial, or scenario-based questions. Exact evidence associated with those alarms was then recovered from Neo4j and organized by maintenance response. In the unanchored Hybrid Fallback, up to two pure-vector cases could remain only with a pending graph verification label; they were never presented as verified diagnostic relations. The fixed models, knowledge resources, and online retrieval limits are summarized in Supplementary Table S3.

3.3. Adaptive retrieval from graph and vector knowledge bases

The third stage selects the least expansive route that can supply relevant evidence. It combines normalization, typed anchoring, query-mode assignment, graph-sufficiency assessment, optional vector-guided expansion, and evidence fusion. Exact graph retrieval is preferred for registered entities; vectors broaden recall for weak scenario evidence or unanchored ordinary questions.

3.3.1. Dictionary-anchored query analysis

A typed dictionary prevents Jieba from fragmenting railway terms. It contains 55 alarm types, 1,723 components, seven relation-field terms, and 15 frequent compounds (1,800 entries). Alarm and component terms provide graph anchors; relation and compound terms assist segmentation and requested-field detection without authorizing traversal.

The system selects longest non-overlapping dictionary matches after whitespace and case normalization. Registered alarms trigger forward traversal and components reverse traversal; an unknown-concept gate blocks incidental substring matches inside unregistered quoted compounds.

3.3.2. Query modes and classification priority

In-domain questions follow the fixed priority Verification → Scenario → Multi-element → Fact. A quoted but unregistered alarm also activates strict exact validation. Scenario questions describe an observed state and request a diagnosis; otherwise, requests for at least two evidence fields are Multi-element and the remainder Fact.

Query mode selects a route but does not determine support. Intent gating intercepts greetings, capability questions, out-of-domain requests, and underspecified inputs. Verification and strict quoted-alarm validation use exact graph lookup without fuzzy component fallback (Supplementary Fig. S2).

3.3.3. Graph sufficiency gate and execution of the five retrieval routes

Five routes execute under fixed priority. Graph Verification handles verification syntax and strict quoted-alarm validation. Scenario questions use the graph-sufficiency gate to select Graph-only or Vector-guided Graph; anchored non-scenario questions use Graph Direct, and unanchored ordinary questions use Hybrid Fallback. These are internal Full-system routes, not ablation conditions (Supplementary Table S4; Fig. 2).

The graph-sufficiency gate is applied only to Scenario questions. Preliminary graph candidates are deduplicated by the pair alarm + maintenance response. Let n_1 and n_2 denote the evidence counts of the two most strongly supported alarms and let N be the number of candidates with an explicit alarm name. Consensus C, support margin M, and independent support U are

$$C = \frac{n_1}{\max(1, N)}, \qquad M = \frac{n_1 - n_2}{\max(1, N)}, \qquad U = \min\left(1, \frac{n_1}{3}\right).$$

Let B_q and B_g be the sets of normalized character bigrams in the question and a candidate evidence record. Candidate relevance is

$$R(B_q, B_g) = \min\left(1, \frac{|B_q \cap B_g|}{\max(1, \min(|B_q|, 12))}\right),$$

and R_e is the maximum relevance among the candidates of the leading alarm. N_o is the proportion of unusually long or unnamed graph candidates. Graph-evidence density is

$$D = \mathrm{clip}_{[0,1]}(0.35C + 0.25M + 0.20U + 0.20R_e - 0.10N_o).$$

State specificity S equals 1.0 for an explicit threshold state, 0.75 for a persistent state, and 0 otherwise. Entity anchoring A equals 1.0 for a registered alarm, 0.4 for a registered component, and 0 otherwise. Temporal complexity $T = \min(1, m/2)$, where m is the number of distinct cues describing short-term change, recovery, switching, interruption, order, concurrence, or cascading events. The final graph-sufficiency score is

$$G_s(q) = \mathrm{clip}_{[0,1]}(0.45S + 0.15A + 0.15R_e + 0.10C + 0.15D - 0.30T).$$

The fixed threshold is $\tau_G = 0.25$. Scores $G_s(q) \geq 0.25$ retain preliminary graph evidence and select Graph-only Route; lower scores discard those candidates and select Vector-guided Graph Route. The gate decides whether vectors are needed for graph entry, not whether a proposition is true, and it does not replace conditional answerability auditing.

Vector-guided Graph Route retrieves at most ten FAISS records and ignores pure-vector scores below 0.20. Parsed alarms are dictionary-validated; the leading alarm is retained, and a second is added only when its score is within 0.20 of the leader:

$$A_v(q) = \{a_1\} \cup \{a_2 : s_1 - s_2 \le 0.20\}, \qquad |A_v(q)| \le 2.$$

Retained alarms are expanded by exact Cypher queries, after which source vector narratives are removed. If dictionary validation or exact expansion yields no graph evidence, the system follows the insufficient-evidence path rather than answering from a similar case (Supplementary Table S5).

3.3.4. Evidence fusion, reranking, and context compression

All routes map candidates to a common five-field evidence record with source, rank, and vector-similarity metadata. Records are deduplicated by normalized alarm-response pair; duplicate source labels are merged, the best rank and similarity retained, and graph fields override same-named vector fields. Candidates conflicting with exact anchors are removed before scoring.

Reranking uses five normalized signals. E is typed entity match, F is coverage of the requested fields, V is normalized vector similarity, B is character-bigram overlap between the question and evidence, and R is the transformed original rank:

$$R = \frac{1}{1 + 0.15(r - 1)}.$$

Graph evidence receives a source bonus $b_{\text{source}} = 0.08$. An exact alarm and an exact component contribute dictionary bonuses of 0.10 and 0.07, respectively, with an additional 0.05 when both match. Fact and Multi-element questions emphasize entity and requested-field matches, whereas Scenario questions assign more weight to semantic and lexical similarity:

$$\text{Score}_{\text{ord}} = 0.40E + 0.20F + 0.15V + 0.15B + 0.10R + b_{\text{source}} + b_{\text{dict}} - P_v - P_a,$$
$$\text{Score}_{\text{scn}} = 0.25E + 0.15F + 0.30V + 0.20B + 0.10R + b_{\text{source}} + b_{\text{dict}} - P_v - P_a.$$

Let L be the combined character count of the response, trigger, cause, and component fields. The verbosity penalty is

$$P_v = \min\left[0.30, \frac{\max(0, L - 800)}{4000}\right].$$

Let n_t, n_c, and n_p be the numbers of trigger, cause, and component items after splitting. The aggregation penalty is

$$P_a = \min\left[0.45, 0.04\max(0, n_t - 3) + 0.04\max(0, n_c - 3) + 0.015\max(0, n_p - 10)\right].$$

At most six records are retained. Graph Verification, Graph Direct, Graph-only, and Vector-guided Graph produce graph-only contexts; Hybrid Fallback allows at most two pure-vector cases labelled pending graph verification. Responses, triggers, causes, components, and pure-vector text are capped at 160, 260, 260, 320, and 240 characters, respectively. Compression

preserves field and response-level boundaries. Fig. 2 summarizes convergence on fusion, quota control, evidence gating, and conditional Guard invocation.

3.4. Evidence-constrained answer generation

The fourth stage serializes selected evidence, applies deterministic evidence gating, conditionally invokes the Guard, and either generates a constrained answer or returns a controlled fallback. The model receives only the current question, requested fields, route, and compressed evidence; history is disabled and temperature is 0.

3.4.1. Structured context serialization and prompt assembly

Each record becomes an independent block keyed by alarm and maintenance response, followed by trigger, cause, and component. This preserves graph path boundaries between responses. Source labels and fusion scores remain in the serialized context, but control variables are not presented as domain facts except for necessary authority labels such as pending graph verification.

$$e_j = [\mathrm{source}_j, \mathrm{alarm}_j, \mathrm{response}_j, \mathrm{trigger}_j, \mathrm{cause}_j, \mathrm{component}_j],$$

$$C(q) = \mathrm{TopK}(\{e_j\}, \mathrm{Score}(e_j, q)), \qquad |C(q)| \le 6.$$

The prompt contains the task boundary, question, requested fields, evidence blocks, and answer rules. It restricts output to requested fields, preserves relation direction, and describes graph hypotheses as possible causes or inspection targets rather than proven root causes. A deterministic sentinel rejects no-reliable-evidence or no-graph-result contexts before generation.

3.4.2. Graph Verification and the Answerability Guard

Graph Verification is an exact-source retrieval policy activated by verification syntax or by strict validation of a quoted but unregistered alarm term. The Answerability Guard is a separate, conditionally invoked post-retrieval audit. If none keyword is present and the evidence-sufficiency sentinel check passes, the system proceeds directly to constrained generation. Graph Verification and Guard invocation are therefore not coextensive: strict quoted-alarm validation can select Graph Verification without invoking the Guard, while a cost, seasonal, or comparison cue can invoke the Guard after another route.

When invoked, the Guard first applies a question-only scope classifier with labels IN_SCOPE and OUT_OF_SCOPE. The supported schema contains alarm type, maintenance response, trigger condition, direct cause, and potentially faulty component. Frequency, uniqueness, normative-action, and absolute-causality claims about these five fields remain IN_SCOPE and

proceed to evidence auditing. Requests for unsupported final products, such as duration, cost, response level, seasonal comparison, responsibility assignment, or a formal SOP, are OUT_OF_SCOPE. An initial OUT_OF_SCOPE result is checked by a second independent prompt; only agreement between both scope decisions is confirmed and mapped to the final verdict UNSUPPORTED_TASK.

For trigger-matched IN_SCOPE questions, the audit returns SUPPORTED, CONTRADICTED, or INSUFFICIENT. Strong frequency, uniqueness, action, and causality claims require direct evidence, and preliminary support receives a rationale-consistency check. Unresolved alternatives, invalid output, label-rationale conflict, or missing cited evidence yield INSUFFICIENT.

For a triggered question, only SUPPORTED permits constrained generation. CONTRADICTED, INSUFFICIENT, and UNSUPPORTED_TASK use fixed fallbacks. Vector similarity cannot establish support, and an unregistered alarm cannot inherit relations merely through a component-name overlap.

3.4.3. Refusal and hallucination control

Hallucination is controlled at entity, source, evidence-availability, proposition, and generation boundaries. Unregistered anchors are rejected; verification routes require graph chains; sentinel contexts stop before audit; triggered claims require proposition-level support; and neither guarded nor unguarded generation may introduce causes, components, or actions absent from C(q).

CONTRADICTED states that a strong claim lacks direct support; INSUFFICIENT requests an alarm name or additional field evidence; and UNSUPPORTED_TASK identifies the five-field knowledge boundary and requests external rules or requirements. All are fixed fallbacks without unconstrained factual generation.

3.4.4. Grounded response generation and audit logging

Generation has two admissible paths. If the Guard is not triggered, GLM-4-9B generates only after the retrieved context passes the deterministic evidence-sufficiency sentinel check. If the Guard is triggered, only a SUPPORTED verdict permits generation from q, C(q).

The model preserves each evidence field's scope and presents multiple supported causes as parallel possibilities with evidence-consistent inspection priorities.

Evaluation logs retain the question, answer, context, retrieval settings, route, source counts, evidence-gate state, fusion scores, timing, and Guard judgment when invoked. Scenario traces

also store $G_s(q)$, its components, threshold decision, and proposed alarms. Stable candidate identifiers, anchor spans, and raw vector similarities were not retained.

3.5. Experimental setup, ablation nomenclature, and evaluation

3.5.1. Datasets, fixed resources, and computing environment

Four 50-question sets (V1-V4) each contained 18 Basic, 10 Multi-element, eight Scenario, eight Hallucination, and six Refusal questions. V1/V3 supported development and testing; V2/V4 used fixed parameters. Because all sets were later inspected, pooled analysis is descriptive rather than a fully blind external test. Latency measurements used one post-warm-up observation per question. The fixed online evaluation parameters are reported in Supplementary Table S6.

3.5.2. Module-level ablations and name alignment

The module-level experiment used the same names in the Method and Results sections. In this nomenclature, Fusion Control refers specifically to the post-route evidence-selection submodule of the broader adaptive retrieval mechanism. Full system denotes the complete four-stage pipeline. w/o Entity Dictionary removes dictionary segmentation, typed alarm/component anchoring, and the unregistered-concept gate. In w/o Fusion Control, route selection and final source-authority policies are preserved while evidence-level deduplication, fusion scoring, redundancy penalties, and quota-based selection are disabled. For Graph Direct, Graph Verification, Vector-guided Graph, and graph-sufficiency-gated policies, graph records are retained in their original candidate order. For routes permitting both graph and vector evidence in the final context, candidates are alternated by within-source retrieval rank until the six-record limit is reached. The reported 200-question w/o Fusion run contained 115 Graph Direct, 31 graph-sufficiency-gated, 34 Graph Verification, and 20 intent-rejection cases, with no Hybrid Fallback case; the dual-source interleaving branch was therefore not activated. w/o Answerability Guard keeps retrieval and generation settings unchanged but disables the conditional proposition-level audit and its controlled fallback decisions. The exact ablation conditions and corresponding method components are aligned in Supplementary Table S7.

Graph-only Route denotes the internal Scenario route selected at $G_s(q) \geq 0.25$, not the experimental Graph-only baseline; Graph Verification and Graph Direct are likewise internal routes.

3.5.3. Graph-topology comparison under optimized Graph-only evaluation

The graph-topology experiment compared the handling-mediated and flat alarm-star graphs under an optimized Graph-only configuration. Both arms retained the same source records, five-field schema, Cypher top-k of 10, evidence fusion, quota-based compression with a maximum of six final evidence records, GLM generator, conditionally invoked Answerability Guard, question order, evaluator, and temperature of 0. Vector and keyword retrieval were disabled (top-k = 0); adaptive routing, graph-sufficiency gating, and vector-guided expansion were also disabled so that a failed gate could not clear graph evidence when no vector branch was permitted. The sole experimental difference was graph topology: Alarm → Maintenance response → {Trigger, Cause, Component} versus Alarm → {Maintenance response, Trigger, Cause, Component}.

3.5.4. Unoptimized source-only baselines

Unoptimized Graph-only used ordinary Jieba segmentation and raw fuzzy Neo4j retrieval; Unoptimized Vector-only returned raw FAISS rankings. Both returned ten records and used the same generator, evaluator, question order, and temperature.

Both source-only baselines disabled the dictionary, typed anchoring, unified candidates, fusion and reranking, quotas, compression, intent and sufficiency gates, vector-guided expansion, the Guard, fixed insufficiency responses, and the optimized prompt.

3.5.5. Scoring and statistical analysis

Evaluator organizes Basic, Multi-element, and Scenario references into response, trigger, cause, and component fields [26]. After synonym, case, punctuation, and whitespace normalization, reference items and two-to-six-character prediction fragments are matched. For field e, n_e, k_e, and m_e denote reference, predicted-fragment, and matched-item counts; recall, heuristic soft precision, and F1 are:

$$R_e = \frac{m_e}{n_e},$$

$$P_e = \min\left\{1, \frac{m_e}{\max\left[1, \min\left(2n_e, m_e + \left\lfloor \frac{k_e - m_e}{3} \right\rfloor\right)\right]}\right\},$$

$$F1_e = \frac{2P_eR_e}{P_e + R_e}, \qquad s_i = \frac{1}{h}\sum_e F1_{i,e},$$

where h is the number of available answer fields. Field scores are rounded to three decimals before equal-weight averaging. Missing-knowledge and unsupported-procedure items score 1.0

for refusal without fabricated steps, 0.4 for refusal followed by unverified procedures, and 0 otherwise. For supported relations, an evidence-marked correct relation scores 1.0, an unmarked relation 0.6, an incorrect rejection 0.2, and other outputs 0. For other absolute claims, a non-conflicting negative stance scores 0.75 plus 0.25 when sufficiently evidenced; mixed affirmation and negation scores 0.4, and acceptance scores 0. Refusal items score 1.0, 0.5, or 0 for full, partial, or unsuccessful compliance. Dataset scores average 50 questions.

For paired differences, positive, zero, and negative values are wins, ties, and losses. A 95% percentile interval resamples question-level differences 100,000 times (seed 20260722).

4. Results

4.1. Knowledge construction results

4.1.1. Data enrichment and rewrite consistency

The 2,345 source records yielded 2,306 usable rewrites (98.34%): 2,299 passed automatically, seven truncated outputs were completed manually, and 39 records retained only structured descriptions. The final index contained 4,651 texts (Supplementary Table S8).

Automatically accepted pairs had mean similarity 0.9367, median 0.9400, interquartile range 0.9261-0.9505, and minimum 0.8135. The seven manual completions were excluded from this distribution, and all retained generated fields were reviewed before graph construction.

4.1.2. Entity and relation extraction

Entity and relation extraction achieved F1 scores of 98.84% and 98.81%, respectively. The final resources contained 6,838 graph nodes, 12,849 relationships, 2,345 alarm-response units, and 4,651 FAISS records; source records and their rewrites remained in one partition (Supplementary Table S9).

4.2. Full-system question-answering performance

Full-system scores were 0.680, 0.658, 0.665, and 0.677 on V1-V4, with a macro-average of 0.670. Macro category scores were 0.552 for Basic, 0.567 for Multi-element, and 0.485 for Scenario; Hallucination and Refusal both scored 1.000 throughout (Table 1; Supplementary Fig. S3).

4.3. Module contribution analysis

4.3.1. Entity Dictionary recognition and per-set module effects

The controlled dictionary check covered 110 alarms and 300 components and achieved 1.000 exact entity-set accuracy, precision, recall, and F1. It verifies longest matching for registered terms, not robustness to misspellings, colloquialisms, abbreviations, or unseen terms.

The Entity Dictionary had a positive point estimate on all four sets. Its largest effect occurred on V1, where Full system exceeded w/o Entity Dictionary by 0.1289. The intervals were above zero on V1 and V2 but crossed zero on V3 and V4. Fusion Control produced three positive point estimates and one small negative estimate; each single-set interval included zero or had zero at its boundary. Under the implemented conditional trigger policy, the Answerability Guard improved every set and did not reduce the score of any question. Most comparisons were ties because the improvement was concentrated in Hallucination and Refusal items. Per-set paired effects are reported in Supplementary Table S10.

4.3.2. Pooled effects across 200 questions

Across 200 questions, the paired gain associated with the Entity Dictionary was 0.0556, with a 95% bootstrap interval from 0.0358 to 0.0778. The conditionally invoked Answerability Guard produced a gain of 0.0500 under the implemented trigger policy, with an interval from 0.0200 to 0.0800. Both intervals were above zero. The pooled answer-quality difference for Fusion Control was 0.0131, with an interval from -0.0038 to 0.0333. The latter interval crossed zero, so this comparison did not resolve an answer-quality improvement from Fusion Control, although its efficiency effects were substantial. Pooled effects are reported in Table 2 and visualized with their bootstrap intervals in Fig. 3.

Only 33 of 200 Full-system records stored a Guard verdict: 21 CONTRADICTED, six SUPPORTED, three UNSUPPORTED_TASK, and three INSUFFICIENT. Invocations followed Graph Verification (26), Graph Direct (six), or sufficiency-gated routing (one); eight other Graph Verification cases bypassed the Guard. Intent gating handled 20 questions, while 143 untriggered questions generated after the sentinel check and four failed it. Thus, +0.0500 estimates the literal-cue-triggered Guard, not universal auditing.

4.4. Retrieval efficiency and graph structure

4.4.1. Context and latency effects of Fusion Control

Mean end-to-end latency across the 200 questions was 8.96 s for Full system and 11.77 s for w/o Fusion Control. The mean saving was 2.81 s per question, with a 95% interval from 1.36

to 4.41 s, corresponding to a 23.9% reduction. Mean context length decreased from 1,204.6 to 968.0 characters, a reduction of 236.6 characters, or 19.6%. The two conditions returned the same mean number of graph records on every set. The efficiency gain therefore arose mainly from deduplication, reranking, redundancy penalties, and quota control. The paired efficiency values are reported in Supplementary Table S11 and summarized in Fig. 4a-b.

4.4.2. Handling-mediated and flat graph topologies under the optimized Graph-only evaluation

The pooled graph-only score was 0.6642 for the handling-mediated graph and 0.6624 for the flat alarm-star graph. Their difference was 0.0018, and the interval from -0.0220 to 0.0287 crossed zero. The experiment therefore did not establish an overall answer-quality advantage. Category results showed a trade-off. The flat graph scored higher on Basic and Multi-element questions, whereas the handling-mediated graph improved Scenario performance by 0.0959. The latter finding is consistent with the value of preserving the association between each maintenance response and its diagnostic hypotheses. Category-level graph-topology comparisons are reported in Table 3.

The flat graph returned fewer numbered bundles because each bundle aggregated all visible neighbours of an alarm; bundle count therefore understated its evidence volume. Flat contexts contained 6,015 parsed field items, compared with 2,493 for the handling-mediated graph. The handling-mediated structure reduced implied response-cause candidates by 97.52%, response-component candidates by 97.01%, and complete-record candidates by 99.28%. The 741,008 flat candidates represent possible Cartesian interpretations of serialized field sets, not records directly retrieved from Neo4j. Generic maintenance responses could still accumulate several cause or component neighbours, so the proposed structure reduced association ambiguity but did not eliminate it completely. The deterministic serialization audit is reported in Supplementary Table S12.

4.4.3. Stage-level and end-to-end latency

Handling mediation increased Neo4j retrieval time from 0.044 to 0.191 s but reduced context and answer lengths by 40.5% and 71.2%. Generation latency fell from 30.582 to 8.903 s and end-to-end latency from 30.633 to 9.101 s (70.3%); P95 fell from 69.84 to 21.17 s (Supplementary Table S13; Fig. 4c).

4.5. Unoptimized Graph-only versus Vector-only source ablation

4.5.1. Overall paired results

Unoptimized Vector-only obtained a higher overall score on all four sets. The Graph-minus-Vector differences ranged from -0.046 to -0.076. The paired intervals excluded zero on V2 and V3, while the V1 and V4 intervals reached or crossed zero. Across all 200 questions, Unoptimized Graph-only scored 0.449 and Unoptimized Vector-only scored 0.506. The pooled paired difference was -0.056 (95% CI: -0.089 to -0.026), with Graph-only winning 56 questions, tying 52, and losing 92. Thus, when every domain-specific and evidence-control mechanism was removed, raw vector retrieval provided broader average access to answer-relevant wording than raw fuzzy graph retrieval. Overall source-only results are reported in Table 4.

4.5.2. Category-level trade-off

The source-level comparison was not uniform across question types. Raw vector retrieval had its clearest advantage on Basic questions and on the designed Hallucination and Refusal categories. Raw graph retrieval retained a small Scenario advantage of 0.018, suggesting that even an unoptimized graph can preserve useful local diagnostic structure when the field description overlaps graph content. However, both unoptimized systems performed poorly on Refusal questions, and neither included the implemented conditional answerability mechanism. The higher Hallucination score of the vector baseline therefore reflects the evaluated outputs of this dataset and should not be interpreted as evidence that vector similarity is intrinsically safe. Category-level values are reported in Supplementary Table S14, and their complementarity is visualized in Supplementary Fig. S3.

The source audit found no cross-source violations. Graph-only returned graph evidence for 184 questions and none for 16, while Vector-only returned ten vector records for every question; neither contained evidence from the other source.

Full system exceeded Vector-only and Graph-only by 0.164 and 0.221, but these descriptive gaps cannot be assigned to one innovation because many controls differ. Individual contributions are therefore supported by the module ablations in Section 4.3.

4.6. Summary of the results

The experiments produced four main findings. First, data enrichment and structured extraction generated a large, internally consistent knowledge resource, with entity and relation F1 scores above 98.8%. Second, the Entity Dictionary and the conditionally invoked Answerability Guard showed positive pooled module effects, whereas the clearest contribution of Fusion

Control was reduced context length and latency. Third, the handling-mediated topology improved Scenario performance and sharply reduced the number of ambiguous combinations implied by serialized evidence, while the two topologies had similar pooled answer scores. Fourth, raw Vector-only retrieval outperformed raw Graph-only retrieval in the deliberately unoptimized source isolation, while the small Scenario advantage of Graph-only and the substantially higher Full-system score support the use of complementary sources under explicit routing and evidence control.

5. Discussion

5.1. What the ablations establish

The results support two of the three proposed modules as direct answer-quality contributors and identify a different role for the third. Removing the Entity Dictionary reduced the pooled score by 0.0556, and the corresponding 95% bootstrap interval remained above zero. The gain was largest on V1 and remained positive on all four sets. This pattern is consistent with the dictionary's intended function: preserving complete railway alarm and component expressions changes which graph entity is entered, so an anchoring error propagates through every later stage. The distribution of these module effects is shown in Fig. 3.

The conditionally invoked Answerability Guard produced a pooled gain of 0.0500 with no question-level losses. Its effect was concentrated in Hallucination and Refusal cases, which explains the large number of ties. The result agrees with selective-QA research showing that retrieval and answerability prediction are separate sources of error [19, 20]. The safety-category scores reflect performance on 56 designed items across the four sets; generalization to other railway queries remains to be evaluated.

Fusion Control primarily improved efficiency: compared with w/o Fusion Control, it removed an average of 236.6 context characters and saved 2.81 s per question. Evidence deduplication, redundancy penalties, source quotas, and field-level compression reduced prompt burden without changing the mean number of returned graph records. These results support its role in context compression and latency reduction (Fig. 4a-b).

5.2. Why adaptive graph-vector retrieval is preferable to a fixed source

In the unoptimized comparison, Vector-only exceeded Graph-only by 0.056 because semantic retrieval better matched paraphrases without canonical keys [11, 12].

The category pattern shows complementarity rather than a universal ranking. Raw vector retrieval was stronger on Basic questions, while raw graph retrieval retained a 0.018 Scenario

advantage. Scenario questions often describe a state that maps to a locally coherent alarm-response structure; once an adequate graph entry is found, traversing the stored relations can preserve useful diagnostic grouping. Vector retrieval is more useful when the graph entry itself is missing from the wording. This is the rationale for the graph-sufficiency gate: it does not ask whether graph retrieval or vector retrieval is globally superior, but whether the current preliminary graph evidence is adequate enough to avoid broader semantic expansion. This source-specific pattern is visualized in Supplementary Fig. S3.

The five internal routes implement different evidence policies. Graph Verification requires exact graph support for the requested relation and also performs strict exact validation of quoted but unregistered alarm terms. Graph Direct uses a registered typed entity for forward or reverse traversal. Graph-only Route is selected for a Scenario query whose preliminary graph score reaches the fixed threshold. Vector-guided Graph Route uses semantic retrieval only to propose registered alarms and then replaces the source narratives with exact graph chains. Hybrid Fallback is reserved for unanchored ordinary questions and may retain at most two vector cases marked as pending graph verification. These routes all access the same underlying resources, but they differ in trigger condition, permitted evidence source, and authority assigned to the result. This separation makes the selected route auditable and prevents the label "hybrid" from obscuring whether a particular answer was ultimately supported by graph evidence or by an unverified vector case. Their decision sequence and common evidence policy are summarized in Fig. 2.

5.3. Graph topology, evidence ambiguity, and latency

The handling-mediated topology improved Scenario performance by 0.0959 and sharply reduced the number of response-cause, response-component, and complete-record combinations implied by serialized context. A flat graph can return all neighbours of an alarm correctly while still making their local correspondence ambiguous after serialization. The handling-mediated topology preserves a smaller evidence unit: one recorded response together with the hypotheses associated with that response. The structural distinction underlying this comparison is illustrated in Supplementary Fig. S1.

This structural precision has a cost. Graph retrieval itself was 0.147 s slower on average. The cost was outweighed by a much larger reduction in generation time because the resulting context and answer were shorter. On 144 knowledge-dependent questions, mean end-to-end latency fell from 30.633 to 9.101 s. The result illustrates an important systems point: optimizing a RAG pipeline for retrieval speed alone can be misleading when evidence

organization changes downstream token volume. Here, a slightly slower graph query produced a substantially faster complete answer because it avoided a combinatorial and verbose context. The stage-level trade-off is shown in Fig. 4c.

6. Conclusion

This study presented a railway fault QA framework combining a typed Entity Dictionary, adaptive graph-vector retrieval, and a conditionally invoked post-retrieval Answerability Guard. Response-mediated graph evidence preserves local diagnostic associations, vectors recover semantic entry points, and fusion compresses candidates into an auditable context. Triggered questions undergo proposition-level auditing; untriggered questions generate only after deterministic evidence gating.

Across four 50-question sets, the Full system scored 0.670. The Entity Dictionary and conditional Guard had positive pooled gains, while Fusion Control mainly reduced context and latency. Raw vectors outperformed raw graphs overall, but graphs retained a small Scenario advantage. Handling mediation reduced ambiguity and generation latency while improving Scenario performance.

Reliable railway QA therefore benefits from semantic discovery, relation-preserving graph traversal, evidence-level context governance, and conditional answerability control. Future external validation should test the fixed trigger policy on unseen data, broader adversarial queries, and multi-expert judgments while reporting selective coverage as well as accuracy.

Data availability

The data supporting the findings of this study are part of an ongoing project and are not currently publicly available due to corporate confidentiality requirements. Researchers may submit reasonable requests for access by emailing the corresponding author, Qizheng Hu, at hqzcars@163.com. Requests will be reviewed by China Academy of Railway Sciences Corporation Limited, the corresponding author's institution, and access is subject to its approval and applicable confidentiality requirements. The corresponding author will respond to requests within 72 hours of receipt.

Code availability

The code used in this study is not publicly available and cannot be shared due to restrictions on the distribution of proprietary software.

References

[1] Wang, C. (2025). Research on coasting control and ride comfort optimization of high-speed railway automatic train operation system. *Railway Signalling & Communication, 61*(12), 21–27. https://doi.org/10.13879/j.issn.1000-7458.2025-12.25296

[2] Wang, H., Ma, Z., Liang, C., Wu, R., & Song, G. (2025). Discussion of objectives and methods of railway hub signal design. *Railway Signalling & Communication, 61*(11), 91–96. https://doi.org/10.13879/j.issn.1000-7458.2025-11.25225

[3] Wang, F., Yao, W., Lu, B., & Fu, R. (2024). ECG-based real-time drivers' fatigue detection using a novel elastic dry electrode. *IEEE Transactions on Instrumentation and Measurement, 73*, Article 9502916. https://doi.org/10.1109/TIM.2023.3343799

[4] Hogan, A. et al. Knowledge graphs. *ACM Comput. Surv.* **54,** 71:1-71:37 (2021). https://doi.org/10.1145/3447772.

[5] Lan, Y. et al. Complex knowledge base question answering: A survey. *IEEE Trans. Knowl. Data Eng.* **35,** 11196-11215 (2023). https://doi.org/10.1109/TKDE.2022.3223858.

[6] Zhang, J., Jiang, S. & Liu, W. Construction and application of knowledge graph for train maintenance field. *Urban Mass Transit* **27,** 198-202 (2024). https://doi.org/10.16037/j.1007-869x.2024.09.035.

[7] Liu, D., Zhang, Z., Zhai, Q. & Yu, J. Knowledge graph construction and application for fault diagnosis of onboard equipment of train control systems. *Railw. Stand. Des.* **69,** 184-192 (2025). https://doi.org/10.13238/j.issn.1004-2954.202307060004.

[8] Liu, Y., Wang, X., Jiang, L., & Luo, Z. (2026). Construction method of fault knowledge graph for subway on-board train control equipment based on alarm logs. *Railway Signalling & Communication, 62*(9), 17–26. https://doi.org/10.13879/j.issn.1000-7458.2026-09.26069

[9] Miao, R., Wu, T. & Zhang, Z. Graph RAG-based fault diagnosis for train bogies using knowledge graphs and large language model. *Knowl.-Based Syst.* **331,** 114855 (2026). https://doi.org/10.1016/j.knosys.2025.114855.

[10] Zhu, Z. et al. Evidence-verifiable intelligent interaction system and defect knowledge graph for operation-and-maintenance diagnosis of high-speed railway infrastructure. *Expert Syst. Appl.* **323,** 132537 (2026). https://doi.org/10.1016/j.eswa.2026.132537.

[11] Lewis, P. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* **33,** 9459-9474 (2020). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[12] Karpukhin, V. et al. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP* 6769-6781 (2020). https://doi.org/10.18653/v1/2020.emnlp-main.550.

[13] Pan, S. et al. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.* **36,** 3580-3599 (2024). https://doi.org/10.1109/TKDE.2024.3352100.

[14] Edge, D. et al. From local to global: A graph RAG approach to query-focused summarization. Preprint at https://arxiv.org/abs/2404.16130 (2024).

[15] Zhang, Y. & Yang, J. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* 1554-1564 (2018). https://doi.org/10.18653/v1/P18-1144.

[16] Li, X., Yan, H., Qiu, X. & Huang, X. FLAT: Chinese NER using flat-lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* 6836-6842 (2020). https://doi.org/10.18653/v1/2020.acl-main.611.

[17] Ji, Z. et al. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **55,** 248:1-248:38 (2023). https://doi.org/10.1145/3571730.

[18] Niu, C. et al. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics* 10862-10878 (2024). https://doi.org/10.18653/v1/2024.acl-long.585.

[19] Rajpurkar, P., Jia, R. & Liang, P. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* 784-789 (2018). https://doi.org/10.18653/v1/P18-2124.

[20] Kamath, A., Jia, R. & Liang, P. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* 5684-5696 (2020). https://doi.org/10.18653/v1/2020.acl-main.503.

[21] Asai, A., Wu, Z., Wang, Y., Sil, A. & Hajishirzi, H. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations* (2024). https://openreview.net/forum?id=hSyW5go0v8.

[22] Liu, S., Xie, H., Lei, Y., Zhao, J. & Song, X. LLM-augmented causal-knowledge heterogeneous graph framework for interpretable reasoning and collaborative knowledge fusion in automotive chip production. *Expert Syst. Appl.* **311,** 131343 (2026). https://doi.org/10.1016/j.eswa.2026.131343.

[23] Sun, H., Bedrax-Weiss, T. & Cohen, W. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of EMNLP-IJCNLP* 2380-2390 (2019). https://doi.org/10.18653/v1/D19-1242.

[24] Yasunaga, M., Ren, H., Bosselut, A., Liang, P. & Leskovec, J. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of NAACL-HLT* 535-546 (2021). https://doi.org/10.18653/v1/2021.naacl-main.45.

[25] Madhusudhan, N., Madhusudhan, S. T., Yadav, V. & Hashemi, M. Do LLMs know when to not answer? Investigating abstention abilities of large language models. In *Proceedings of COLING* 9329-9345 (2025). https://aclanthology.org/2025.coling-main.627/.

[26] Min, S. et al. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of EMNLP* 12076-12100 (2023). https://doi.org/10.18653/v1/2023.emnlp-main.741.

[27] Liu, Y. et al. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of EMNLP* 2511-2522 (2023). https://doi.org/10.18653/v1/2023.emnlp-main.153.

[28] Xiao, S. et al. C-Pack: Packed resources for general Chinese embeddings. Preprint at https://arxiv.org/abs/2309.07597 (2024).

[29] Hu, E. J. et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations* (2022). https://openreview.net/forum?id=nZeVKeeFYf9.

[30] GLM Team et al. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. Preprint at https://arxiv.org/abs/2406.12793 (2024).

[31] Johnson, J., Douze, M. & Jegou, H. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **7,** 535-547 (2021). https://doi.org/10.1109/TBDATA.2019.2921572.

Acknowledgements

We thank China Academy of Railway Sciences Corporation Limited for its support of this research.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62073282 and Grant 12332002, in part by the Natural Science Foundation of Hebei Province under Grant F2022203092, in part by the S&T Program of Qinhuangdao City under Grant 202302B015, in part by the China Academy of Railway Sciences Group Co., Ltd under Grant 2024YJ179, and in part by the Project of the State Key Laboratory for Novel Software Technology under Grant KFKT2025B88.

Author contributions

Rongrong Fu contributed to the conceptualization, study design and supervision. Qiyun Su contributed to data preparation, software development, experiments, data analysis and drafting the manuscript. Yong Yang contributed domain expertise, reviewed the data and assisted with the interpretation of the results. Qizheng Hu contributed to study design, supervision and manuscript revision. All authors reviewed and approved the final manuscript.

Additional information

Competing interests

The authors declare no competing interests.

Supplementary information

Supplementary Information accompanies this manuscript and contains Supplementary Tables S1–S14 and Supplementary Figures S1–S3.

Tables

Table 1. Overall and category scores of the Full system.

Set	Role	Overall	Basic	Multi-element	Scenario	Hallucination	Refusal
V1	Development/test	0.680	0.569	0.592	0.480	1.000	1.000
V2	Validation	0.658	0.520	0.582	0.462	1.000	1.000
V3	Development/test	0.665	0.548	0.534	0.507	1.000	1.000
V4	Validation	0.677	0.571	0.561	0.493	1.000	1.000
Macro	Descriptive	0.670	0.552	0.567	0.485	1.000	1.000

Each set contains 50 questions: 18 Basic, 10 Multi-element, eight Scenario, eight Hallucination and six Refusal items. Macro denotes the equally weighted mean across V1–V4; pooled interpretation is descriptive.

Table 2. Descriptive module effects pooled across the four sets.

Method component	Pooled delta	95% CI	W/T/L	Positive sets
Entity Dictionary	+0.0556	[+0.0358, +0.0778]	93/70/37	4/4
Fusion Control	+0.0131	[-0.0038, +0.0333]	60/74/66	3/4
Answerability Guard	+0.0500	[+0.0200, +0.0800]	10/190/0	4/4

Delta = Full system − ablated condition. Intervals are 95% percentile bootstrap intervals from 100,000 paired question-level resamples (seed 20260722; n = 200). W/T/L = wins/ties/losses. Pooled estimates are descriptive across the four inspected question sets.

Table 3. Graph-only answer-quality comparison of the two graph topologies under the optimized evaluation pipeline.

Scope	n	Handling-mediated	Flat alarm-star	Delta	95% CI	W/T/L
Overall	200	0.6642	0.6624	+0.0018	[-0.0220, +0.0287]	41/58/101
Basic	72	0.5519	0.6148	-0.0628	[-0.0874, -0.0386]	12/5/55
Multi-element	40	0.5673	0.6218	-0.0545	[-0.0712, -0.0378]	5/1/34
Scenario	32	0.4506	0.3547	+0.0959	[+0.0315, +0.1629]	20/0/12
Hallucination	32	1.0000	0.9063	+0.0938	[+0.0000, +0.2188]	3/29/0
Refusal	24	1.0000	0.9583	+0.0417	[+0.0000, +0.1250]	1/23/0

Delta = handling-mediated − flat alarm-star. Intervals are 95% percentile bootstrap intervals based on 100,000 paired question-level resamples (seed 20260722). n is the number of questions; W/T/L = wins/ties/losses for the handling-mediated topology.

Table 4. Unoptimized Graph-only versus Unoptimized Vector-only results on V1-V4.

Dataset	Unoptimized Graph-only	Unoptimized Vector-only	Graph − Vector	95% bootstrap CI	W/T/L
V1	0.419	0.468	-0.048	[-0.109, +0.003]	14/16/20
V2	0.457	0.503	-0.046	[-0.098, -0.001]	13/14/23
V3	0.446	0.501	-0.055	[-0.120, -0.007]	15/12/23
V4	0.475	0.551	-0.076	[-0.160, +0.004]	14/10/26
Macro/pooled	0.449	0.506	-0.056	[-0.089, -0.026]	56/52/92

Delta = Unoptimized Graph-only − Unoptimized Vector-only. Intervals are 95% percentile bootstrap intervals based on 100,000 paired question-level resamples (seed 20260722; n = 50 per set and n = 200 pooled). W/T/L = wins/ties/losses for Graph-only. Differences use the original reported precision and may not equal subtraction of the displayed rounded means.

Figure 1

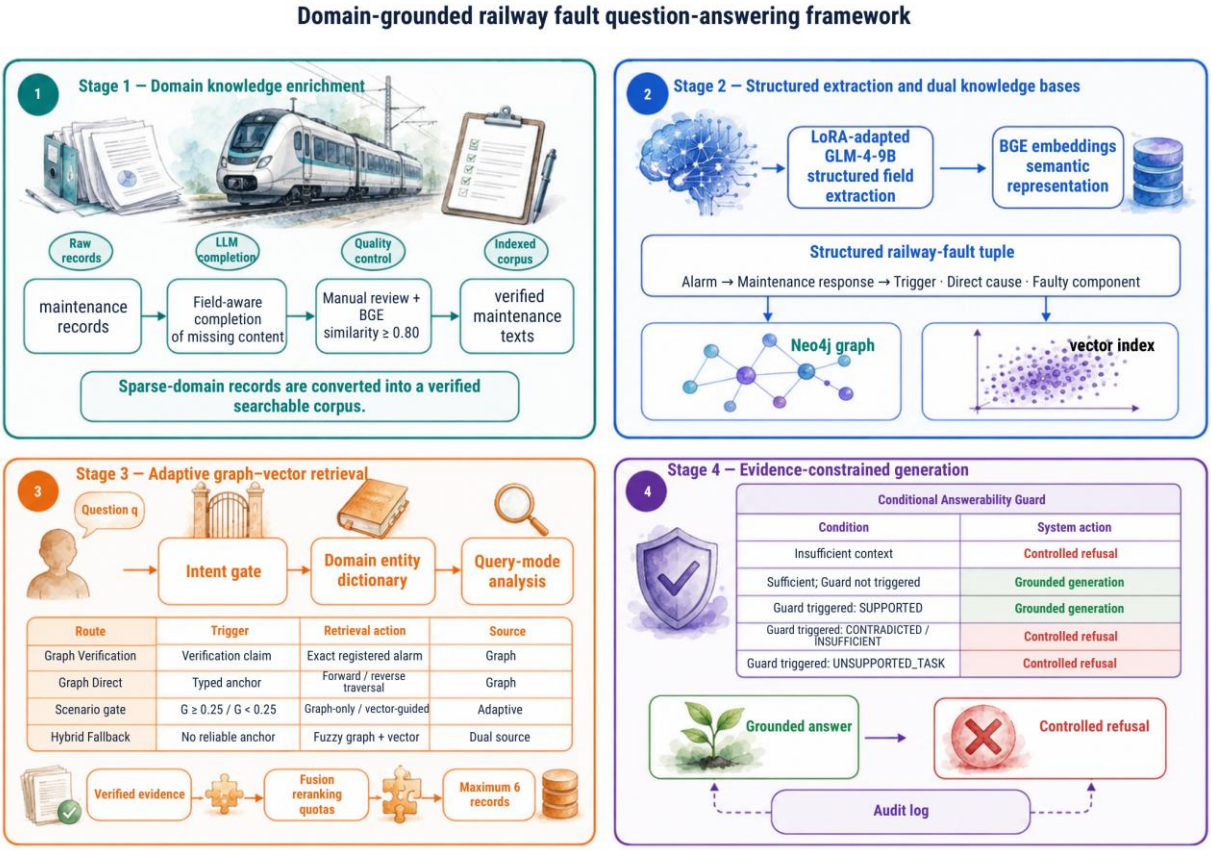


Figure 1. Four-stage framework for railway fault knowledge construction, adaptive retrieval, and evidence-constrained answer generation with conditional answerability auditing.

Figure 2

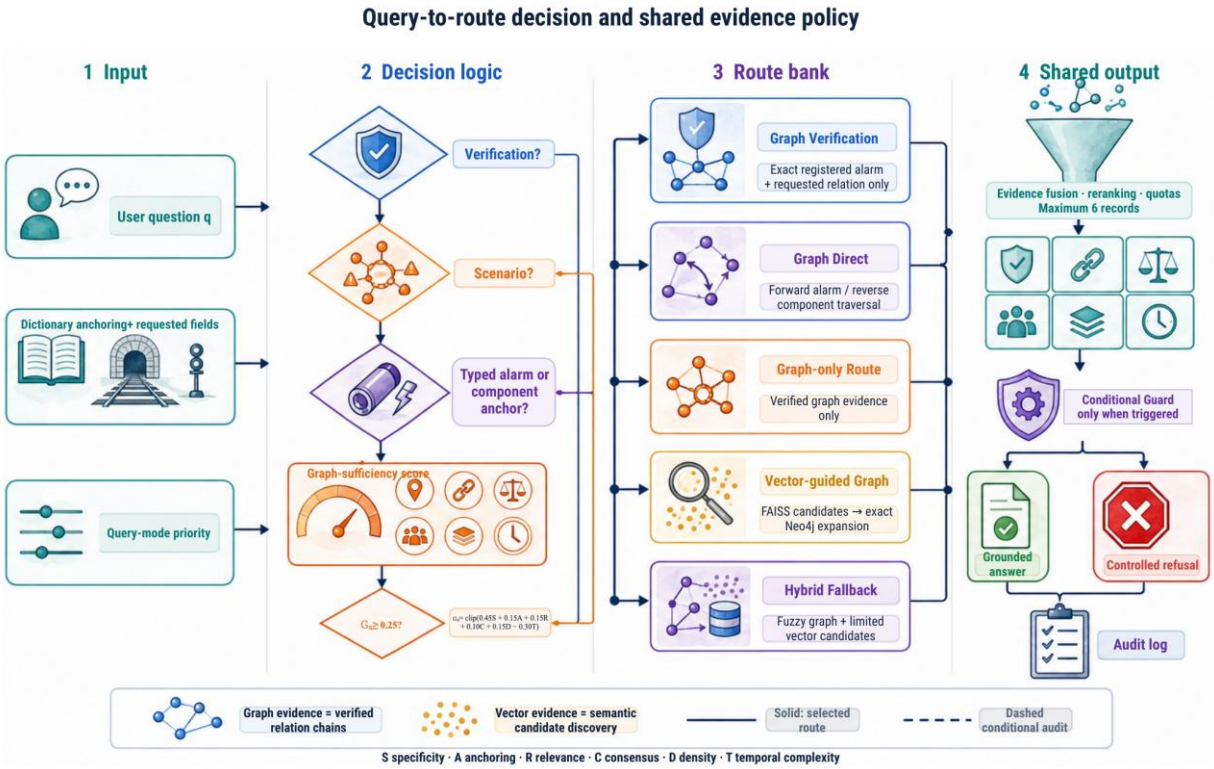


Figure 2. Query-to-route decision logic, shared evidence policy, and conditional Answerability Guard path of the adaptive graph-vector retrieval mechanism. The Verification decision includes strict exact validation of quoted but unregistered alarm terms; Guard invocation uses the separate lexical trigger listed in Section 3.4.2. The requested-relation label denotes answer scope: the implementation serializes exact graph chains and the prompt limits the returned fields.

Figure 3

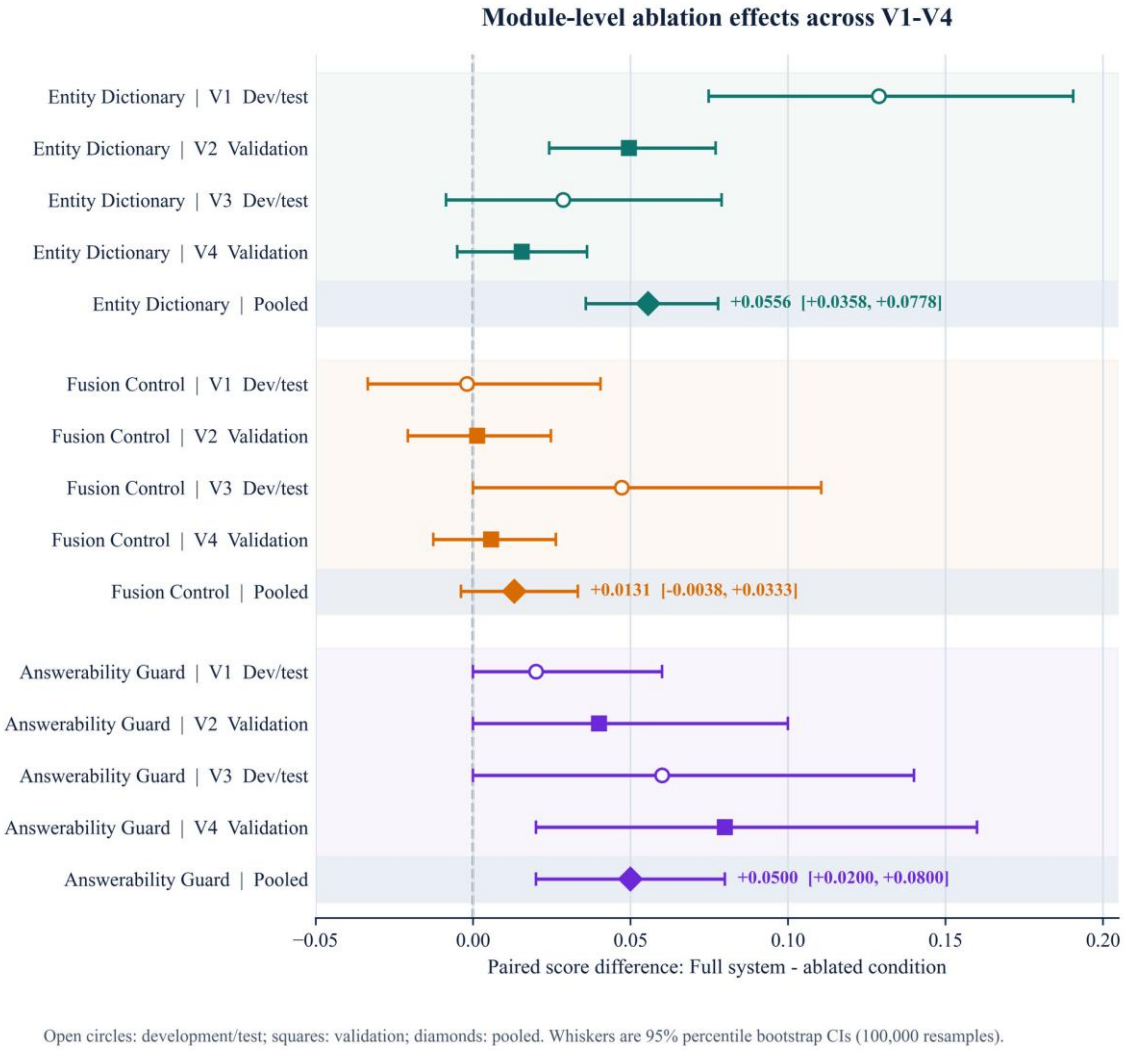


Figure 3. Question-level and pooled effects of the Entity Dictionary, Fusion Control, and Answerability Guard across V1-V4. Whiskers indicate 95% percentile bootstrap intervals.

Intervals use 100,000 paired question-level bootstrap resamples; n = 50 for each set and n = 200 for the pooled comparison. Positive differences favour the Full system.

Figure 4

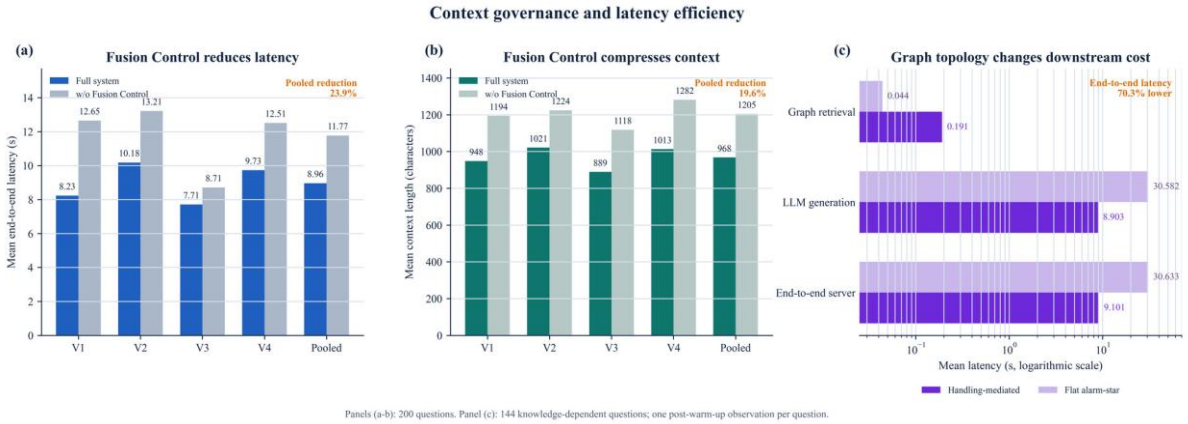


Figure 4. Context-governance and latency effects of Fusion Control and graph topology. Panels (a-b) summarize the 200-question fusion comparison; panel (c) reports 144 knowledge-dependent questions.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryInformationsubmission.pdf