

Trust-Aware KG-RAG: a framework for trustworthy question answering for engineering equipment

Qi Shi^{a,1}, Ye Yang^{a,1}, Jiaqi Di^a, Lin Li^b, Jie Liu^c, Youmin Hu^{a,e}, Jinhua Xiao^d, Zhongxu Hu^{a,*}

^a Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

^b State Grid Electric Power Research Institute, Wuhan Nari Group Corporation, Wuhan 430070, China

^c School of Civil and Hydraulic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

^d Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Milan 20133, Italy

^e Hust-Wuxi Research Institute, Wuxi 214174, China

ARTICLE INFO

Keywords:

Knowledge graphs
Retrieval-Augmented Generation (RAG)
Equipment maintenance
Trustworthy
Question answering

ABSTRACT

Trustworthy Question Answering (Q&A) is essential in safety-critical engineering domains, where answers must strictly align with authoritative documentation for equipment testing and inspection. While Large Language Models (LLMs) exhibit significant potential for automating this process, their direct application is hindered by the unique characteristics of engineering data and knowledge bases, including strict rule constraints, multi-source and cross-format data, and prevalent issues of data redundancy and information conflicts. These challenges lead to factual hallucinations and erroneous inferences, undermining the trustworthiness of existing large language model-based frameworks in this domain. To bridge the trustworthiness gap, this paper proposes a Trust-Aware KG-RAG framework (TAG) that systematically integrates a trust mechanism across the entire pipeline of extraction, graph construction, fusion, retrieval, inference, and question answering. Specifically, the proposed approach employs tailored context-engineering and domain-specific extraction strategies for structured and unstructured documents. During the extraction phase, a Trust-Aware Extraction Agent performs Trust-Weighted Relation Extraction to assign trust weights to entities and relations. In the fusion stage, a Cross-Graph Trust Validation mechanism acts as an arbiter to resolve conflicts among sources. Finally, a Trust-Guided Retrieval & Q&A strategy enhances answer trustworthiness through weight-driven retrieval and evidence voting. Experiments are conducted on an engineering-equipment Q&A benchmark derived from power-transformer testing reports, which provide a representative case study within safety-critical engineering documentation. The proposed framework achieves an F1-score of 0.785, representing a 54.5% improvement over a conventional KG-RAG baseline. These results suggest that the trust-aware pipeline is effective in prioritizing more trustworthy evidence and improving answer alignment with source data in the evaluated setting.

1. Introduction

The development of intelligent, integrated platforms is central to modern engineering systems and large-scale infrastructures [1–3]. In safety-critical engineering domains, it is essential to manage the testing and inspection of key equipment, such as transformers and high-voltage switchgear, to ensure safe and stable operation. Unlike open-domain linguistic resources, engineering documentation follows strict technical and regulatory standards. In safety-critical equipment testing across engineering domains, we refer to the multi-source, cross-format

corpus commonly found in equipment testing across domains, including national/industry standards, manufacturer manuals, inspection/test reports, operational logs. These documents, while domain-specific, exhibit similar characteristics across different engineering fields: they are multi-source, mixed-format, such as structured tables, semi-structured records, and unstructured manuals, and often contain redundancies or inconsistencies due to version updates or differing authority levels. These characteristics complicate automated knowledge extraction and reasoning. For example, in power transformer inspection, engineers typically need to consult national standards such as GB/T 6451, manufacturer manuals, and historical test reports. These materials

* Corresponding author.

E-mail address: Zhongxu.hu@hust.edu.cn (Z. Hu).

¹ The authors contributed equally to this work.

Nomenclature

KG	Knowledge Graph
NER	Named Entity Recognition
LLMs	Large Language Models
RAG	Retrieval-augmented generation
RE	Relation Extraction
KGQA	Knowledge Graph Question Answering
GPT	Generative Pre-trained Transformer

are stored in different formats, including PDF tables, scanned records, and spreadsheets, and may contain inconsistent requirements due to standard updates or differing authority levels. Cross-checking this information manually is time-consuming and prone to error, especially when decisions must be made under strict compliance constraints. This situation reflects the broader challenges faced in engineering equipment testing, where multi-source, mixed-format documents must be interpreted and reconciled before reliable conclusions can be drawn. In current practice, engineers spend considerable time searching for relevant clauses, comparing versions of standards, and verifying test results across multiple sources. This process is not only inefficient but also increases the risk of oversight, particularly in safety-critical contexts where inspection judgments must strictly follow formal criteria. Automated extraction helps ensure that decisions are grounded in up-to-date and authoritative information, thereby supporting consistency, compliance, and traceability. It also provides a structured knowledge basis for downstream tasks such as fault diagnosis, condition assessment, and inspection plan formulation.

In this study, LLMs are primarily used for knowledge extraction from multi-source, mixed-format documents and for generating inspection conclusions and reports under the guidance of retrieved trustworthy evidence. When general-purpose Large Language Models (LLMs) are used in such environments, the outputs often fail to conform to normative requirements, and reasoning may be difficult to verify. This is problematic, particularly for inspection report generation, where high reliability and traceability are crucial. Given the limitations of traditional manual inspection in equipment testing and the challenges of acquiring domain knowledge and ensuring accurate judgments [4–6], LLMs like ChatGPT [7] and DeepSeek [8] have shown potential in natural language understanding and generation [9,10]. For example, in power transformer testing, manual inspection involves tasks such as reviewing insulation test records, checking winding deformation measurements against historical trends, and interpreting oil chromatographic analysis results in light of standard thresholds. These tasks are not only time-consuming but also prone to human error, especially when documents from different sources contain conflicting information. However, their key limitations, such as insufficient domain knowledge and susceptibility to factual errors, are magnified when handling complex reasoning tasks that require specialized expertise [11,12]. For example, determining whether the insulation resistance of a transformer complies with the latest standard is not a simple threshold check. The applicable limit may vary with voltage level, temperature correction factors must be applied according to standard [appendices](#), and historical test records are often reviewed to assess performance trends. Such evaluation requires integrating information scattered across multiple documents and reconciling potential inconsistencies between versions or sources. Current general-purpose LLMs struggle with this type of cross-document and rule-constrained reasoning. Similar requirements arise in other safety-critical domains, such as aircraft maintenance and nuclear equipment inspection, where compliance decisions depend on structured standards and multi-source evidence.

To mitigate this issue, Retrieval-Augmented Generation (RAG) constrains model outputs by introducing external knowledge bases [13], yet

conventional RAG systems [14] face challenges such as low retrieval efficiency, semantic mismatch, and context fragmentation. Knowledge Graphs (KGs), valued for their ability to structurally store and associate domain-specific knowledge [15], have been used to enhance RAG systems. Recent work has increasingly focused on KG-augmented RAG, a paradigm often referred to as KG-RAG, which leverages the structural information of KGs to guide retrieval and reasoning, proving particularly valuable for managing long-tail knowledge [16].

Although the KG-RAG paradigm is a useful baseline, its adaptability to engineering-equipment testing scenarios is constrained by the single-domain yet multi-source characteristics of engineering test documents [17]. First, the content is governed by strict logical constraints and often exhibits cross-document inconsistencies: standards, manuals, and test reports contain domain-specific terminology and complex dependencies, while contradictory or erroneous statements can appear across sources [18–20]. Second, the engineering-equipment corpus consists of multi-source and cross-format data, ranging from structured tables to semi-structured records and unstructured manuals, yet current frameworks lack mechanisms for reliable and format-aware processing of this heterogeneity [12]. Third, the lack of explicit inter-document relations and precedence, such as standard versions and authority levels, leads to unguided and inefficient retrieval, increasing variability in generated results. Current KG-RAG pipelines rarely include built-in mechanisms to assess source trust, resolve conflicts, and fuse such multi-source engineering evidence end-to-end, leaving a “trustworthiness gap” that the proposed trust-aware framework aims to address.

To address the aforementioned challenges, this study proposes a Trust-Aware KG-RAG framework (TAG) for safety-critical engineering equipment. Although transformer inspection is used as a representative case in our experiments, TAG is intended for safety-critical engineering equipment settings and may also be relevant to other scenarios with similar documentation characteristics. Positioned at the intersection of KG-RAG, Generative AI/LLMs, and Engineering Equipment Maintenance, as conceptualized in the Venn diagram in [Fig. 1](#), this framework provides targeted solutions to the three major problems by introducing a systematic trust mechanism:

1. To handle content complexity and conflicts, a Trust-Aware Extraction Agent is introduced. In addition to domain-aware extraction, trust weights are assigned to entities and relations during extraction, so that the reliability of each knowledge item is quantified at its origin.
2. To resolve multi-source and cross-format data, independent heterogeneous knowledge extraction modules and a Cross-Graph Trust Validation mechanism are proposed. The former performs differentiated processing on structured and unstructured data, while trust weights and provenance metadata arbitrate conflicts during fusion and support the construction of a logically consistent global knowledge graph.
3. To compensate for the lack of relational and priority logic, a Trust-Guided Retrieval & Q&A method is developed that leverages the established trust weights within the graph to guide retrieval, prioritizing the exploration of high-trust knowledge paths, and employs an evidence-based voting mechanism to replace random selection, improving the reliability and evidence grounding of the final answer.

The arrangement of the paper is as follows: [Section 2](#) reviews related work; [Section 3](#) presents the design and implementation of the proposed framework and analyzes its applicability to the target scenarios; [Section 4](#) reports experimental evaluation and analysis; [Section 5](#) concludes the paper and discusses future research directions.

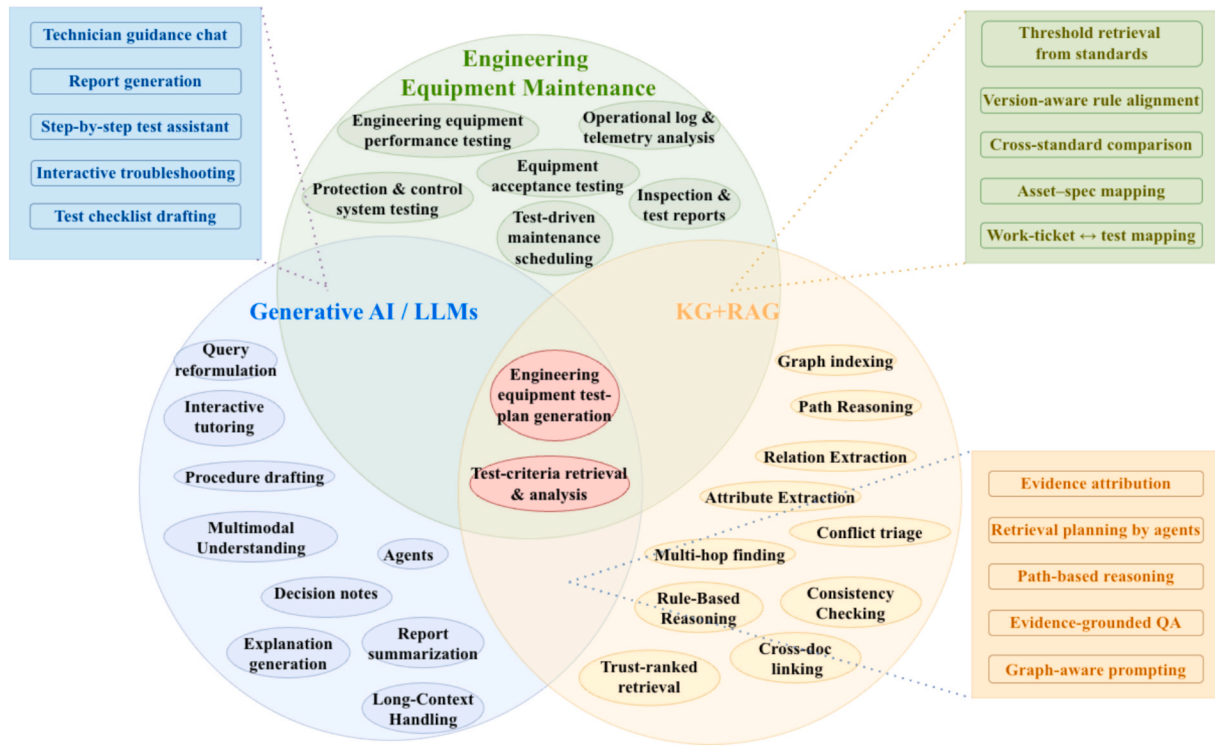


Fig. 1. Venn diagram of the proposed research focus at the intersection of Generative AI/LLMs, KG-RAG, and engineering equipment maintenance.

2. Related work

2.1. Evolution of retrieval-augmented generation: from rules to knowledge graphs

Retrieval augmented generation (RAG) has moved from rule-based matching to embedding driven semantic retrieval, and is now shifting toward more explicit forms of grounding. Most RAG systems follow a simple pattern: they retrieve supporting evidence from an external knowledge base and feed it into a language model, which often improves performance on knowledge intensive tasks [21–23]. In text-only RAG, candidate passages are embedded and ranked by semantic similarity [24]. In standards-constrained domains, however, similarity is an unreliable proxy for equivalence. Here, we refer to engineering contexts in which equipment testing and certification must adhere to formal technical specifications and codified evaluation criteria. Representative examples include power equipment testing (e.g., GB/T 6451, IEC 60076), materials testing under ASTM standards, and mechanical system certification governed by ISO standards. In such settings, inspection decisions are determined by specific clauses and procedural requirements rather than general semantic similarity. A top-ranked chunk can still omit the clause that governs the decision. To reduce this gap, several studies introduce structure. Melnyk et al. [25] propose multi-stage pipelines that build knowledge graphs (KGs) and extract triples efficiently. Shi et al. [26] introduce REPLUG, a model agnostic method that prepends retrieved documents to a frozen black box model. Even with these advances, it remains difficult to distinguish answers grounded in retrieved evidence from answers driven by parametric model knowledge, which becomes a serious concern when traceability is required. Taken together, this line of work points toward structure-aware augmentation and motivates KG-based RAG.

Recent work explores both KG-based enhancements to RAG [27] and large language model-assisted KG construction [28,39]. Knowledge graph augmented RAG is now common in research and analytic settings [29–33]. Traditional KG construction from text relies on a mix of rule matching, statistical patterns, clustering, and embedding methods

[34–36]. In industrial applications, intelligent maintenance often requires fine-grained fault diagnosis graphs, and LLM driven augmentation can improve nested entity extraction when annotated data are scarce [35]. Baek et al. [37] incorporate subgraphs and structural attributes into prompts as factual references. He et al. [38] propose interactive graph interfaces enriched with textual attributes. Chen et al. [40] develop a distant supervision pipeline, PCNN ATTRA RL, that combines sentence level attention with reinforcement learning to reduce noise, while still depending on a predefined ontology. Shen et al. [41] model dynamic knowledge and bidirectional fusion across processes, resources, and features via a knowledge graph facet, though scalability can suffer on highly heterogeneous corpora. Agentic and iterative graph LLM systems, such as GraphAgent and Graphusion [30–32], enable multi-turn exploration and verification, but they also introduce additional system complexity and prompt engineering effort.

Most KG-based RAG pipelines prioritize representation and retrieval efficiency, while domain level reliability is often treated as an assumption rather than an explicit design target. This has encouraged work that adapts graph retrieval to provide stronger grounding. GraphRAG is a representative example: it leverages document structure and inter-entity links to support multi-hop reasoning and clause-level evidence tracking. The next section reviews these graph-based retrieval paradigms.

2.2. Graph-based retrieval paradigms and structural optimization

Within the KG-RAG family, graph-based variants refine indexing and traversal strategies for different settings, yet standards constrained and safety-critical applications remain relatively unexplored. One prominent direction is end-to-end graph retrieval, with Microsoft GraphRAG [34,42] as a representative implementation. GraphRAG builds document element graphs and retrieves along explicit structure; compared with vector only RAG, it often enables faster multi-hop reasoning and yields more interpretable evidence selection in retrieve than generate pipelines [43,44]. This makes graph-based augmentation appealing for large industrial corpora and motivates further KG-RAG instantiations.

Subsequent work concentrates on structure and efficiency. LightRAG

[45] proposes a graph aware index with a two level retrieval strategy that combines entity level and topic level search, and it supports incremental updates. Fast GraphRAG [46] uses a lightweight graph representation and traversal optimizations to reduce redundant hops and long prompt overhead, seeking a practical balance between speed and accuracy. Wang [47] builds a document element graph in which paragraphs and tables serve as nodes, and lexical, semantic, and structural relations form edges; an LLM based agent then traverses the graph to retrieve evidence. RAFT [48] improves robustness through fine tuning with distractor documents and chain of thought, helping the model separate relevant evidence from noise.

Practical limits remain in generalization, efficiency, and reliability. Liu et al. [49] improve traversal based retrieval but assume preconstructed graphs, which restricts adaptability. Knollmeyer et al. [50] propose hybrid retrieval strategies, IKG and DKG, that expand cross-document coverage while trading off some global estimation accuracy. Bahr et al. [51] and Heredia Alvaro et al. [52] show that domain specific indexing and pipeline customization can help, yet many systems depend on highly structured inputs and degrade when data quality varies. FastRAG [53] reduces reliance on LLM calls via block sampling and improves efficiency, but answer veracity remains challenging in safety-critical tasks.

Even as efficiency and scalability improve, most KG-RAG systems treat heterogeneous sources as equally reliable. In engineering inspection, that authority agnostic stance is risky and motivates the trust-aware extension developed in this paper. Parallel work has also explored domain specific question answering frameworks that combine KG-RAG pipelines with large language models, as discussed next.

2.3. Augmenting LLMs with knowledge graphs for Q&A: pipeline and design choices

Beyond retrieval speedups, a growing body of work integrates KG-RAG with large language models to support domain specific question answering and reasoning. Typical end-to-end pipelines include graph acquisition, indexing, retrieval orchestration, evidence integration into the model context, and some form of reliability governance.

Safety-critical engineering domains add pressure on both KG construction and LLM use: documentation is fragmented across sources, standards evolve over time, and annotations are often scarce [54]. Existing domain adaptations in RAG and KG-RAG reflect a range of design choices. Wu [55] introduced U Retrieval for medical scenarios, combining top down precise retrieval with bottom up response refinement. Naganawa [56] proposed a RAG framework for manufacturing execution systems, addressing sensor data, complex workflows, and domain knowledge integration. Yuan [57] presented a telecommunications solution in which a knowledge graph captures relationships among network protocols, standards, and entities, improving domain Q&A accuracy. MGRAG [58] integrates text, images, and topology graphs for unified retrieval and reasoning on multimodal work ticket datasets; its effectiveness depends on the completeness and freshness of historical records, and multimodal indexing plus subgraph matching can increase computational cost in large scale grids. Bernier et al. [59] introduced SafePowerGraph LLM, which combines grid graph structure with tabular representations to support modeling and reasoning for optimal power flow. Grid Codex [60] improves code interpretation through multi-stage query optimization and RAPTOR based retrieval augmentation, but its reliance on large scale models and complex pipelines can lead to high inference latency and unstable behavior on long text or cross-language inputs.

Engineering equipment documentation brings its own difficulties. Knowledge is scattered across multiple systems and documents, and many queries require combining theoretical guidance with field records. Redundancy is routine, and operational data can still contain inaccuracies [61]. Retrieval can therefore surface incomplete or conflicting evidence. The framework proposed in this paper targets this setting by

assigning and propagating trust over entities, relations, and clauses, validating conflicts across graphs, and combining trust-guided retrieval with post generation credibility checks to support auditable, domain compliant Q&A.

Knowledge Graph Question Answering (KGQA) has been widely studied for querying structured knowledge bases, where natural language questions are translated into formal queries executed over a predefined KG [62–64]. Such systems are effective for fact-oriented queries on well-curated and schema-stable datasets. However, they typically assume a fully structured, query-ready graph and a relatively fixed ontology. Recent work has shown that KGQA models struggle when faced with unseen schema items or novel relation compositions, highlighting limitations in generalization [65,66]. In safety-critical engineering inspection, documentation is heterogeneous, partially structured, and evolves across standards and field records. Constructing and continuously maintaining a complete KG suitable for classical KGQA is therefore difficult. Moreover, inspection tasks often require explanatory or report-style outputs and reconciliation of multi-source evidence, rather than retrieval of isolated entities. For these reasons, a KG-RAG architecture that combines graph-structured guidance with generative reasoning is better aligned with the requirements of standards-driven and multi-source engineering environments.

2.4. Summary

KG-RAG has developed quickly, with progress in retrieval efficiency, robustness, and domain adaptation. Many systems improve performance through better graph construction, retrieval refinement, and multi-hop reasoning. These advances have expanded KG-RAG beyond open domain settings and into engineering and industrial applications.

A persistent weakness remains: most work focuses on how to retrieve and reason more effectively, while paying less attention to whether the retrieved knowledge is reliable. Many frameworks implicitly assume uniform trust across heterogeneous sources and do not explicitly weigh or reconcile evidence by authority, temporal validity, or version consistency.

This assumption may be acceptable in general purpose settings, but it is problematic in standards driven and safety-critical tasks such as equipment testing and inspection. Engineering documentation is domain specific and multi-source, with authority hierarchies, version precedence, and conditional applicability. Inconsistencies and uneven data quality coexist with authoritative clauses. A system that cannot distinguish a governing standard from an informal note or a noisy field record can produce conclusions that are hard to justify and potentially unsafe.

To our knowledge, existing KG-RAG frameworks rarely treat trust-worthiness as a first class design goal across the full pipeline, including knowledge extraction, graph fusion, retrieval, reasoning, and final answer generation. Without end-to-end trust governance, a gap remains between efficiency oriented retrieval and reliability oriented decision support. Closing that gap is necessary for credible and auditable Q&A in safety-critical engineering equipment, and it also supports the broader push toward explainable and verifiable industrial reasoning systems.

3. Methodology

This study proposes the TAG framework instantiated for engineering equipment testing. While demonstrated using power transformer monitoring as a case study, the framework is designed to be domain-agnostic and applicable across various engineering domains. In this paper, engineering equipment testing refers to standards-driven compliance verification activities in which equipment performance is assessed through documented procedures, such as electrical, thermal, and mechanical qualification tests. It is an end-to-end pipeline that keeps an explicit, continuous trust score alongside the data. The pipeline turns raw, multi-source test records—such as inspection reports, test

data sheets, operational logs, and sensor readings—into knowledge artifacts that can be audited and traced. This knowledge serves two complementary purposes. It supports evidence-grounded adjudication of existing tests by enabling traceable verification of inspection results against relevant standards. It assists in preparing compliant test plans for subsequent inspection cycles or newly deployed equipment by organizing thresholds, measurement procedures, and decision rules derived from authoritative standards and historical inspection records. While

test records are generated through past test executions, the structured knowledge extracted from them can serve as a reference to support future inspection planning under applicable standards and operational requirements. The approach retains the standard KG-RAG interface and a comparable computational footprint. It is anchored in the engineering domain through metadata on provenance, source authority, document version, and test conditions, which together shape trust score estimation and guide downstream choices.

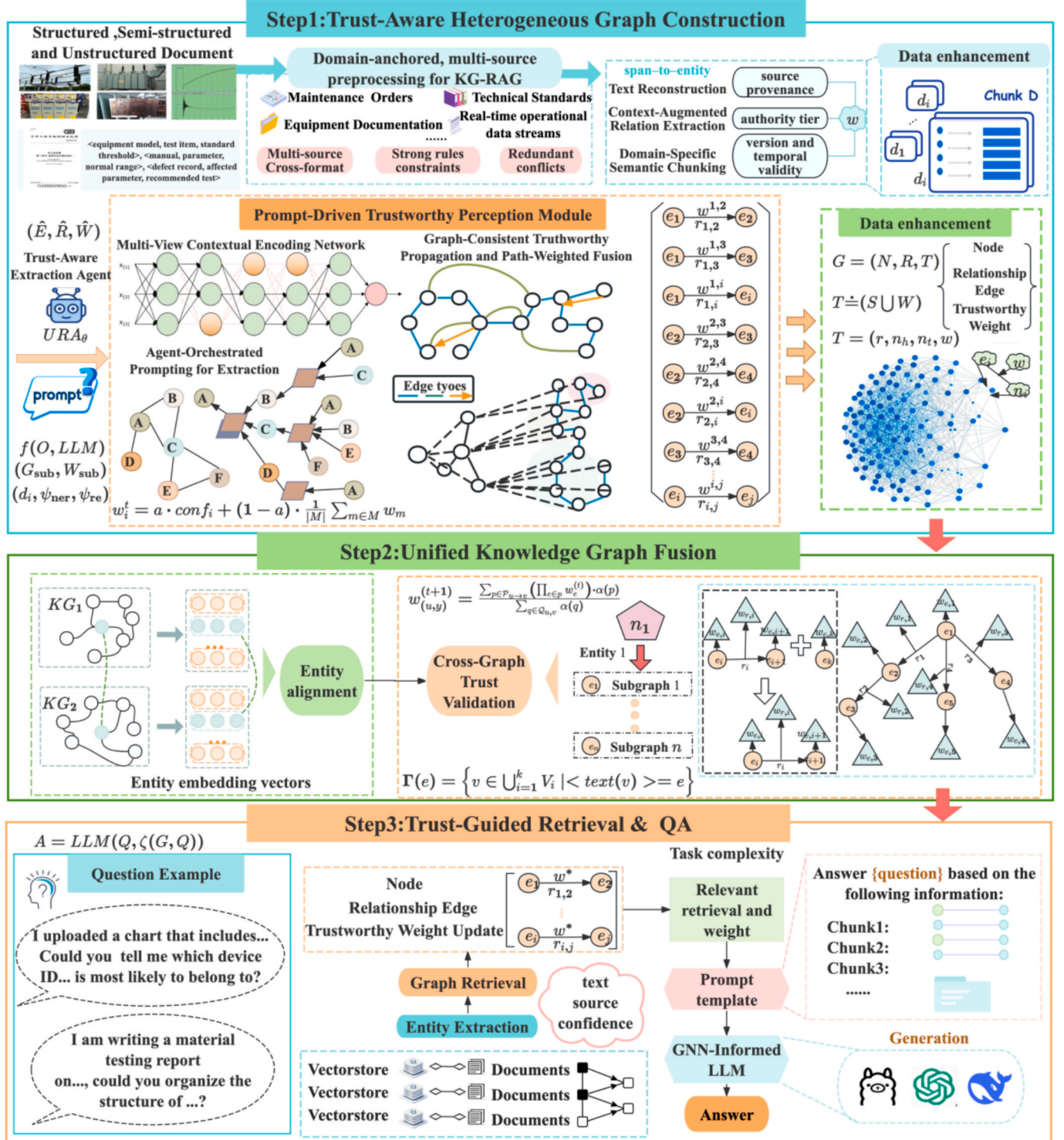


Fig. 2. The framework comprises three tightly integrated modules: trust-aware heterogeneous graph construction, unified graph fusion, and Trust-Guided Retrieval and question answering. Together, they implement a KG-RAG workflow that enables trustworthy reasoning from preprocessing to final Q&A.

The methodology contains three closely connected modules, as summarized in Fig. 2. Trust-aware heterogeneous graph construction integrates domain-anchored context engineering and trust-aware extraction. It standardizes both structured, semi-structured and unstructured sources, correction of condition descriptors, and clause-preserving text chunking. The framework establishes separate processing channels for structured, semi-structured and unstructured documents. The trust-aware extraction mechanism integrates agent-enhanced relational and entity extraction to generate heterogeneous graphs. Unified graph fusion merges the resulting source-specific subgraphs. Cross-graph trust validation identifies conflicts, resolves them using authority and revision history, and refines trust scores through iterative propagation across the graph. Trust-guided retrieval and question answering applies the trust score during inference by retrieving evidence with trust weights, aggregating support via weighted voting, and verifying clause-level credibility before generating the final answer. Overall, the modular pipeline manages equipment test data at scale without a noticeable rise in computational complexity, addressing the trust gap often seen in conventional KG-RAG systems.

3.1. Trust-aware heterogeneous graph construction

3.1.1. Domain-anchored context engineering

In this study, heterogeneous data sources related to engineering equipment are first standardized and organized as preparation for subsequent graph construction. The corpus is single-domain yet multi-source and cross-format, spanning national/industry standards, manufacturer manuals, inspection/test reports, operations logs, and real-time data such as SCADA (Supervisory Control and Data Acquisition) and PMU (Phasor Measurement Unit) records. This design demonstrates the framework's capability to accommodate various data modalities, reflecting its extensibility to different application scenarios.

The data can be broadly categorized into structured, semi-structured, and unstructured forms. Structured data typically includes maintenance records and fault monitoring logs, where entities and parameter values are explicitly recorded in tabular form. For example, a test record may contain a table with winding resistance measurements at a specific oil temperature (e.g., 26.0°C), with values such as $R_{AB} = 0.9927 \Omega$; a fault monitoring log might record a timestamp, equipment ID, parameter name, value, and unit (e.g., "2025-06-27 14:30, Transformer T1, top oil temperature, 73.3°C"). Semi-structured data, such as formatted inspection sheets or tagged maintenance reports, contain identifiable field labels and recurring templates but do not follow a fully relational schema. Unstructured data, such as technical manuals and experimental reports, embed relevant knowledge within continuous text, where relationships between entities are often implicit.

Because these three types of data originate from different systems and documentation practices, their reliability, completeness, and contextual clarity vary significantly. Therefore, rather than processing all sources in a uniform manner, they are first prepared separately according to their format characteristics.

Structured sources are standardized through schema-guided transformation to convert tables and forms into consistent entity and parameter representations. Data cleaning addresses missing values and obvious anomalies, and unit normalization is applied where necessary to ensure comparability across records. Semi-structured sources are parsed using template-based field recognition to extract key-value pairs and logical sections, which are then aligned with the same entity and parameter definitions used for structured data. Unstructured sources undergo document parsing and clause-preserving segmentation to produce text units that retain contextual boundaries and source information. These prepared textual units are structured to support subsequent extraction (performed in Section 3.1.2), such as < equipment model, test item, standard threshold>, < manual, parameter, normal range>, and < defect record, affected parameter, recommended test>.

Through these preprocessing steps, structured, semi-structured, and

unstructured data are transformed into standardized, source-aware representations while preserving provenance and contextual information. The resulting representations serve as inputs for heterogeneous graph construction and semantic extraction in the subsequent module.

3.1.2. Trust-aware extraction via intelligent agent

Building upon the standardized and source-aware representations prepared in Section 3.1.1, we now construct a domain-specific heterogeneous graph and perform semantic extraction. At this stage, structured records and clause-level text units are transformed into entity-relation instances, and trust weights are introduced to support reliability-aware reasoning.

In the context of trust-weighted dimensions, a domain-specific heterogeneous graph is constructed using a knowledge graph to represent the input [51]. The heterogeneous graph G for engineering equipment data is formulated as Equation (1):

$$G = (\mathbf{N}, \mathbf{R}, \mathbf{T}) \quad (1)$$

where $\mathbf{N}$ denotes the set of all entity nodes, and $\mathbf{R}$ represents the set of all relation edges. The term $\mathbf{T}$ refers to element types, as defined in Equation (2):

$$\begin{aligned} \mathbf{T} &= (n_h, r, n_t, w) \\ \mathbf{T} &\triangleq \mathbf{S} \cup \mathbf{U} \end{aligned} \quad (2)$$

where n_h is the head node, r is the relationship, n_t is the tail node, and w is the relationship weight. $\mathbf{S}$ and $\mathbf{U}$ are the heterogeneous types of nodes and edges, respectively. In this heterogeneous graph, for each edge, a *meta*-type attribute is assigned, which includes the *meta*-types of the head node n_h , relationship r , and tail node n_t . Unlike vanilla KGs in natural language text, each entity node in this graph is assigned a trust weight w , which facilitates the subsequent trust mechanism tailored to the characteristics of engineering equipment data. This trust weight is obtained simultaneously with the entity nodes during the entity-relation extraction phase. During this extraction process, clause-level text units prepared in Section 3.1.1 are aligned to graph entities through span-to-entity mappings. Each chunk retains source provenance, authority tier, version information, and condition context. These node-to-chunk alignments connect the heterogeneous graph with the clause-anchored text representation, ensuring that graph nodes are traceable to their original document evidence and providing auditable inputs for subsequent cross-source fusion and trust-aware retrieval. The process is detailed in the following section.

The proposed architecture introduces the necessary intelligent agent only at the knowledge graph stage to implement the trust mechanism, without incorporating multi-task agents at other stages. It is motivated by the fact that the Vanilla GraphRAG operates effectively without agents, and the practical consideration that introducing unnecessary agents would add extra overhead and delay.

A Trust-Aware Extraction Agent is introduced to construct the aforementioned heterogeneous graph, aiming to optimize the entity relationship information and the semantic dependencies of data in the engineering systems knowledge graph. The process of generating the heterogeneous graph using the Trust-Aware Extraction Agent can be defined as $Y = f(O; LLM)$ [30], where the agent function $f(\cdot)$ combines observations O , including explicit graph connections (structured data) and unstructured data (such as textual information). The agent then generates a heterogeneous graph generation action Y , with the agent URA_θ input and output in Equation (3):

$$(\hat{E}, \hat{R}, \hat{W}) = URA_\theta(d_i, G_{sub}, W_{sub}; \psi_{ner}, \psi_{re}) \quad (3)$$

where $\hat{E}, \hat{R}$ is the high-trust set of entities and relation triples, $\hat{W}$ represents the trust weight set $W = \{w_e, w_r\}$. URA is the Trust-Aware Extraction Agent. It corresponds to the trainable parameters θ . d_i is the text block. G_{sub} is the current knowledge graph subgraph. W_{sub} is the subgraph trust weight set. ψ_{ner}, ψ_{re} is the entity/relation extraction

model. w_e is the entity set weight. w_r is the relation triple weight. Unlike methods that rely on carefully handcrafted prompt templates, the TAG framework adopts a trust-aware dynamic prompt construction strategy.

In this process, entity extraction is first performed based on the intelligent agent, as defined in Equation (4):

$$(E_{raw}, w_{raw}) = \psi_{ner}(d_i; p_t) = \{(e_i, type_i, conf_i) | i, \dots, n\}$$

$$w_{e_i} = a \cdot conf_i + (1 - a) \cdot \frac{1}{|M|} \sum_{m \in M} w_m \quad (4)$$

where E_{raw} represents the extracted base entity set, w_{raw} represents the extracted base weight, p_t is the prompt vector, e_i refers to the i -th entity, $type_i$ is the entity correspondence type, $conf_i$ represents the trust of the Named Entity Recognition (NER) model, w_{e_i} is the entity trustworthiness, M is the set of other entities in the same sentence, a represents the entity trustworthiness balancing coefficient. The process also requires an optimized prompt. To design the Prompt-driven trust-awareness module, the Trust-Aware Extraction Agent dynamically generates prompts with trust perception. Specifically, the input question, retrieved textual evidence, and textualized subgraph context are combined and encoded through a lightweight attention module to form the LLM input representation. This design is intended to reduce reliance on manually optimized prompt wording by tying the LLM input more closely to trust-weighted evidence selection. When extracting a standard threshold, the prompt vector is enriched with contextual metadata like test date and temperature, guiding the agent to perform context-sensitive inference that respects the dynamic and conditional nature of equipment standards.

This ensures that entities and relationships with high trust receive higher weight in the prompt generation. Specifically, the textual description of the text d_i and subgraph G_{sub} is concatenated and passed through a lightweight attention layer to generate the prompt vector. Finally, select a subset $w_{e_i} > \chi$ from E_{raw} to form the high-trust entity set $\hat{E}$, where the trust threshold χ determines which entities are included.

In the relation extraction Equation (5):

$$\hat{R} = \{(e_p, r_{pq}, e_q) | conf_{base} > \eta\}$$

$$conf_{base} = \psi_{re}(d_i, e_p, e_q)$$

$$w_r = \beta \cdot conf_{base} + (1 - \beta) \cdot \sqrt{w_{e_p} \cdot w_{e_q}} \quad (5)$$

where $\hat{R}$ is the relation-extraction model, $conf_{base}$ is the base relation trust, η is the trust threshold, $w_{e_p} \cdot w_{e_q}$ is the trust weights of the two endpoint entities in the relation, β is the coefficient for base-trust weighting, and w_r is the relation-level trust score. During heterogeneous graph construction, update each heterogeneous subgraph according to the subgraph-update rule Equation (6):

$$G'_{sub} = G_{sub} \cup \{\hat{E}, \hat{R}\} \quad (6)$$

After each subgraph update, re-search existing relations and conduct relation validation. Apply the following weight-update rule Equation (7):

$$w_e^{new} = \gamma \cdot w_e^{old} + (1 - \gamma) \cdot w_e^{current}$$

$$w_r^{new} = \delta \cdot w_r^{old} + (1 - \delta) \cdot w_r^{current} \quad (7)$$

In this rule, superscripts on w_e^{new} indicate the state before or after each iteration, subscripts on w_e^{new} index entities or relations, γ is the entity-weight update parameter, and δ is the relation-weight update parameter. This model preserves computational efficiency. The trust weight mechanism substantially improves the quality of the resulting knowledge graph.

3.2. Unified knowledge graph fusion

3.2.1. Fusion through cross-graph trust validation

After generating the heterogeneous graphs, heterogeneous graph fusion is performed to obtain the final global knowledge graph. Conventional pipelines typically apply direct entity resolution via alignment-based strategies to produce the final graph. In engineering systems, faulty records in heterogeneous graphs introduce noise, which is often concentrated in structured data. Conflicts typically arise across subgraphs constructed from different document sources. While aligned entities are merged into unified global nodes during entity alignment, conflicting relations associated with those entities are preserved as parallel edges with calibrated trust weights. Trust-aware extraction assigns initial trust priors within each subgraph, and cross-graph trust validation subsequently recalibrates these weights during fusion. In this process, entity fusion ensures structural consistency, whereas relation-level conflicts are resolved through weight propagation rather than deletion.

To address this issue, denoising is first applied to each heterogeneous subgraph to remove spurious or inaccurate entities introduced by faulty records. To identify those entities, we adopt a reliability assessment mechanism based on trust weight and node degree. The reliability of an entity is estimated by aggregating the trust weights of its associated relations. Entities whose connected edges consistently exhibit low trust weights are considered weakly supported. Structural support is measured by its degree, defined as the number of other entities directly connected to it. Degree reflects the local structural importance of a node within the graph [67]. Degree-based structural evaluation is widely adopted in network analysis to identify weakly connected or potentially noisy nodes [68,69]. In sparse knowledge graphs, most entity pairs are not directly connected. Therefore, spurious entities tend to appear as low-degree isolated nodes, which justifies degree-based pruning.

An entity is pruned only when both its aggregated trust weight and its degree fall below predefined thresholds. It is worth noting that entity extraction from heterogeneous sources may occasionally generate multiple nodes referring to the same real-world entity. This scenario is addressed during the graph fusion stage through entity alignment based on text normalization and predefined linking thresholds. When two nodes are determined to represent the same entity, they are merged into a single global entity node. The fused entity may retain multiple aliases for traceability, and its associated trust weights are iteratively updated according to the cross-graph validation mechanism described in Equation (10). This process ensures semantic consistency, reduces redundancy, and preserves multi-source evidence in the final global knowledge graph.

This dual-criteria strategy avoids removing valid but infrequently connected entities in sparse regions of the graph, while effectively suppressing isolated low-trust nodes. Entities with low trust weights but relatively higher degree are retained to preserve graph completeness; however, due to the weight-aware retrieval mechanism, their influence during downstream reasoning remains limited. This design balances robustness and noise suppression.

As illustrated in Fig. 3, during fusion, Cross-Graph Trust Validation is performed on entities and relations using trust weights, and the trust weights are updated iteratively. This process suppresses noise in the input subgraphs and identifies and removes inaccurate nodes and links, thereby improving the trustworthiness of the global knowledge graph.

First, the target entity and relationship set at this stage, as shown in Equation (8):

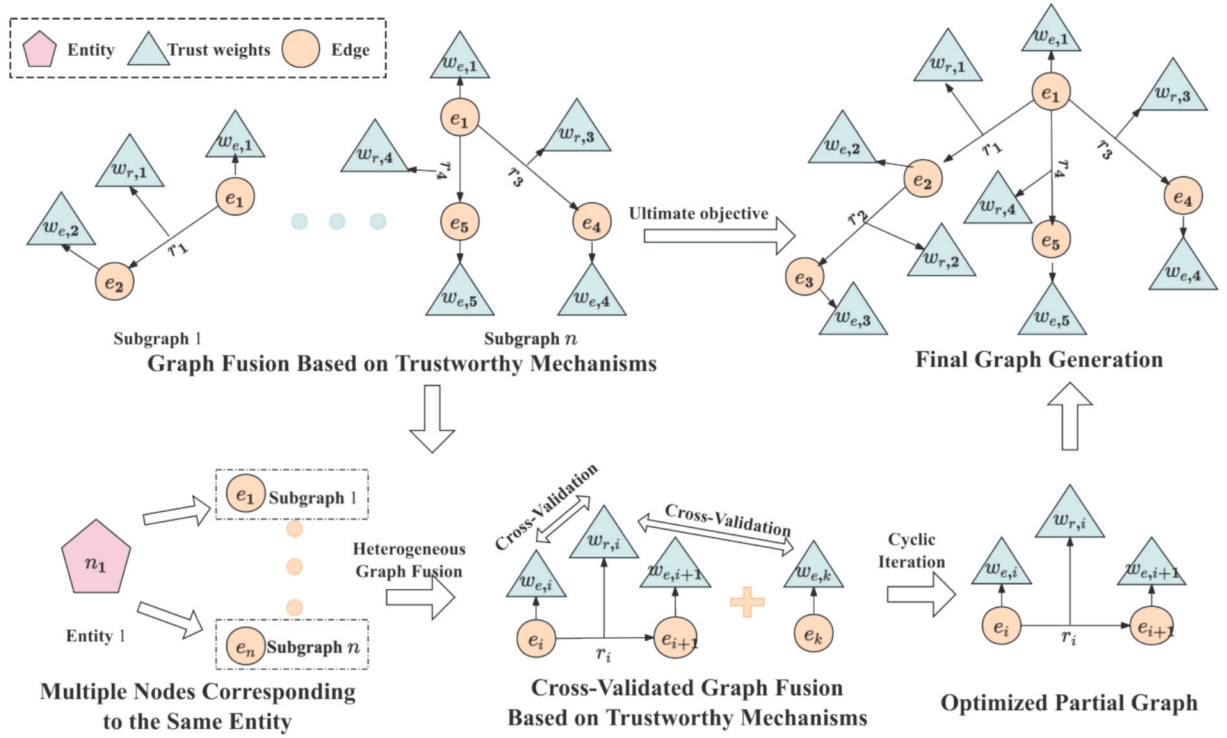


Fig. 3. Schematic diagram of entity-node-edge-weight mapping. During heterogeneous graph fusion, cross-validation across different subgraphs is performed based on trust weights, which are iteratively updated throughout the fusion process.

$$V_{fused} = \left(\bigcup_{i=1}^k \{s_i || v | v \in V_i\} \right) \cup \{h_e | e \in E, |\Gamma(e)| \geq \kappa\}$$

$$E_{fused} = \left(\bigcup_{i=1}^k \{s_i || u, s_i || v | (u, v) \in \hat{R}_i\} \right) \cup \{(v, h_e) | e \in E, v \in \Gamma(e)\}$$
(8)

where V_{fused} denotes the fused global node set, k denotes the number of subgraphs, s_i denotes the data source identifier, $||$ denotes the node ID concatenator, v denotes a node in an original subgraph, V_i denotes the node set of a subgraph, h_e denotes the hub node of entity e , E denotes the set of unique entities, $\Gamma(e)$ denotes the set of nodes containing the entity, κ denotes the linking threshold, E_{fused} denotes the fused global edge set, $\hat{R}_i$ denotes the original edge set of a subgraph G_i , and (u, v) denotes the set of node pairs in the original subgraph.

The proposed fusion mechanism supports incremental integration, where newly acquired datasets are incorporated as subgraphs without reconstructing previously fused components.

3.2.2. Entity alignment and trust weight propagation

A distinction is made between nodes and entities because a single entity may correspond to multiple nodes before fusion. Entity and relation alignment are therefore required. The entity to node mapping is defined as:

$$\Gamma(e) = \{v \in \bigcup_{i=1}^k V_i | \langle \text{text}(v) \rangle = e\}$$
(9)

where $\text{text}(\cdot)$ denotes the node text normalization function for this mapping, and V_i denotes the union of nodes across all subgraphs. The above establishes a mapping from entities, represented by text, to nodes in the knowledge graph.

During fusion, trust weight updates are defined as Equation (10):

$$w_{(u,v)}^{(t+1)} = \frac{\sum_{p \in P_{u \rightarrow v}} \prod_{e \in p} w_e^{(t)} \cdot \alpha(p)}{\sum_{q \in Q_{u,v}} \alpha(q)}$$
(10)

In the trust weight update process, $w_{(u,v)}^{(t+1)}$ denotes the trust weight of edge u, v in the t -th iteration, P represents the set of all paths from entity u to entity v , p is a single path, $w_e^{(t)}$ is the current weight of edge e in the path, α is the path decay factor, and P is the set of all candidate paths.

Introducing an iterative mechanism for trust weight updates helps address semantic consistency during graph fusion. Fusion across heterogeneous subgraphs may introduce noise, such as homonymous or ambiguous entities. The iterative updates mitigate this noise by adjusting weights using feedback from neighboring nodes.

3.3. Trust-guided retrieval & Q&A

After trust weight information is generated during graph construction, an optimized retrieval strategy for question answering is designed on this foundation. This enhancement increases the proportion of higher trust content in the retrieved results while maintaining recall.

First, given a natural language question Q and the knowledge graph G constructed as described above, the goal of TAG is to provide a guide ζ that enables the extraction of relevant entities, relations, and logical paths from the graph G . Next, the guide text is fed into a graph neural network-enhanced LLM to generate the final answer A . The process can be modeled as Equation (11):

$$A = LLM(Q, \zeta(G, Q))$$
(11)

Based on the above problem model and the specialized graph provided in the previous section, a trust-assisted retrieval strategy for question answering has been custom-designed. As illustrated in Fig. 4(a)

Trust-Guided Retrieval vs Connectivity-Driven Retrieval

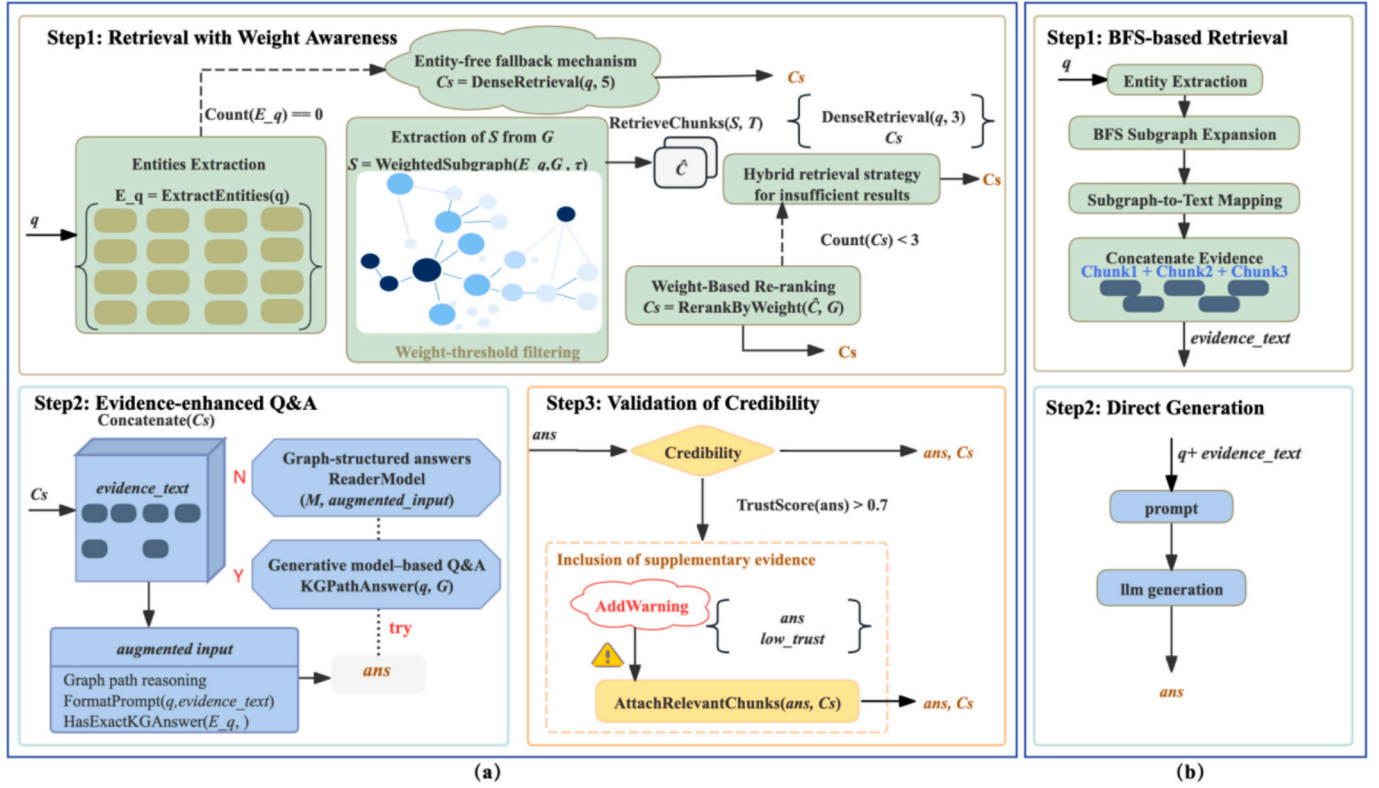


Fig. 4. Trust-Guided Retrieval vs Connectivity-Driven Retrieval. (a) TAG Trust-Guided Retrieval & Q&A strategy. The workflow combines trust-aware subgraph retrieval, evidence-augmented reasoning, and trust-based verification to generate reliable answers supported by contextual chunks. (b) Vanilla GraphRAG Q&A strategy. Vanilla GraphRAG relies entirely on generative models for answer production. Unlike connectivity-driven subgraph expansion in Vanilla GraphRAG, the proposed method introduces weight-based filtering, structured-first reasoning, and trust validation to improve reliability and mitigate hallucination.

and Algorithm 1, the proposed question-answer retrieval strategy is as follows:

In the proposed framework, T represents the text block index (the mapping from entities to text blocks), M is the question answering generation model such as LLMs, and C_s is the list of supporting evidence text blocks. E_q denotes the set of entities extracted from the question, S is the knowledge subgraph filtered by weight, τ is the weight threshold, is the set of candidate text blocks retrieved initially, and ans is the final answer, composed of three components: *text*, *source*, *trust*. Specifically, *evidence_text* is the concatenated evidence text, and *augment_input* refers to the augmented input text fed into the model. The Reader Model M is a generative Large Language Model (LLM) that produces natural language answers by synthesizing the user query q and the trust-aware evidence C_s . It serves as a robust fallback mechanism when a direct knowledge graph path is unavailable.

As illustrated in Fig. 4(b), in Vanilla GraphRAG, the retrieval and question answering process is based on the connectivity of entities within the knowledge graph, which expands subgraphs and then maps them to text blocks. These text blocks are then used to generate answers. In this pipeline, relation edges are treated uniformly, without explicitly modeling their reliability or trust. By introducing trust weights as the core filtering and sorting criterion, it is integrated into the entire retrieval and question answering process. During the answer generation phase, the graph paths are used directly along with their corresponding weights, which helps filter out noise. High-weight graph paths are prioritized for generating structured answers to avoid hallucinations in generative models, and the generation model is only called when a structured answer cannot be produced. As illustrated in Fig. 4(a), unlike the breadth-first traversal (BFT) based entity-centric retrieval of Vanilla GraphRAG, this method optimizes retrieval with weight-driven mechanisms. By applying threshold filtering, dynamic adjustment, and weight-

based sorting, quality control is performed at each stage of the information retrieval process. This approach results in higher precision and trustworthiness in the output.

Algorithm 1 Proposed Knowledge-Graph Enhanced Retrieval Q&A strategy

Algorithm 1: Trust-Guided Retrieval & Q&A strategy
 Require: User question q , Knowledge Graph G , Text index T , Reader model M
 Ensure: Answer ans with supporting chunks C_s
 1: I. Weighted perception retrieval:
 2: $E_q = \text{ExtractEntities}(q)$
 3: if $\text{Count}(E_q) == 0$ then
 4: $C_s = \text{DenseRetrieval}(q, 5)$
 5: else
 6: $S = \text{WeightedSubgraph}(E_q, G, \tau)$
 7: $\hat{C} = \text{RetrieveChunks}(S, T)$
 8: $C_s = \text{RerankByWeight}(\hat{C}, G)$
 9: if $\text{Count}(C_s) < 3$ then
 10: $C_s += \text{DenseRetrieval}(q, 3)$
 11: end if
 12: end if
 14: II. Evidence enhances Q&A
 15: $\text{evidence_text} = \text{Concatenate}(C_s)$
 16: $\text{augment_input} = \text{FormatPrompt}(q, \text{evidence_text})$
 18: try:
 20: if $\text{HasExactKGAnswer}(E_q, G)$ then
 21: $\text{ans} = \text{KGPathAnswer}(q, G)$
 23: else
 25: $\text{ans} = \text{ReaderModel}(M, \text{augment_input})$
 27: end if
 33: III. Credibility verification
 34: if $\text{TrustScore}(\text{ans}) < 0.7$ then
 35: $\text{ans} = \text{AddWarning}(\text{ans}, \text{low_trust})$
 36: $\text{AttachRelevantChunks}(\text{ans}, C_s)$
 37: end if
 39: return ans, C_s

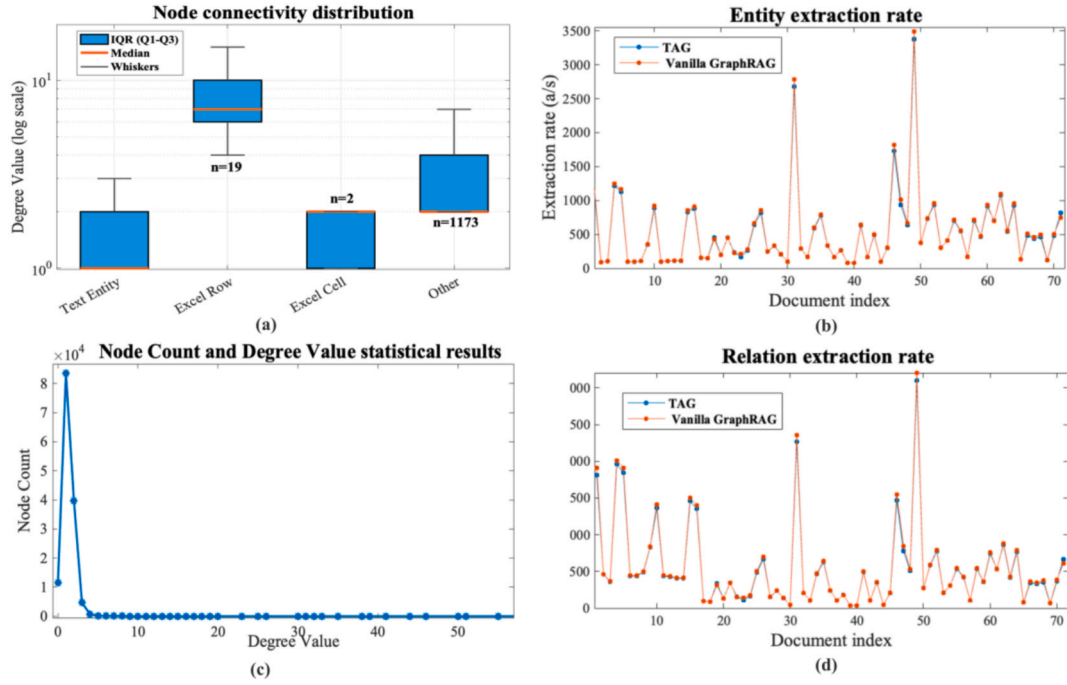


Fig. 5. (a) An illustration of the node connectivity distribution, (b) a comparison of entity extraction rates, (c) node degree statistics for the corresponding knowledge graph of the file collection, (d) a comparison of relation extraction rates.

4. Experimental results

In this section, TAG is evaluated for trustworthy question answering in safety-critical engineering equipment testing. Comparisons are limited to KG-RAG variants to isolate the effect of trust weighting under a consistent graph-based retrieval setting. The system is instantiated as a hybrid retrieval architecture, where subgraph search over the knowledge graph and text chunk retrieval operate under a shared retrieval budget and are combined by a trust-guided reranker.

To validate TAG, we conduct a case study on power transformer monitoring, using a dataset composed primarily of offline inspection reports (e.g., box-type substation test records). These reports exemplify the data commonly encountered in engineering practice and serve to demonstrate the framework's core capability of establishing a clear proof-of-concept for the text processing and evidence integration aspects.

Based on this dataset, we construct a Q&A benchmark from power transformer testing and maintenance documents, treating power transformers as a typical instance of critical engineering equipment. The Vanilla GraphRAG framework is adopted as the reference for evaluating the improvements introduced in this study. Knowledge graph construction is first examined using a curated dataset to assess the contribution of the trust-assisted mechanism. Subsequently, question answering dialogues are conducted, and the reliability of the resulting answers is measured to evaluate the overall effectiveness of the enhanced approach.

4.1. Experiment setup

The proposed approach was deployed on a server with an RTX 4090D GPU (24 GB). To ensure a fair comparison between the proposed and vanilla methods, the same version of Llama 3 was employed for all agents in the answer generation process.

4.2. Knowledge graph construction and quality evaluation

To evaluate the graph generation module, particularly its trust-aware mechanisms, we constructed a dataset of 71 documents. These properties are not unique to power transformers but are common in engineering equipment documentation, where strict regulatory standards coexist with heterogeneous document formats.

This dataset was curated to reflect the core challenges of power grid documentation: multiple sources, mixed formats, and varying levels of trustworthiness. It is composed of two categories: 57 unstructured documents, which typically provide trustworthy and standardized knowledge, and 14 structured data files, which often contain operational data that may be noisy or have lower initial trust. All sources are single domain but multi-source and cross-format, including national and industry standards, manufacturer manuals, inspection and test reports, operation logs. Representative examples of the regulatory standards, document formats, unstructured/semi-structured documents, and structured data fields used in the experiments are provided in [Appendix A \(Table A1 and Table A2\)](#).

Each triple includes provenance, an authority tier, version or time validity, a clause identifier, unit normalization, and condition corrections for temperature and voltage class.

To illustrate the characteristics of these files, [Fig. 5](#) reports the node degree distribution of the graph built from the data file collection. [Fig. 5 \(a\)](#) shows that text entity nodes account for about 27% of the total, while other entity types, such as formulas, are also present in the files. In addition, structured data nodes, such as Excel rows, make up about 23% of the total and include fault records and testing reports, which can vary in trustworthiness. This supports the need for the trust-aware mechanism introduced in this method.

Given the trustworthiness challenges in structured data, the proposed method adds an additional interface for structured data to the TAG framework and uses an extraction method better suited to

Table 1

Comparison of entity and relation extraction rates under different chunk sizes.

Method	Chunk size	Entity extraction rate (items/s)	Relationship Extraction Rate Entity Extraction Rate (items/s)
TAG	100	2965.88	2662.02
	500	2858.87	2570.59
	1000	2946.74	2643.92
Vanilla Graphrag	100	3483.77	3151.21
	500	3425.85	3099.48
	1000	3499.56	3165.10

structured files. In contrast, the baseline GraphRAG framework applies the same extraction method to both structured and unstructured sources. Fig. 5(c) shows the relationship between node ranking and node degree for the data file collection.

After introducing the dataset, we evaluated improvements in entity extraction and graph generation by comparing the proposed method with the traditional method. The resulting graphs show sparsity and density levels that fall within expected ranges for this dataset.

Fig. 5(b) and Fig. 5(d) compare the extraction rates of the proposed method and the traditional method for different data files, with the chunk size set to 500. As shown, the entity extraction rate curve, relation extraction rate curve, and memory usage curve for the proposed method closely resemble those of the traditional method, suggesting that the Trust-Aware Extraction Agent and the improvements introduced in this method do not add substantial computational overhead. The overall running speed of both methods is comparable.

To compare the graph generation rate under different chunk sizes, additional comparisons for three representative chunk sizes were performed, as shown in Table 1. When the chunk size is the same, the entity extraction rate, relation extraction rate, and average memory usage for the proposed method are similar to those of the traditional method. Both methods also show the same trend in their respective metrics as the chunk size is adjusted. After analyzing the extraction rates, the focus shifts to entity extraction quality and knowledge graph generation quality.

To improve practical transparency, we further summarize the practical overhead of TAG. We additionally provide summarized throughput and memory-related statistics in Table 2. Since both methods use the same frozen Llama3 model without fine-tuning, these results reflect pipeline-level overhead rather than model training cost. Overall, the additional overhead introduced by the trust-aware graph construction/fusion stage is limited under the present experimental setting, which helps make the performance–cost trade-off of the TAG framework more explicit.

In power grid data management, where large volumes of documentation must be consulted during fault diagnosis, the speed of user query and retrieval efficiency are very important. If the graph results are too complex and overly sophisticated retrieval strategies are used, the system response speed may further decrease. The proposed method therefore aims to balance efficiency and the correctness of the knowledge used in the answers. Unlike methods such as GraphAgent, this framework does not rely on a training set, making it more adaptable to zero-shot scenarios in the power grid domain.

Fig. 6 compares the structural evolution of TAG and Vanilla

GraphRAG during heterogeneous graph fusion. Multiple structural and performance metrics are jointly analyzed to assess the effectiveness and engineering feasibility of the proposed trust-aware mechanism. Fig. 6(a) shows the variation in the number of nodes and edges during the graph fusion process. Fig. 6(b) illustrates the evolution of the average degree and graph density. Fig. 6(c) presents the variation in the number of connected components. Fig. 6(d) reports the processing time as the fusion step increases.

In terms of graph scale and fusion behavior, both methods exhibit rapid growth in node and edge counts during early fusion steps, followed by stabilization. However, TAG maintains a smaller overall graph size compared to Vanilla GraphRAG, indicating that trust-guided alignment avoids unnecessary structural expansion. When combined with the connected components metric, a clearer difference emerges: TAG produces significantly fewer connected components and reaches structural convergence earlier. This suggests that the cross-graph trust validation facilitates more effective entity alignment and reduces fragmentation across heterogeneous data sources, resulting in a more integrated graph structure.

Regarding structural quality indicators, TAG demonstrates higher average degree and higher graph density than Vanilla GraphRAG. The increased average degree reflects stronger local connectivity among entities, while the higher density indicates more consolidated relational structures under controlled node growth. Importantly, this enhancement is achieved without uncontrolled graph expansion, as evidenced by the stabilized scale metrics. Rather than relying on sparsification, the trust-weight mechanism promotes the integration of high-trust relations, leading to a more cohesive knowledge structure. Such structural consolidation provides richer relational paths that can support downstream reasoning and question answering, as further evaluated in the QA experiments.

From a computational perspective, the processing time for both methods increases approximately linearly with respect to fusion steps. Although TAG introduces additional trust-weight propagation and cross-validation operations, its per-step processing time remains within the same order of magnitude as Vanilla GraphRAG (approximately 1.9 higher). No exponential growth or instability is observed. This indicates that the proposed mechanism achieves structural refinement with manageable computational overhead, maintaining practical feasibility for engineering applications in safety-critical domains such as power equipment maintenance.

Overall, the results show that TAG improves structural integration and relational consolidation while preserving controllable graph scale and computational complexity. These structural characteristics establish a stable foundation for reliable multi-source knowledge graph construction and subsequent QA performance.

Fig. 7 illustrates the changes in the trust weights introduced by the proposed method during the fusion process. The trust weights of edges and nodes are iteratively updated, showing larger fluctuations in the early stage. This suggests that the initial phase of graph formation establishes most multimodal semantic associations and entity linkages, leading to an increase in node connectivity. In the example power grid dataset, the trust weights gradually stabilize in the later stage. This stabilization supports the effectiveness of the fusion strategy in reducing isolation across modalities and improving the connectivity of the knowledge graph. Trust propagation adjudicates conflicts by authority

Table 2

Practical overhead statistics of the proposed framework and Vanilla GraphRAG. Memory is summarized as the absolute fluctuation of the recorded memory variation. Zero-valued entries indicate no observable memory change for the corresponding file.

Method	Avg. Entity Throughput (items/s)	Avg. Relation Throughput (items/s)	Avg. Absolute Volatility of Non-zero Events (MB)	Peak Memory Fluctuation (MB)
TAG	535.29	544.00	22.74	80.37
Vanilla GraphRAG	548.91	558.16	21.49	73.66

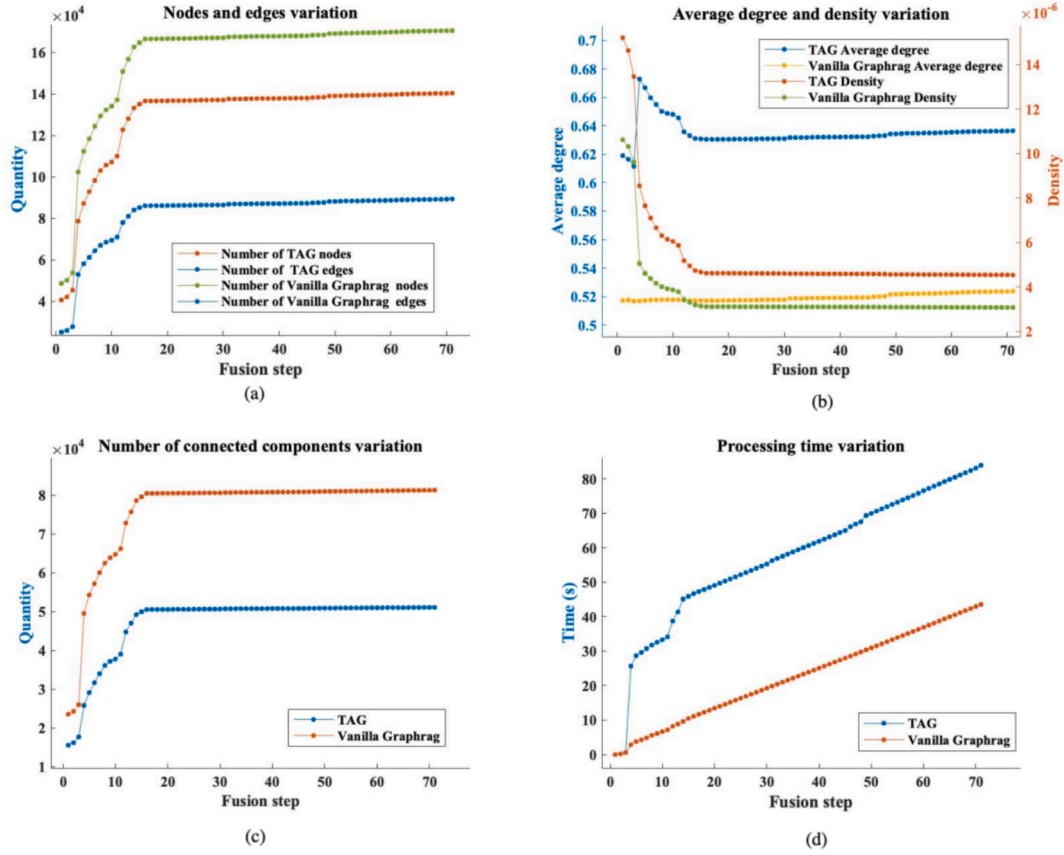


Fig. 6. Comparative evaluation of structural indicators and computational performance during the graph generation process (TAG and Vanilla GraphRAG). The subplots track the evolution of node-edge scale, average degree, density, connected components, and processing time across fusion steps. The results suggest that the TAG strategy achieves superior knowledge consolidation (higher density and average degree) and better structural integration (fewer connected components) with a manageable and linear increase in computational cost. This confirms that the TAG construction is stable, topologically optimized, and scalable, providing a reliable foundation for trust-guided retrieval and reasoning.

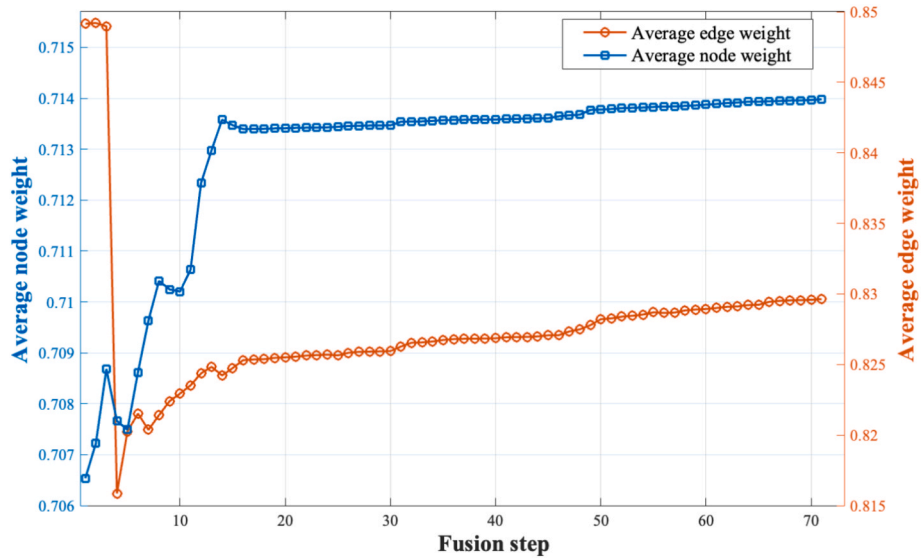


Fig. 7. The trust-weighted update process in the heterogeneous graph fusion process of TAG, both node and edge weights undergo rapid adjustments in the early fusion steps and progressively stabilize as more cross-source evidence is integrated. This convergence pattern indicates that the trust-aware updating mechanism effectively filters unreliable information, reinforces consistent multi-source knowledge, and establishes a stable and reliable weighting structure for downstream retrieval and reasoning.

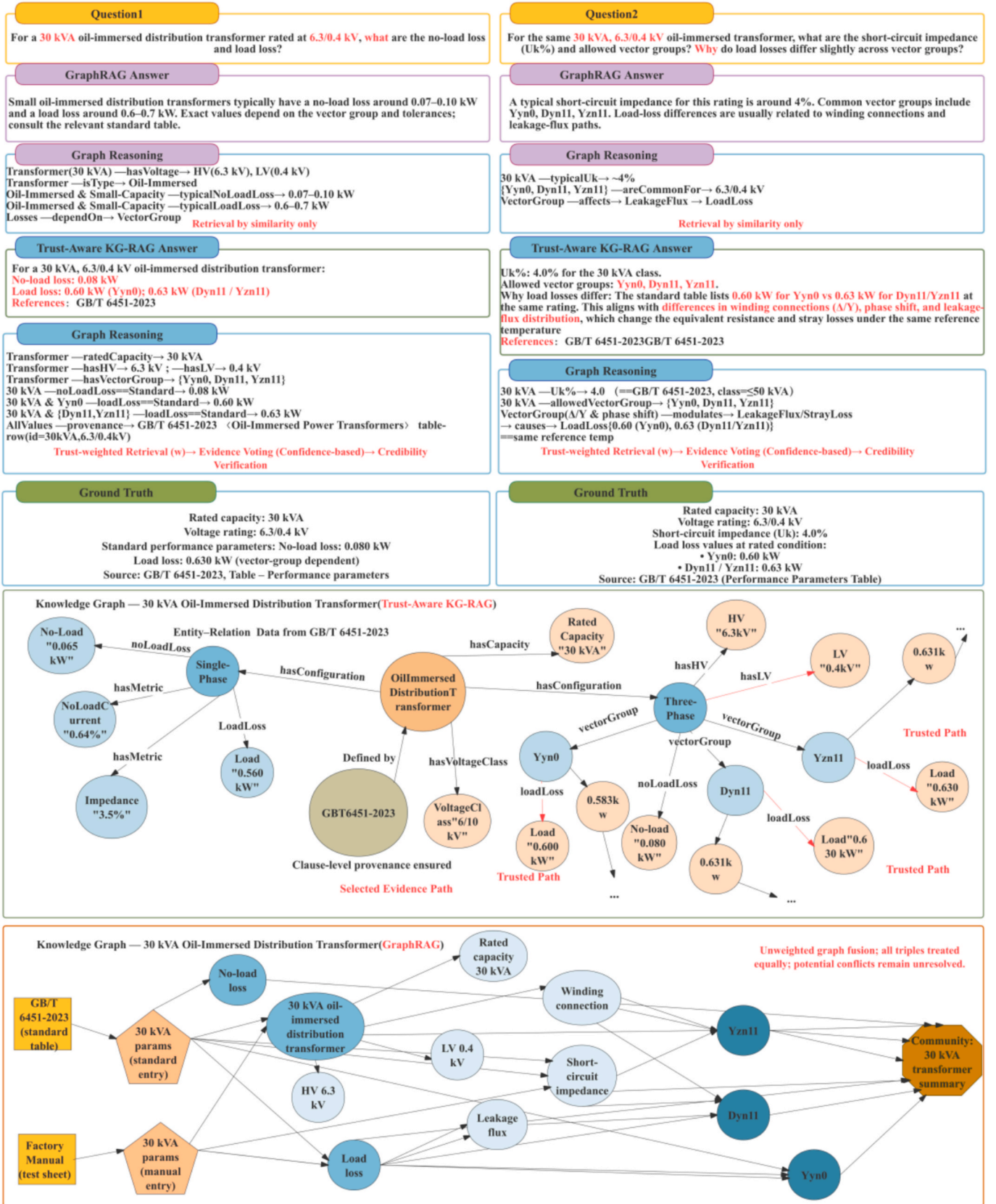


Fig. 8. Test of the Trust-Guided Retrieval & Q&A system. The figure compares the outputs, reasoning paths, and supporting evidence of Vanilla GraphRAG and the proposed TAG on two transformer-related queries. The trust-guided approach retrieves more accurate parameters, produces cleaner and more standard-aligned reasoning paths, and yields interpretable answers grounded in high-trust multi-source knowledge, demonstrating substantial improvements in retrieval precision and reasoning reliability.

Table 3

Stratified performance comparison under different evidence interaction and conflict conditions.

Question Type	Samples	GraphRAG			TAG			$\Delta F1$
		P	R	F1	P	R	F1	
Single-document questions,	100	0.85	0.83	0.84	0.91	0.89	0.90	+0.06
Cross-document questions, without conflict	80	0.83	0.79	0.81	0.86	0.83	0.84	+0.03
Cross-document questions, with conflict	100	0.47	0.56	0.51	0.70	0.90	0.79	+0.28

tier and version precedence and aligns thresholds under operating conditions. This is particularly important for grid testing, where revisions of standards and condition dependent corrections can change pass or fail decisions.

4.3. Trust-guided retrieval & Q&A dialogue experiment

After analyzing and evaluating the entity generation process and graph quality of the proposed method, question answering experiments were conducted to assess the practical value of the TAG framework in power equipment testing. To better approximate real-world applications, the sample corpus was extended with multi-source power equipment test data, including standard regulations, manufacturer manuals, and experimental records, to construct a large-scale domain knowledge graph. Representative questions were then selected to examine both clause-level accuracy and reasoning transparency.

Fig. 8 illustrates the end-to-end reasoning process and the differences between the Vanilla GraphRAG and the proposed TAG. While Vanilla GraphRAG retrieves evidence solely based on semantic similarity, the Trust-Aware framework integrates a continuous trust score w throughout retrieval and inference. During retrieval, each candidate clause is re-ranked by a trust-weighted relevance score, ensuring that authoritative and up-to-date standards are prioritized. In the reasoning phase, evidence voting aggregates high-trust paths, and credibility verification eliminates low-trust or conflicting statements before generating the final answer.

An enlarged version of the source-document excerpt corresponding to the reference answer in Fig. 8 is provided in Appendix B for readability. This supplementary figure allows readers to inspect the supporting evidence from the regulatory standard document in greater detail.

The resulting knowledge graphs differ substantially. The Vanilla GraphRAG fusion treats all triples equally and may leave potential conflicts unresolved. By contrast, the TAG constructs a refined subgraph in which edges and nodes carry trust weights, highlighting high-trust evidence chains that connect authoritative standards to the query results. This process enhances clause-level grounding and helps ensure that the final answer reflects verified, regulation-compliant knowledge, supporting the reliability and interpretability of the trust mechanism in domain reasoning.

To better understand where the proposed trust-aware KG-RAG framework is most beneficial, we further conduct a stratified analysis based on evidence interaction complexity and conflict level. Specifically, we divide the evaluation questions into three categories: single-document questions, cross-document questions without conflicts, and cross-document questions with conflicting information across sources.

As shown in Table 3, the performance gap between the two methods is relatively small on single-document questions, indicating that the additional trust-aware mechanism provides limited marginal benefit when no substantial cross-source aggregation or conflict resolution is required. On cross-document questions without conflicts, TAG yields a

moderate improvement, suggesting that trust-aware graph-based integration is still helpful for multi-source evidence aggregation. The largest gain is observed on cross-document questions with conflicts, suggesting that trust-aware reasoning is especially helpful when inconsistent evidence must be reconciled across sources.

Evaluation is conducted using clause-level grounding with a shared retrieval budget across all methods. To isolate the effect of trust-aware reasoning under a controlled setting, we compare TAG not only with graph-based baselines (Fast-GraphRAG, LightRAG, and Vanilla GraphRAG) but also with non-graph baselines, including traditional text RAG and Dual-Modal RAG (Text + KG). Fig. 9 and Table 4 report Precision, Recall, and F1 under identical top-k settings, matched context lengths, and the same chunking strategy. All clause citations are validated against authoritative source documents.

Under this controlled setting, TAG achieves the best overall performance. Compared with Vanilla GraphRAG, TAG improves end-to-end F1 by 54.5%, with corresponding gains of 49.3% in Precision and 60.8% in Recall. TAG also outperforms the text-centric baselines by a clear margin, indicating that trust-aware structured reasoning is beneficial in engineering corpora characterized by multi-source evidence, revision-sensitive standards, and cross-format documentation. These improvements are consistent with the role of the unified trust pipeline, which assigns trust weights w to entities, relations, and clauses on the basis of authority tier, version precedence, and condition correction.

The relatively weak performance of Dual-Modal RAG (Text + KG) further highlights the importance of trust calibration. Although this baseline combines textual and graph-derived evidence, its fusion strategy remains largely uncalibrated. Retrieved text and KG triples are concatenated without cross-source trust validation, so redundant or inconsistent triples may introduce noise rather than complementary support. Recall improves only marginally compared with traditional text RAG, while precision decreases due to evidence contamination and misaligned retrieval focus. As a result, multimodal integration alone does not guarantee improved answer quality; without trust-aware co-ordination, it may even reduce precision through evidence contamination and unstable grounding.

In TAG, a single trust pipeline assigns a trust weight to entities, relations, and clauses. During retrieval, candidates are re-ranked by trust-weighted relevance, so high-authority and up-to-date standards are retained. This reduces early recall loss and increases recall at a fixed budget, addressing a key weakness in Fast-GraphRAG, where tiered retrieval without trust parameters may prune valid but lower-frequency evidence.

During answer generation, high-weight paths and clauses are aggregated through evidence voting, and a decision threshold on accumulated trust mass suppresses stale or conflicting evidence, which improves precision, a key differentiator from LightRAG, which lacks explicit conflict mitigation. The final credibility verification step checks tentative answers against the highest-weight criteria and filters outputs without sufficient trust mass or condition alignment, stabilizing overall trustworthiness. The advantages of TAG are most evident when

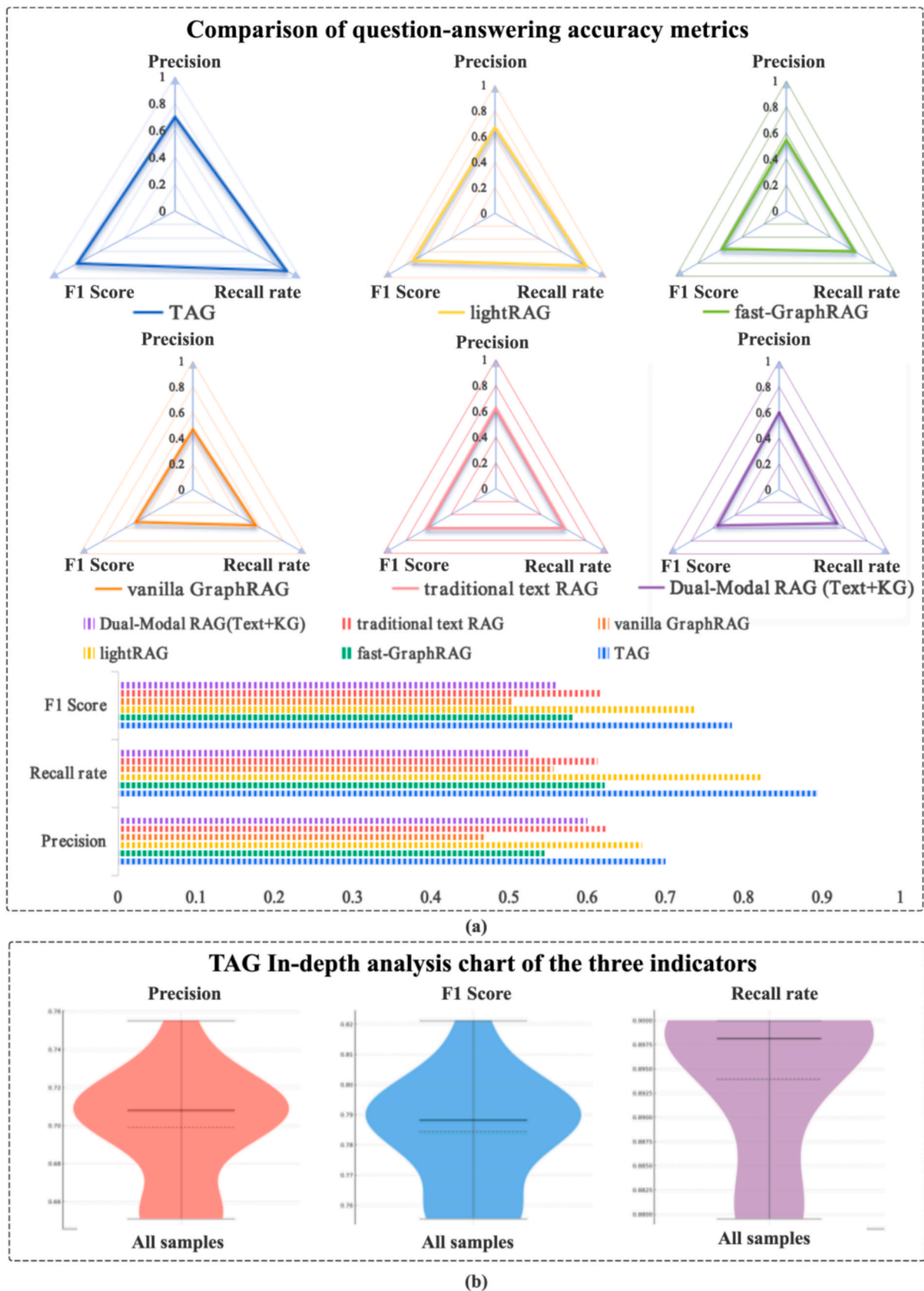


Fig. 9. Overall comparison of question answering accuracy and detailed indicator analysis of TAG. (a) Performance comparison among five representative KG-RAG and text-RAG baselines and TAG across precision, recall, and F1 score. TAG achieves higher overall scores than the baselines, indicating improved accuracy and a more balanced retrieval and reasoning capability. (b) Violin plot analysis of TAG precision, recall, and F1 score over all test samples, showing stable distributions with high central values, which supports the robustness of TAG Q&A performance.

Table 4

Precision, Recall rate, F1 Score of different frameworks.

	Precision	Recallrate	F1Score
TAG	0.699	0.894	0.785
Fast-GraphRAG	0.545	0.622	0.581
LightRAG	0.669	0.821	0.737
Vanilla GraphRAG	0.468	0.556	0.508
Traditional text RAG	0.624	0.613	0.619
Dual-Modal RAG (Text + KG)	0.560	0.525	0.560

questions require cross-document evidence aggregation. As shown in the stratified analysis in Table 3, performance gains are relatively small on single-document questions, where direct matching is often sufficient, but become larger on cross-document questions, especially when conflicting information must be reconciled across standards, manuals, and inspection records. This pattern supports the view that the main contribution of TAG lies not in simple retrieval enhancement, but in trust-aware conflict handling and evidence prioritization under multi-source inconsistency.

Among the graph-based baselines, LightRAG remains competitive on clause-centric corpora because its lightweight indexing reduces summarization loss. However, TAG surpasses it when answer validity depends on authority tier, version recency, or conflict resolution. Similarly, Fast-GraphRAG benefits from efficient graph traversal but lacks explicit trust calibration, which limits its ability to preserve valid but lower-frequency evidence under a fixed retrieval budget. Overall, these comparisons suggest that the principal benefit of TAG comes from integrating trust modeling into graph construction, retrieval, and answer verification in a unified manner.

On the inference side, the hybrid retrieval strategy of TAG further reduces variance. Text-first retrieval recovers long, clause-anchored explanations, whereas graph-first retrieval preserves multi-hop connectivity among equipment, parameters, and test items. The unified trust-guided pipeline leverages the same weight w for both channels, preventing unsupported or stale reasoning chains. Gains are most significant on cross-document queries that require integrating standards, manuals, and inspection records with frequent revisions, while single-clause lookups show smaller improvements where direct matching suffices. Precision may initially lag recall when the candidate retention policy emphasizes safety-oriented recall, a deliberate trade-off suitable for compliance-critical power equipment testing and adjustable through higher decision thresholds.

Although prompt robustness is not the primary focus of this work, our framework still includes an LLM-based stage, and it is therefore useful to examine whether the end-to-end system is sensitive to moderate changes in question wording. Since the prompt in TAG framework is dynamically constructed rather than manually written as a fixed template, the relevant robustness issue in our setting is sensitivity to semantically equivalent rephrasings of the input question.

To provide a minimal empirical check, we selected a small representative subset of the evaluation set and created two semantically equivalent paraphrases for each question while keeping all other settings unchanged. We selected 20 representative questions covering various device types and task categories. We then ran the full pipeline on these reformulated inputs and evaluated the outputs using the same Precision, Recall, and F1 metrics as in the main experiments. The results

Table 5

Performance under semantically equivalent query rephrasings. All formulations preserve the same question intent, and only moderate wording changes are introduced.

Query Formulation	Samples	Precision	Recall	F1
Original wording	20	0.701	0.892	0.785
Paraphrase A	20	0.690	0.784	0.734
Paraphrase B	20	0.706	0.888	0.785

are summarized in Table 5. Overall, the performance variation across different formulations is limited, providing preliminary evidence of stability under moderate query rewording in this small-scale check. Nevertheless, this check is small in scale and should not be interpreted as a comprehensive robustness evaluation.

4.4. Discussion

The results suggest that the main advantage of TAG lies in the combination of trust-aware graph construction and trust-guided inference. During graph construction and fusion, trust priors derived from authority tier, version recency, and condition correction help preserve high-value evidence while down-weighting stale or lower-authority alternatives. During retrieval and answer generation, the same trust signal is used to re-rank candidate evidence, aggregate support, and filter weakly grounded outputs. This unified design appears to be particularly useful in engineering QA settings where multiple documents describe the same object or procedure from different viewpoints and with different reliability levels.

The stratified results further clarify the boundary of this benefit. TAG provides only limited marginal improvement on single-document questions, where exact clause matching is often sufficient. By contrast, the gains become more substantial on cross-document questions, especially when conflicting evidence must be reconciled across standards, manufacturer manuals, and inspection or test reports. This pattern indicates that the contribution of the trust-aware mechanism is most evident when the task requires source prioritization and conflict resolution rather than simple retrieval. The observed Precision–Recall profile is also consistent with this interpretation: the system favors conservative evidence retention to avoid missing authoritative clauses, which supports Recall but may admit some redundant or weakly informative candidates before final verification.

Several limitations should also be noted. First, the effectiveness of trust calibration depends on the quality of metadata and normalization. Missing clause identifiers, incomplete revision information, or inconsistent unit and operating-condition normalization can reduce the discriminative value of trust scores during fusion and retrieval. Second, the current study focuses on a power-transformer case study and therefore does not establish external validity across all engineering domains. The trust signals used here, such as authority tier, version precedence, and condition correction, are meaningful in this setting, but their exact formulation may require adaptation in other domains. Third, the current method is more effective for explicit multi-source inconsistency than for subtle or weakly expressed conflicts. When reliability cues are ambiguous or conflicts are only implicit, the trust-aware mechanism may not fully resolve the underlying uncertainty.

The additional analyses in this revision should also be interpreted with appropriate scope control. The prompt-related experiment provides only a limited robustness check under moderate query rewording; it does not constitute a comprehensive prompt-robustness study. Similarly, the reported runtime and memory statistics are intended to improve practical transparency at the pipeline level, rather than to establish a full deployment-cost evaluation. We therefore interpret these analyses as supplementary evidence that clarifies behavior under the current setting, not as complete validation of robustness or deployment readiness.

Within these boundaries, the results still suggest that trust-aware KG-RAG is a promising direction for other safety-critical engineering environments that share similar documentation characteristics, such as multi-source procedural records, evolving standards, and strong traceability requirements. Potentially relevant settings include maintenance, inspection, and compliance-oriented reasoning tasks in domains such as aviation, energy, and transportation. However, this broader relevance should be understood as a transfer hypothesis supported by structural similarity of the documentation conditions, rather than as an empirically established claim in this paper.

Future work should therefore proceed along two directions. One is methodological refinement, including adaptive retrieval budgets, precision-oriented thresholds, improved duplicate suppression, and stronger clause-aware graph-text matching. The other is broader validation, including additional engineering domains, richer temporal and operating-state context, more detailed ablation studies, and settings involving streaming data or more specialized data types such as formulas and equations. These extensions would help clarify both the sources of the current gains and the range of conditions under which the trust-aware design remains effective.

5. Conclusion

This paper proposes TAG, a trust-aware KG-RAG framework for question answering over safety-critical engineering documentation. The framework integrates trust modeling into graph construction, graph fusion, retrieval, and answer verification so that entities, relations, and clauses are not treated as equally reliable during reasoning. In the evaluated power-transformer case study, this design improves answer quality and strengthens alignment between generated responses and authoritative source evidence.

Experimental results on an engineering-equipment Q&A benchmark derived from power-transformer testing documents support the framework's effectiveness. Compared to the Vanilla GraphRAG baseline, the proposed method improves precision by over 49.3% and recall by over 60.8%, while providing more consistent alignment between answers and their cited evidence.

Beyond this application, TAG provides a promising trust-aware framework for knowledge-intensive reasoning in safety-critical engineering settings with similar documentation characteristics. Many safety-critical domains share similar challenges, including multi-source and cross-format data, evolving technical standards, and strict traceability requirements. The trust-aware mechanisms developed here, including authority-tier weighting, cross-version validation, and clause-level provenance tracking, can be adapted to other industrial domains such as automated production, predictive maintenance, robotic inspection, and intelligent manufacturing systems. By embedding trust evaluation into each stage of retrieval and reasoning, the framework offers a transferable direction toward reliable, interpretable, and verifiable LLM-based systems.

At the same time, the present study should be interpreted within its current scope. The evaluation is conducted on offline engineering documentation in a power-transformer case study, while the prompt-

related analysis is only a limited robustness check and the cost-related analysis is intended as practical transparency rather than a full deployment study. Therefore, the current results establish the effectiveness of TAG within the evaluated setting, but broader validation across domains, prompt conditions, and deployment scenarios remains future work.

More broadly, TAG suggests a promising trust-aware design perspective for knowledge-intensive reasoning in safety-critical engineering environments with similar documentation characteristics.

Future work will extend the evaluation to other RAG paradigms and engineering domains, investigate streaming data integration, enrich graph representations for more specialized data types such as formulas and equations, and conduct more detailed ablation studies to better understand the sources of performance gains.

CRedit authorship contribution statement

Qi Shi: Writing – original draft, Validation, Software, Methodology, Conceptualization. **Ye Yang:** Writing – original draft, Methodology, Investigation, Conceptualization. **Jiaqi Di:** Writing – original draft, Software. **Lin Li:** Validation, Supervision, Project administration. **Jie Liu:** Supervision, Funding acquisition. **Yumin Hu:** Supervision, Funding acquisition. **Jinhua Xiao:** Writing – review & editing, Supervision. **Zhongxu Hu:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing 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.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (No.2024YFC3505605), National Natural Science Foundation of China (No.52575572, No.52575115), Key Project of the Natural Science Foundation of Hubei Province (No. 2026AFA057), Science and Technology Program of Hubei Province (No.2024BEB025), Joint Research Project of the Yangtze River Delta Science and Technology Innovation Community (No.2024CSJGG1400), UNSW-HUST Global Research & Impact Program (No.5003100202).

Appendix

Appendix A. . Representative examples of data Sources, document Types, and structured data fields

To improve the traceability of the experimental data while keeping the main text concise, this appendix provides representative examples of the regulatory standards, document formats, unstructured/semi-structured documents, and structured data files used in the experiments. These examples are intended to illustrate the composition and heterogeneity of the data sources involved in the proposed framework. The representative examples cover TXT, DOCX, PDF, XLSX, X_T, JSONL, CSV, JSON, and GraphML formats.

Table A1

Representative regulatory standards and document examples used in the experiments.

Category	Identifier	Representative title / description	Format	Source category / note
Regulatory standard	GB20052-2020	Energy efficiency limit values and energy-efficiency grades for power transformers	TXT	Standards corpus
Regulatory standard	GB1094.1-2013	Power transformers — Part 1: General	TXT	Standards corpus
Regulatory standard	GB/T1094.2-2013	Power transformers — Part 2: Temperature rise of liquid-immersed transformers	TXT	Standards corpus
Regulatory standard	GB/T6451-2015	Technical parameters and requirements of oil-immersed power transformers	TXT	Standards corpus
Regulatory standard	GB/T1094.10-2022	Power transformers — Part 10: Determination of sound levels	TXT	Standards corpus

(continued on next page)

Table A1 (continued)

Category	Identifier	Representative title / description	Format	Source category / note
Regulatory standard	GB/T1094.3-2017	Power transformers — Part 3: Insulation levels, dielectric tests and clearances in air	TXT	Standards corpus
Regulatory standard	GB/T1094.11-2007	Power transformers — Part 11: Dry-type transformers	TXT	Standards corpus
Regulatory standard	GB/T507-2002	Insulating oils — Determination of breakdown voltage	TXT	Standards corpus
Regulatory standard	GB/T5654-2007	Liquid insulation — Relative permittivity, dielectric loss factor and DC resistivity measurements	TXT	Standards corpus
Regulatory standard	GB/T7251.1-2023	Low-voltage switchgear and controlgear assemblies — Part 1: General	TXT	Standards corpus
Regulatory standard	JB/T501-2021	Technical parameters and requirements of oil-immersed power transformers	TXT	Standards corpus
Unstructured document	XM99018	Vocational competence training manual for sampling inspection technicians in power grid materials (fittings volume)	TXT	Training / reference material
Unstructured document	SYN-WI-OIL-BDV-2026	Work instruction: breakdown voltage test for transformer insulating oil	DOCX	Enterprise work instruction
Unstructured document	SYN-CHK-SWGR-INCOMING-2026	Incoming inspection checklist for low-voltage switchgear assemblies	PDF	Inspection checklist
Semi-structured document	SYN-WI-TEMPLATE-SUBSTATION-2026	Substation equipment maintenance work-instruction template	XLSX	Operational template
Semi-structured document	SYN-CAD-INSULATOR-FITTING-2026	CAD-related file for insulator fitting	X_T	Engineering design file

The following table further summarizes representative structured data files used in the experiments, together with their key fields, in order to illustrate the heterogeneity of tabular, JSON/JSONL, and graph-based data sources involved in the proposed framework.

Table A2

Representative structured data files and their key fields.

Structured dataset / file	Format	Source category / note	Column count	Representative fields
Power-grid sampling inspection (Excel template)	XLSX	Template (schema only)	13	Inspection date; Inspection order No.; Supplier; Material code; Material name; Equipment type; Voltage level (kV); Test item; Standard requirement; Measured value; Result; Inspector; Notes
Q&A evaluation set (power-grid materials)	JSONL	Q&A pairs	2	question; ground_truth
Q&A set (power-grid materials)	JSONL	Q&A pairs	2	question; ground_truth
Text chunks export	CSV	Intermediate chunk-level export	3	text; source; chunk_index
Knowledge graph export	GraphML	Knowledge graph	5	metadata; similarity; text; type; weight
Chunk store export	JSON	Intermediate storage export	8	content; tokens; file_path; chunk_order_index; llm_cache_list; create_time; update_time; id
Vector DB export	JSON	Intermediate storage export	3	embedding_dim; data; matrix
Incoming inspection batch detail report	XLSX	Inspection record	16	Inspection date; Inspection order No.; Inspection method; Supplier code; Supplier name; Material code; Material description; Equipment type; Voltage level (kV); Batch No.; Sample size; Defective samples; Defect type; Disposition method; Inspector; Remark
Distribution transformer parameters	XLSX	Parameter sheet	12	Transformer model; Rated power (kVA); Primary voltage (kV); Secondary voltage (kV); Vector group; No-load loss (W); Load loss (W); Impedance (%); Oil temperature rise (K); Sound level (dB); Energy efficiency grade; Standard
Transformer routine test results	CSV	Test-result export	8	equipment_id; test_item; standard_id; acceptance_criteria; measured_value; unit; result; test_date
Power-grid QA dataset	JSONL	Q&A pairs with engineering attributes	8	question; ground_truth; equipment_type; voltage_level_kv; standard_id; test_item; clause; note
Substation equipment ledger	JSON	Asset / equipment record	13	equipment_id; equipment_name; equipment_type; voltage_level_kv; location; commissioning_date; manufacturer; supplier; status; maintenance_cycle_months; last_maintenance_date; next_due_date; standard_ref
Synthetic KG schema	GraphML	Knowledge graph schema	9	node_type; equipment_id; equipment_type; voltage_level_kv; standard_id; relation_type; confidence; source_doc; clause

Appendix B. . Enlarged Source-Document evidence for the reference answer

This figure provides an enlarged view of the source-document excerpt corresponding to the reference answer shown in Fig. 8. It is included to improve readability and to allow readers to examine the supporting evidence from the regulatory standard document in greater detail.

5.1 Performance Parameters

Table 5: 20kV and 10kV Class Three-Phase Double-Winding No-Load Tap-Changing Distribution Transformers, 30kVA to 2500KVA								
Rated Capacity kVA	Voltage Combination and Tap Range			Connection Group Designation	No-Load Loss kw	Load Loss kw	No-Load Current %	Short-Circuit Impedance %
	High Voltage kv	High-Voltage Tap Range%	Low Voltage kv					
30					0.080	0.660/0.630	1.7	5.5
50					0.100	0.960/0.910	1.6	
63					0.120	1.14/1.09	1.5	
80					0.140	1.37/1.30	1.4	
100					0.160	1.64/1.57	1.3	
125					0.190	1.98/1.88	1.2	
160					0.230	2.41/2.30	1.1	
200					0.270	2.85/2.72	1.0	
250					0.232	3.34/3.18	0.96	

4.1 Performance Parameters

Table 1: 6kV and 10kV Class Three-Phase Double-Winding No-Load Tap-Changing Distribution Transformers, 30kVA to 2500KVA								
Rated Capacity kVA	Voltage Combination and Tap Range			Connection Group Designation	No-Load Loss kw	Load Loss kw	No-Load Current %	Short-Circuit Impedance %
	High Voltage kv	High-Voltage Tap Range %	Low Voltage kv					
30					0.080	0.630/0.600	1.20	4.0
50					0.100	0.910/0.870	1.04	
63					0.110	1.09/1.04	0.96	
80					0.130	1.31/1.25	0.96	
100					0.150	1.58/1.50	0.88	
125					0.170	1.89/1.80	0.88	
160					0.200	2.31/2.20	0.80	
200					0.240	2.73/2.60	0.80	
250					0.290	3.20/3.05	0.72	
315					0.340	3.83/3.65	0.72	
400					0.410	4.52/4.30	0.64	
500	6			Dyn11	0.480	5.41/5.15	0.64	
630	6.3	±2×2.5		Yzn11				
	10	±5	0.4	Yyn0	0.570	6.2	0.48	

4.1 Performance Parameters

Table 4: 6kV and 10kV Class Single-Phase Double-Winding No-Load Tap-Changing Distribution Transformers, 5kVA to 160KVA								
Rated Capacity kVA	Voltage Combination and Tap Range			Connection Group Designation	No-Load Loss kw	Load Loss kw	No-Load Current %	Short-Circuit Impedance %
	High Voltage kv	High-Voltage Tap Range %	Low Voltage kv					
5					0.025	0.130	0.96	3.5
10					0.035	0.235	0.88	
16					0.045	0.330	0.80	
20					0.050	0.385	0.72	
30	6				0.065	0.560	0.64	
40	6.3				0.080	0.700	0.64	
50	10	±5	0.22-0.24	ii0	0.095	0.855	0.56	
63	10.5	±2×2.5	2×(0.22/0.24)	ii6	0.115	1.02	0.48	
80	11				0.130	1.26	0.48	
100					0.150	1.48	0.48	
125					0.185	1.75	0.4	
160					0.230	2.13	0.4	

Fig. B1. Enlarged excerpt from the regulatory standard document used as evidence for the reference answer in .

Data availability

Data will be made available on request.

References

- [1] C. Deng, P. Pang, C. Wan, S. Zhao, Construction of automation operation management and control platform for power grid enterprises based on digital transformation, in: 2024 International Conference on Electrical Drives, Power Electronics & Engineering (EDPEE), IEEE, 2024, pp. 833–838.
- [2] G. Qian, Y. Junyi, H. Yu, S.U.N. Xiaolei, Z.H.U. Qianjin, Y.U. Tian, L.I. Zesen, Research on digital transformation architecture and path of power grid development planning business under new power system blueprint, Power Generation Technol. 43 (6) (2022) 851.
- [3] C. Chen, K. Zhao, J. Leng, C. Liu, J. Fan, P. Zheng, Integrating large language model and digital twins in the context of industry 5.0: framework, challenges and opportunities, Rob. Comput. Integr. Manuf. 94 (2025) 102982.
- [4] A. Christina, M.A. Salam, Q.M. Rahman, F. Wen, S.P. Ang, W. Voon, Causes of transformer failures and diagnostic methods—a review, Renew. Sustain. Energy Rev. 82 (2018) 1442–1456.
- [5] X. Li, L. Zhao, L. Zhang, Z. Wu, Z. Liu, H. Jiang, D. Shen, Artificial general intelligence for medical imaging analysis, IEEE Rev. Biomed. Eng. (2024).
- [6] J. Leng, K. Zheng, R. Li, C. Chen, B. Wang, Q. Liu, W. Shen, AIGC-empowered smart manufacturing: prospects and challenges, Rob. Comput. Integr. Manuf. 97 (2026) 103076.

- [7] Y. Li, L. Li, D. Hu, X. Hao, M. Litvak, N. Vanetik, Y. Zhou, Improving factual error correction for abstractive summarization via data distillation and conditional-generation cloze (2024). arXiv preprint arXiv:2402.08581.
- [8] Y. Huang, J. Song, Z. Wang, S. Zhao, H. Chen, F. Juefei-Xu, L. Ma, Look before you leap: an exploratory study of uncertainty analysis for large language models, *IEEE Trans. Softw. Eng.* (2025).
- [9] G. Marcus, The next decade in AI: four steps towards robust artificial intelligence (2020). arXiv preprint arXiv:2002.06177.
- [10] J. Lee, S. Ahn, D. Kim, D. Kim, Performance comparison of retrieval-augmented generation and fine-tuned large language models for construction safety management knowledge retrieval, *Autom. Constr.* 168 (2024) 105846.
- [11] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, C. Zhang, Quantifying memorization across neural language models. The Eleventh International Conference on Learning Representations, 2022.
- [12] P. Lee, S. Bubeck, J. Petro, Benefits, limits, and risks of GPT-4 as an AI chatbot for medicine, *N. Engl. J. Med.* 388 (13) (2023) 1233–1239.
- [13] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, B. Cui, Retrieval-augmented generation for ai-generated content: A survey (2024). arXiv preprint arXiv:2402.19473.
- [14] J. Dagdelen, A. Dunn, S. Lee, N. Walker, A.S. Rosen, G. Ceder, A. Jain, Structured information extraction from scientific text with large language models, *Nat. Commun.* 15 (1) (2024) 1418.
- [15] X. Chen, S. Jia, Y. Xiang, A review: knowledge reasoning over knowledge graph, *Expert Syst. Appl.* 141 (2020) 112948.
- [16] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G.D. Melo, C. Gutierrez, A. Zimmermann, Knowledge graphs, *ACM Comput. Surv. (CSUR)* 54 (4) (2021) 1–37.
- [17] M.E. Za'fer, A. Sajadi, B.M. Hodge, Semi-supervised multi-task learning based framework for power system security assessment (2024). arXiv preprint arXiv:2407.08886.
- [18] S. You, L. Zhu, Y. Liu, H. Liu, Y. Liu, M. Shankar, T. King, A survey on next-generation power grid data architecture, in: 2015 IEEE Power & Energy Society General Meeting, IEEE, 2015, pp. 1–5.
- [19] O.M. Butt, M. Zulfqarnain, T.M. Butt, Recent advancement in smart grid technology: future prospects in the electrical power network, *Ain Shams Eng. J.* 12 (1) (2021) 687–695.
- [20] H. Choi, J. Jeong, Domain-specific manufacturing analytics framework: an integrated architecture with retrieval-augmented generation and ollama-based models for manufacturing execution systems environments, *Processes* 13 (3) (2025) 670.
- [21] A. Fink, A. Rau, M. Reiser, F. Bamberg, M.F. Russe, Retrieval-augmented generation with large language models in radiology: from theory to practice, *Radiol.: Artif. Intell.* 7 (4) (2025) e240790.
- [22] C. Zakka, R. Shad, A. Chaurasia, A.R. Dalal, J.L. Kim, M. Moor, W. Hiesinger, Almanac—retrieval-augmented language models for clinical medicine, *Nejm AI* 1 (2) (2024) A10a2300068.
- [23] K. Soman, P.W. Rose, J.H. Morris, R.E. Akbas, B. Smith, B. Peetoom, S.E. Baranzini, Biomedical knowledge graph-optimized prompt generation for large language models, *Bioinformatics* 40 (9) (2024) btae560.
- [24] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, H. Wang, Retrieval-augmented generation for large language models: a survey. arXiv preprint arXiv:2312.10997, 2(1).
- [25] I. Melnyk, P. Dognin, P. Das, Knowledge graph generation from text, (2022). arXiv preprint arXiv:2211.10511.
- [26] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, W.T. Yih, Replug: retrieval-augmented black-box language models, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (volume 1: Long Papers), 2024, pp. 8371–8384.
- [27] D. Sanmartin, Kg-rag: Bridging the gap between knowledge and creativity, (2024). arXiv preprint arXiv:2405.12035.
- [28] Z. Tan, X. Zhao, W. Wang, Representation learning of large-scale knowledge graphs via entity feature combinations. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (2017) (pp. 1777–1786).
- [29] Y. Wan, Z. Chen, Y. Liu, C. Chen, M. Packianather, Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing, *Adv. Eng. Inf.* 65 (2025) 103212.
- [30] Y. Yang, J. Tang, L. Xia, X. Zou, Y. Liang, C. Huang, Graphagent: agentic graph language assistant, in: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 26360–26379.
- [31] L. Yang, Z. Huang, Z. Zhang, Z. Liu, S. Hong, W. Zhang, L. Zhang, Graphusion: latent diffusion for graph generation, *IEEE Trans. Knowl. Data Eng.* 36 (11) (2024) 6358–6369.
- [32] H. Chen, C. Xu, L. Zheng, Q. Zhang, X. Lin, Diffusion-based graph generative methods, *IEEE Trans. Knowl. Data Eng.* (2024).
- [33] J. Ngangmeni, D.B. Rawat, Swamped with too many articles? GraphRAG makes getting started easy, *AI* 6 (3) (2025) 47.
- [34] A. Yates, M. Banko, M. Broadhead, M.J. Cafarella, O. Etzioni, S. Soderland, Textrunner: open information extraction on the web, in: Proceedings of Human Language Technologies: the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 2007, pp. 25–26.
- [35] X. Liao, C. Chen, Z. Wang, Y. Liu, T. Wang, L. Cheng, Large language model assisted fine-grained knowledge graph construction for robotic fault diagnosis, *Adv. Eng. Inf.* 65 (2025) 103134.
- [36] D. Kim, L. Xie, C.S. Ong, Probabilistic knowledge graph construction: compositional and incremental approaches, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 2257–2262.
- [37] J. Baek, A.F. Aji, A. Saffari, Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, (2023). arXiv preprint arXiv:2306.04136.
- [38] X. He, Y. Tian, Y. Sun, N. Chawla, T. Laurent, Y. LeCun, B. Hooi, G-retriever: retrieval-augmented generation for textual graph understanding and question answering, *Adv. Neural Inf. Proces. Syst.* 37 (2024) 132876–132907.
- [39] J. Wang, H. Li, H. Wang, S.J. Pan, X. Xie, Trustworthy machine learning: robustness, generalization, and interpretability, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5827–5828.
- [40] C. Chen, T. Wang, Y. Zheng, Y. Liu, H. Xie, J. Deng, L. Cheng, Reinforcement learning-based distant supervision relation extraction for fault diagnosis knowledge graph construction under industry 4.0, *Adv. Eng. Inf.* 55 (2023) 101900.
- [41] X. Shen, X. Li, B. Zhou, Y. Jiang, J. Bao, Dynamic knowledge modeling and fusion method for custom apparel production process based on knowledge graph, *Adv. Eng. Inf.* 55 (2023) 101880.
- [42] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, J. Larson, From local to global: a graph rag approach to query-focused summarization, (2024). arXiv preprint arXiv:2404.16130.
- [43] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing, *ACM Comput. Surv.* 55 (9) (2023) 1–35.
- [44] K. Jeon, G. Lee, Hybrid large language model approach for prompt and sensitive defect management: a comparative analysis of hybrid, non-hybrid, and GraphRAG approaches, *Adv. Eng. Inf.* 64 (2025) 103076.
- [45] Z. Guo, L. Xia, Y. Yu, T. Ao, C. Huang, Lightrag: simple and fast retrieval-augmented generation, (2024). arXiv preprint arXiv:2410.05779.
- [46] Z. Xiang, C. Wu, Q. Zhang, S. Chen, Z. Hong, X. Huang, J. Su, When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation, (2025). arXiv preprint arXiv:2506.05690.
- [47] Y. Wang, N. Lipka, R.A. Rossi, A. Siu, R. Zhang, T. Derr, Knowledge graph prompting for multi-document question answering. In Proceedings of the AAAI conference on artificial intelligence (2024) (Vol. 38, No. 17, pp. 19206–19214).
- [48] T. Zhang, S.G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, J.E. Gonzalez, Raft: adapting language model to domain specific rag, (2024). arXiv preprint arXiv:2403.10131.
- [49] R. Liu, H. Jiang, X. Yan, B. Tang, J. Li, PolyG: effective and efficient GraphRAG with adaptive graph traversal (2025). arXiv preprint arXiv:2504.02112.
- [50] S. Knollmeyer, O. Caymazer, D. Grossmann, Document GraphRAG: knowledge graph enhanced retrieval augmented generation for document question answering within the manufacturing domain, *Electronics* 14 (11) (2025) 2102.
- [51] L. Bahr, C. Wehner, J. Wewerka, J. Bittencourt, U. Schmid, R. Daub, Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis, *J. Ind. Inf. Integr.* 45 (2025) 100807.
- [52] J.A.H. Alvaro, J.G. Barreda, An advanced retrieval-augmented generation system for manufacturing quality control, *Adv. Eng. Inf.* 64 (2025) 103007.
- [53] A. Abane, A. Bekri, A. Battou, S. Bensalem, Fastrag: Retrieval augmented generation for semi-structured data, (2024). arXiv preprint arXiv:2411.13773.
- [54] Z.T. Sworna, Z. Mousavi, M.A. Babar, NLP methods in host-based intrusion detection systems: a systematic review and future directions, *J. Netw. Comput. Appl.* 220 (2023) 103761.
- [55] J. Wu, J. Zhu, Y. Qi, J. Chen, M. Xu, F. Menolascina, V. Grau, Medical graph rag: towards safe medical large language model via graph retrieval-augmented generation, (2024). arXiv preprint arXiv:2408.04187.
- [56] H. Naganawa, E. Hirata, Enhancing policy generation with GraphRAG and youtube data: a logistics case study, *Electronics* 14 (7) (2025) 1241.
- [57] D. Yuan, H. Zhou, D. Wu, X. Liu, H. Chen, Y. Xin, J.C. Zhang, Enhancing large language models (llms) for telecommunications using knowledge graphs and retrieval-augmented generation, in: 2025 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2025, pp. 486–491.
- [58] J. Mao, C. Zheng, W. Liu, Z. Zhang, MGRAG: multimodal grid-aware retrieval augmentation generation framework for power grid work tickets, *Pattern Recogn.* 111845 (2025).
- [59] F. Bernier, J. Cao, M. Cordy, S. Ghamizi, SafePowerGraph-LLM: novel power grid graph embedding and optimization with large language models, (2025). arXiv preprint arXiv:2501.07639.
- [60] J. Shi, Y. Cheng, F. Zhang, M. Jiang, J. Lin, Y. Shen, GridCodex: a RAG-driven AI framework for power grid code reasoning and compliance, (2025). arXiv preprint arXiv:2508.12682.
- [61] I. Jendoubi, F. Bouffard, Data-driven sustainable distributed energy resources' control based on multi-agent deep reinforcement learning, *Sustainable Energy Grids Networks* 32 (2022) 100919.
- [62] J. Berant, A. Chou, R. Frostig, et al., Semantic parsing on freebase from question-answer pairs[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2013: 1533–1544.
- [63] A. Borde, N. Usunier, S. Chopra, et al., Large-scale simple question answering with memory networks[EB/OL]. (2015). <https://arxiv.org/abs/1506.02075>.
- [64] M. Dubey, D. Banerjee, A. Abdelkawi, et al., LC-QuAD 2.0: a large dataset for complex question answering over Wikidata and DBpedia, in: International Semantic Web Conference, Springer, 2019, pp. 69–78.
- [65] Y. Gu, S. Kase, M. Vanni, et al., Beyond iid: three levels of generalization for question answering on knowledge bases[C], in: Proceedings of the Web Conference

- 2021, International World Wide Web Conferences Steering Committee, 2021, pp. 3477–3488.
- [66] L. Jiang, R. Usbeck, Knowledge graph question answering datasets and their generalizability: are they enough for future research?, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 3209–3218.
- [67] P. Cimiano, H. Paulheim, Knowledge graph refinement: a survey of approaches and evaluation methods[J], *Semantic Web* 8 (3) (2016) 489–508.
- [68] J. Pujara, E. Augustine, L. Getoor, Sparsity and noise: where knowledge graph embeddings fall short, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1751–1756.
- [69] G. Ji, K. Liu, S. He, et al., Knowledge graph completion with adaptive sparse transfer matrix. Proceedings of the AAAI Conference on Artificial Intelligence, AAAI Press, 2016.