

A Synergized LLM-KG Framework for Domain Knowledge Cognition

Ruiyan Zheng¹ and Jinying Xu²

^{1,2}Department of the Built Environment, National University of Singapore, Singapore
zhengruiyan@u.nus.edu, jinying.xu@nus.edu.sg

Abstract –

Large language models (LLMs) evolve rapidly, yet their probabilistic generation is inherently non-traceable and leaves a persistent bottleneck in domain knowledge cognition. Retrieval augmentation via texts or knowledge-graphs (KGs) improves coverage and factual accuracy, but existing studies still depend largely on static knowledge structures and manually governed updating, so the obstacle to sustained and reliable domain knowledge cognition remains. This study therefore proposes a synergized LLM–KG framework for domain knowledge, centered on ontology-governed KG automatic generation (AutoGen) and update, and hybrid vector–graph retrieval for clause-grounded reasoning. Specifically, guided by a manually developed ontology, the KG is automatically constructed, incrementally enriched, and jointly used with vector retrieval to ground each query against both textual evidence and graph structure. A case study is conducted in construction contract management. Results demonstrate the robust KG AutoGen performance with average F1 scores of 0.904 for entity extraction and 0.856 for triple extraction. Across three query stages, the framework further improves multi-hop reasoning and clause-grounded accuracy, with its F1 advantage over GraphRAG widening from 0.00 to 0.37 and that over LLM-only prompting increasing from 0.25 to 0.45. This study provides a traceable route to deeper, more reliable, and continually refined domain knowledge cognition.

Keywords –

LLM; Knowledge graph; Domain knowledge; Knowledge cognition

1 Introduction

Recent advances in large language models (LLMs) have remarkably improved machines' natural language processing performance even in general but complex contexts. However, when application scenarios shift from general-purpose settings to domain-specific

environments, the performance is no longer determined solely by interpretative and generative ability [1]. Rather, it fundamentally depends on domain knowledge cognition, defined as the processes by which input information is transformed, processed, stored, retrieved, and used [2]. Despite their strong parametric memory, LLMs remain susceptible to hallucinations due to their probabilistic generative nature, so factual reliability remains a concern in specialized domain knowledge. More importantly, LLMs do not naturally convert unstructured evidence into reusable, explicit knowledge representations, and dependable domain deployment remains difficult [3].

As structured representations of entities and their relationships, knowledge graphs (KGs) are widely regarded as a promising approach to domain knowledge cognition due to their support for logical reasoning and cross-task reuse [4]. Nevertheless, constructing and maintaining high-quality KGs remains costly, and practical deployments often suffer from limited coverage and delayed updates, and high maintenance effort. To address these limitations, increasing attention has been devoted to integrating LLM and KG. Existing studies largely fall into two categories: (i) LLM for KG, where LLMs are leveraged to accelerate entity alignment and relation extraction for automated KG construction; and (ii) KG for LLM, where KGs serve as external knowledge sources that enhance LLM reasoning through approaches such as Graph-based Retrieval-Augmented Generation (GraphRAG) and Think-on-Graph (ToG) [5]. Despite progress along both directions, most existing work focuses on isolated stages of the cognitive process. LLMs are typically used for one-off triple extraction and KG construction, whereas KGs primarily serve as static or manually updated stores for retrieval. As domain evidence is often ambiguous and context-dependent, there is an increasing need for a synergized LLM–KG framework that leverages their complementarity to update domain cognition through query-driven interaction, in a manner akin to human experts [6].

To address this gap, this study proposes a synergized LLM–KG framework tailored for domain knowledge cognition, which integrates ontology-governed KG

AutoGen and update with hybrid retrieval-augmented reasoning. Departing from conventional functionally isolated designs, this framework aims to construct a continuously improving cognitive system that synergizes KG and LLM. Specifically, the domain ontology is first manually established as the governing schema and representational boundary for domain knowledge. Building on this, an LLM-driven and ontology-governed KG AutoGen pipeline is developed to support domain KG initialization and incremental update. At the query stage, the framework can acquire relevant external resources online, update the vector database (DB), and incorporate domain knowledge via KG AutoGen into KG DB. More importantly, it introduces a hybrid reasoning mechanism, in which the ontology schema, vector DB, and KG DB are jointly leveraged to ground evidence and strengthen reasoning capability.

A construction contract management case study is conducted to evaluate the proposed framework. Using the Public Sector Standard Conditions of Contract for Construction Works 2020 (PSSCOC CW 2020) in Singapore as the source corpus, this study excerpts 3 subparts for the comparative experiment. To ensure a fair comparison with baseline methods, online search is excluded. Instead, the first subpart is used for initialization, while the remaining two are introduced sequentially in two query-driven update stages. The evaluation focuses on two aspects: the accuracy and stability of KG AutoGen, assessed against manually annotated entity- and triple-level gold standards for each subpart; and reasoning performance, evaluated through a 15-question hierarchical benchmark with manually defined gold-standard references, in comparison with LLM-only prompting and GraphRAG under the same information setting at each stage. After systematic examination, the results demonstrate both the reliability of ontology-governed KG construction and update, and the advantage of hybrid retrieval in improving multi-hop reasoning and clause-grounded answering during query stage.

The remainder of this paper is organized as follows. Section 2 reviews related work on LLM-KG integration for domain knowledge cognition. Section 3 presents the proposed methodology. Section 4 describes the construction contract management case study and the corresponding experimental evaluation. Section 5 concludes the paper and discusses future work.

2 Related work on domain knowledge augmentation for LLM

Among existing domain knowledge augmentation research for LLM, domain knowledge cognition can be treated as an operational capability that supports (i) perception of domain evidence, (ii) memory organization

and reliable recall, (iii) reasoning over constraints and relational structure, and (iv) decision-oriented use of the derived knowledge [7]. Building on this, existing approaches can be positioned into four representative paradigms: prompting engineering, Naive RAG, Graph RAG, and synergized LLM-KG, as shown in Figure 1.

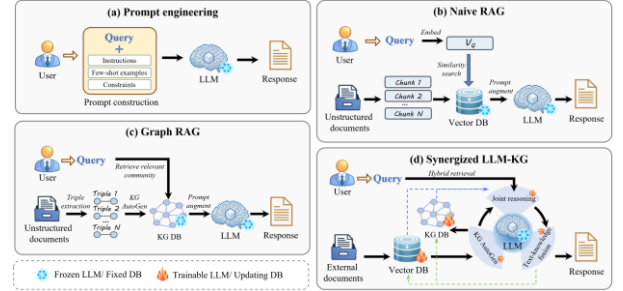


Figure 1. Comparison of Prompt engineering, Naive RAG, Graph RAG and Synergized LLM-KG.

(1) Prompt engineering

Prompt-based methods treat cognition primarily as an in-model process, with manual prompt construction based on scaffolds, including few-shot exemplars and structured reasoning instructions, to elicit task behavior. Their main advantage is low cost and rapid adaptation, since no external memory layer is required. However, domain knowledge remains inadequate, largely implicit and non-auditable, its reliability depends heavily on manual prompt design and the model itself, and systematic provenance tracking is typically unavailable.

(2) Naive RAG

Naive RAG grounds cognition in external text source materials, i.e., vector database (Vector DB), utilizing retrieval mechanisms as a proxy for memory recall. While this approach effectively mitigates hallucinations and supports efficient retrieval over large document collections, it relies on unstructured, fragmented text chunks. Consequently, the system often lacks the connectivity required for cross-document consistency, struggling to reconstruct the complex causal chains necessary for robust knowledge reasoning [8].

(3) Graph RAG

Exemplified by approaches like GraphRAG and ToG, this paradigm introduces a structured memory in the form of a KG to act as a semantic index. By retrieving entities, navigating relational paths, and aggregating neighborhood contexts alongside text, these methods significantly enhance precision and support for compositional queries. While this improves explainability and reasoning depth, conventional pipelines often treat the KGs as static DBs with high maintenance costs, lacking a mechanism for dynamic, user-driven updates to the knowledge state.

(4) Synergized LLM–KG

This paradigm integrates LLMs and KGs as mutually complementary components in domain knowledge cognition. The KG serves as structured domain memory that preserves entities, relations, and constraints, while the LLM supports extraction, retrieval orchestration, and reasoning over the grounded evidence. Although prior studies have demonstrated the promise of combining LLMs with structured knowledge, most focus on either KG-assisted reasoning or LLM-assisted KG construction in isolation. Much less attention has been given to ontology-governed KG automatic generation and incremental update under newly introduced evidence [6]. Consequently, existing synergy remains largely static at the retrieval stage, rather than extending to controlled knowledge expansion.

Motivated by these observations, the unresolved issue is not simply how to retrieve more evidence, but how to support a continuous domain cognition process in which knowledge can be structured, queried, verified, and selectively updated. This study accordingly tries to address this through two coupled capabilities: ontology-governed KG AutoGen and update, and hybrid vector-graph retrieval for clause-grounded reasoning.

3 Methodology

To operationalize the proposed paradigm, this study develops a synergized LLM-KG framework to support domain knowledge cognition, as illustrated in Figure 2. The framework comprises three parts: document parsing, ontology-governed KG AutoGen and update, and QA with hybrid reasoning. Operationally, the framework starts from a parsed corpus and a manually prepared ontology. The domain ontology is manually developed to provide a consistent schema of concepts, relations, and constraints. Document parsing then converts raw documents into a cleaned corpus and a vector database for retrieval. Governed by the ontology, KG AutoGen transforms extracted evidence into ontology-aligned entities and triples, which are consolidated into the KG as structured domain memory and incrementally updated when new evidence is introduced. At the query stage, the QA module jointly exploits the ontology schema, vector DB, and KG DB to support clause-grounded reasoning, with agentic external search activated when internal evidence is insufficient.

3.1 Document parsing

The process begins with document parsing, where domain documents are uploaded as raw sources. The system first applies structural segmentation to preserve the original information organization, such as section boundaries and figure-related content, so later retrieval can return evidence in coherent units rather than

fragmented text. After segmentation, text cleaning and normalization will be conducted to reduce noise introduced by formatting and heterogeneous layouts. The cleaned text is then tokenized and stored in a vector DB, which serves as the retrieval substrate for later evidence access. In short, document parsing turns unstructured materials into a normalized corpus and a vector index, which later supports both ontology construction and evidence grounding in QA.

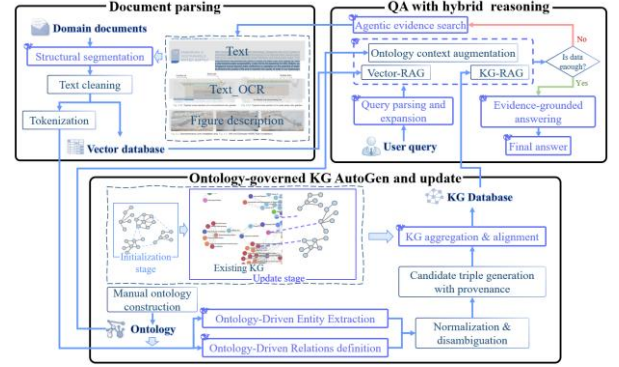


Figure 2. Research methodology.

3.2 Domain ontology manual development

Domain ontology is developed to provide a stable semantic boundary for KG AutoGen and update by defining the admissible classes, relation types, and constraints required for the target domain. The development process follows four steps: identifying core domain concepts from existing domain ontology (such as IfcOWL for BIM/IFC semantics; BOT for building structure; Brick for building systems) or authoritative domain sources (IFC for BIM data structure; bSDD for the built environment; OSHA 29 CFR 1926 for construction safety; FIDIC/NEC for construction contract); defining relation types and semantic constraints to capture domain logic; normalizing synonymous or ambiguous expressions into controlled labels and stable identifiers; and formalizing the resulting schema in RDF/OWL as a machine-readable governance layer. The ontology is then evaluated through competency questions to verify its adequacy for entity typing, relation extraction, and reasoning support. Once established, it serves as the schema boundary for KG extraction and validation, while the KG is incrementally constructed and updated under its governance.

3.3 KG AutoGen

The ontology specifies the normative schema and constraints, while the KG serves as the instance-level memory populated and governed under this schema for subsequent retrieval and reasoning. Given the ontology,

KG AutoGen therefore provides a systematic pathway to transform dispersed textual evidence into a structured and auditable KG. The ontology is maintained in RDF/OWL to provide a formally specified schema of concepts, relations, and constraints, and it functions as a constraint layer that steers extraction toward schema-consistent structures. Specifically, entities are extracted in an ontology-driven manner so that mentions are typed by predefined professional concepts rather than ad hoc labels, and relations are derived under the ontology’s relation schema to reduce structural drift and improve constraint consistency. To ensure global consistency across sources and iterations, normalization and disambiguation are performed so that variant surface forms resolve to stable identifiers.

During implementation, the operational KG is stored in Neo4j as a labeled property graph (LPG), enabling efficient subgraph operations and low-latency incremental updates through Cypher queries [9]. Candidate triples are generated with provenance, including source documents and evidence spans, which supports traceable verification before consolidation. To enforce ontology governance over an LPG-based KG, KG AutoGen applies a lightweight schema mapping and validation layer that projects ontology classes and properties to Neo4j labels, relationship types, and admissible property constraints, and uses these constraints to gate triple insertion. Importantly, the generation and consolidation behavior evolve between iterations. In the first run, no existing KG is available, and the newly generated triples are connected and organized to form an initial KG structure. In other runs, the system operates in an incremental mode: newly generated triples are aggregated and aligned against the existing KG, where entity matching, relation alignment, and duplicate resolution enable integration, refinement, and expansion of the graph. Finally, evaluation and validation filter out inconsistent or weakly supported triples before accepted updates are committed to the KG database. Through this iterative construction-and-alignment process, KG AutoGen establishes a validated memory substrate that the QA module can reliably retrieve and reason over, while also enabling controlled assimilation of domain knowledge over time.

3.4 QA system with hybrid reasoning

When user makes a query, the framework runs a QA pipeline that tightly couples synergized retrieval and structured reasoning. A user query is first parsed and expanded into retrieval-ready representations, including key entities, constraints, and decomposed sub-questions. Based on this structured interpretation, the system launches two complementary retrieval streams. The first stream follows a vector-led Naive RAG workflow that retrieves semantically relevant text segments from the

vector database. The second stream follows GraphRAG on the KG DB, where Cypher queries retrieve query-conditioned communities and construct a task-specific subgraph that captures relational structure and constraint dependencies that plain text retrieval does not expose. This study further augments the LLM with ontology text, which supplies schema and constraint information to enforce schema-consistent interpretation and reduce structurally invalid inferences. During the process, the LLM acts as an orchestrator that aligns text evidence with graph evidence, checks consistency against ontology constraints, and resolves mismatches between retrieved passages and KG assertions.

When internal evidence is insufficient or outdated, external evidence retrieval is activated. Agentic web search acquires supplementary sources, which are subsequently processed through document parsing and ontology-governed KG AutoGen to produce query-relevant structured evidence. This evidence is then cross-checked against the existing subgraph and ontology constraints before entering the reasoning pipeline. On this basis, the final answer is generated through evidence-grounded synthesis, with explicit reconciliation between text-derived evidence and graph-derived structure.

Reasoning is not treated as a terminal step when externally retrieved evidence is verified. Instead, the framework performs a controlled update, under which only qualified statements are written back to the KG DB. Specifically, a candidate statement is committed to the KG only when (i) its supporting evidence is traceable to concrete sources, (ii) it does not contradict existing KG facts or the evidence base, and (iii) it complies with the ontology-defined schema and constraints. Through this update process, each interaction not only produces an evidence-grounded response but also incrementally improves the knowledge base, increasing coverage and precision over repeated use while preventing unverified errors from being written back and amplified.

4 Case study

To evaluate the proposed framework, a construction contract management case study is conducted using the Public Sector Standard Conditions of Contract for Construction Works 2020 in Singapore (PSSCOC CW 2020). This corpus is well suited to the study because contract reasoning is clause-intensive, procedurally constrained, and inherently multi-hop: practical questions often require navigation across notices, causal events, entitlement conditions, payment procedures, and dispute routes.

4.1 Experimental design

Three subparts are excerpted from PSSCOC CW 2020. Subpart A covers time, claims, loss and expense,

and payment workflow; Subpart B adds instructions, contract documents, ambiguity, and delay causation; and Subpart C further adds variations, valuation, dispute routes, and service of notices. To ensure fair comparison, online search is excluded. The system is initialized with Subpart A only, then updated with Subpart B in the first growth stage and with Subpart C in the second growth stage. This design makes knowledge expansion explicit and allows the effects of KG update on reasoning depth to be examined stage by stage.

4.2 Experimental setting

The evaluation follows two tracks. First, KG AutoGen is assessed against manually annotated gold standards at the entity and triple levels for each subpart. The gold annotations cover 20 entities and 18 triples in Subpart A, 15 entities and 12 triples in Subpart B, and 15 entities and 12 triples in Subpart C. Second, reasoning performance is evaluated using a 15-question hierarchical benchmark with manually defined gold-standard clause references. The benchmark contains five Stage-1 questions over Subpart A only, five Stage-2 questions over Subparts A+B, and five Stage-3 questions over Subparts A+B+C. Their reasoning depth increases from single-chain 3-4 hop, to cross-subpart 5-6 hop, to multi-subpart 7-9 hop, while the number of gold-standard sections rises from 8-10 in Stage 1, to 12-14 in Stage 2, and to 18-24 in Stage 3. LLM-only prompting and GraphRAG are adopted as baselines, and all models receive the same information at each stage. Precision, recall, and F1 are reported for both KG AutoGen and reasoning experiments. Figure 4 illustrates the implemented interface for clause-grounded answering and graph inspection.

4.3 Results and discussion

For the construction contract management case, the domain ontology was manually developed as a contract-variant-specific governance layer for KG AutoGen and update. Rather than aiming to cover construction contracts in general, the ontology was bounded to the conceptual and procedural scope of the target corpus, namely the Public Sector Standard Conditions of Contract for Construction Works 2020 in Singapore (PSSCOC CW 2020), with emphasis on the three selected subparts used in the staged experiment: time extension, claims and payment workflow; instructions, document hierarchy, ambiguity, and delay causation; and variations, valuation, disputes, and notice service. Within this boundary, it specifies the main entity classes, relation types, and semantic constraints needed for clause-grounded reasoning, including actors such as Contractor and Superintending Officer, instruments such as notices and instructions, events such as delay and ambiguity, and

procedural objects such as claims, payment certificates, variations, valuation steps, and dispute routes. The ontology was formalized in RDF/OWL and validated through competency questions derived from the benchmark, such as whether the schema could represent the notice and substantiation chain for time and loss-and-expense claims, the contractual effectiveness of oral instructions, the resolution path for ambiguity-linked delay and cost, the variation-to-valuation-to-payment route, and the effect of service-of-notice rules on later dispute procedures.

Building on the ontology, KG AutoGen constructs the knowledge graph through staged initialization and incremental update. In the initialization stage, ontology-typed entities and schema-consistent relations are extracted from the initial contract corpus, transformed into provenance-backed triples, and consolidated into an initial KG. In later stages, when additional contract subparts are introduced, KG AutoGen performs incremental consolidation by aligning newly generated triples with the existing graph through entity matching, relation alignment, and duplicate resolution. In this way, the KG grows progressively with the expanding corpus while remaining governed by the same ontology-defined schema and constraints. Figure 3(a) visualizes the resulting graph at the final stage, and Figure 3(b) shows a representative clause-centered subgraph.

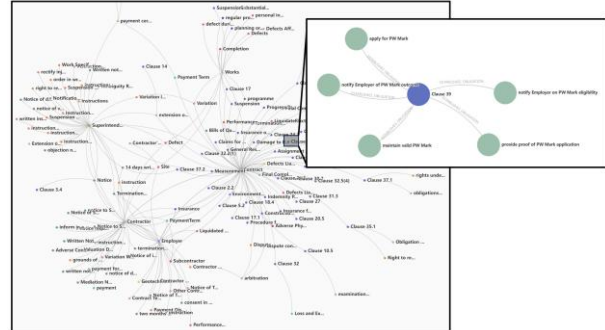


Figure 3. Visualization of (a) the 3rd update stage KG and (b) the detailed subgraph.

The framework is implemented as an end-to-end operational system, where LLM services via the DeepSeek API are used to orchestrate query interpretation, evidence alignment, and verification, while the KG is stored in Neo4j to support efficient subgraph retrieval and incremental updates. As illustrated in Figure 4, users interact with the system through a QA interface. For each query, the system retrieves both passage-level evidence from the vector database and structure-level evidence from the KG, constructs a query-conditioned subgraph, and generates an evidence-grounded answer with clause-linked support. In parallel, the interface exposes the retrieved related

knowledge as an interpretable graph view and triple list, enabling users to inspect the reasoning substrate rather than receiving a black-box response. For more details, please check the demo exhibition video from the website: <https://www.youtube.com/watch?v=IDWH5f-LsZE>.

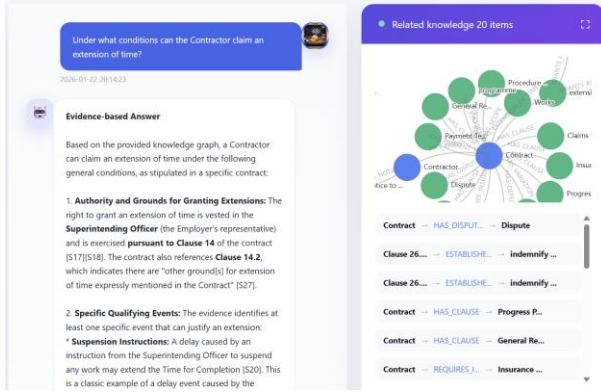


Figure 4. The synergized LLM-KG cognition system.

To further assess hybrid reasoning performance, the proposed framework is compared with two baselines: DeepSeek (LLM-only) and Neo4j GraphRAG (GraphRAG). To ensure a fair comparison, the LLM-only systems operate without access to the internal KG or any online search; the same base LLM and embedding model are used across GraphRAG and the proposed framework, while the difference lies in whether ontology-governed KG AutoGen, KG incremental update, and ontology-augmented retrieval are enabled. All methods are evaluated on the same set of contract management questions under the same staged information setting.

Tables 1 and 2 summarize KG AutoGen performance across the three subparts. Entity extraction remains strong across all stages, with F1 scores of 0.927, 0.903, and 0.882 for Subparts A, B, and C, respectively, yielding an average F1 of 0.904. Triple extraction is more demanding but still robust, with F1 scores of 0.889, 0.800, and 0.880 and an overall average of 0.856. These results show that ontology governance provides a stable extraction boundary not only during initialization but also during incremental update, even when later subparts introduce more cross-clause and procedural relations.

Table 1. Performance of machine annotated entity across subparts.

Subpart	Precision	Recall	F1
A	0.905	0.950	0.927
B	0.875	0.933	0.903
C	0.833	0.938	0.882

Table 2. Performance of machine annotated triple across subparts.

Subpart	Precision	Recall	F1
A	0.889	0.889	0.889
B	0.769	0.833	0.800
C	0.846	0.917	0.880

Table 3 shows that the reasoning advantage of the proposed framework becomes more evident as the knowledge state grows and the questions become more multi-hop. At initialization, the proposed method matches GraphRAG at F1 = 0.76 and already exceeds LLM-only prompting by 0.25. After the first update, its F1 rises to 0.79, outperforming GraphRAG (0.55) and LLM-only prompting (0.43). After the second update, the advantage widens further to 0.82 versus 0.45 for GraphRAG and 0.37 for LLM-only prompting. This trend indicates that the main benefit does not lie merely in storing more information, but in updating the KG under ontology governance and then retrieving evidence jointly from vector space and graph structure.

Table 3. Results of evaluation metrics for compared models.

Stage	LLM-only			GraphRAG			Ours		
	P	R	F1	P	R	F1	P	R	F1
Startup	0.42	0.66	0.51	0.72	0.81	0.76	0.72	0.82	0.76
Update 1	0.36	0.53	0.43	0.49	0.62	0.55	0.75	0.84	0.79
Update 2	0.31	0.46	0.37	0.39	0.52	0.45	0.78	0.86	0.82

In qualitative terms, LLM-only can produce fluent but weakly grounded answers, whereas GraphRAG improves structural retrieval but remains limited when the graph is incomplete and processing capability is limited at one time. By contrast, the proposed framework returns clause-linked answers together with an inspectable subgraph, making the reasoning path auditable. This is particularly important for contract management, where whether a claim succeeds often depends on the exact interaction among notice clauses, evidentiary requirements, and procedural escalation.

5 Conclusion

This study proposes a synergized LLM-KG framework for domain knowledge cognition, centered on ontology-governed KG automatic generation and update, and hybrid vector-graph retrieval for clause-grounded reasoning. Unlike conventional approaches that treat KG construction, updating, and querying as isolated functions, the proposed framework connects these stages within a unified workflow, in which structured knowledge can be automatically constructed from domain evidence, incrementally expanded under

ontology governance, and jointly exploited with vector retrieval during reasoning. In this way, the framework provides not only evidence-grounded answers, but also a traceable and continuously improvable knowledge base.

To ensure reliability, the framework uses a manually developed ontology as the schema boundary for entities and triples extraction and validation, thus KG is constructed and updated through provenance-backed and schema-consistent triples. At the query stage, textual evidence and graph structure are jointly retrieved to support clause-grounded reasoning. The construction contract management case study demonstrates that the proposed framework achieves robust KG AutoGen performance across initialization and update stages, while also improving multi-hop reasoning and clause-grounded answering over both LLM-only prompting and GraphRAG baselines. These results indicate that the main advantage lies not merely in storing more domain knowledge, but in governing KG growth through ontology constraints and integrating vector-based and graph-based evidence during reasoning.

Several limitations also remain. Future work will further validate the framework compared to broader advanced LLMs and retrieval frameworks with real-world dynamic document settings. Other important directions include more systematic handling of conflicting evidence, more principled criteria for ontology refinement, and broader evaluation of long-term knowledge growth under repeated updates. In addition, future research may explore closer integration between external evidence acquisition and internal KG consolidation, so that domain knowledge can be expanded in a more timely and trustworthy manner.

6 Acknowledgement

This research is funded by NUS Start-up Grant Project: “Digitally Empowered Project Cognition for Sustainability”.

References

- [1] Y. Wan, Y. Liu, C. Chen, M. Packianather, Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing, *Advanced Engineering Informatics* 65 (2025) 103212. <https://doi.org/10.1016/j.aei.2025.103212>.
- [2] U. Neisser, *Cognitive Psychology: Classic Edition*, Psychology Press, New York, 2014. <https://doi.org/10.4324/9781315736174>.
- [3] M. Li, Z. Wang, BuildingGPT: Query building semantic data using large language models and vector-graph retrieval-augmented generation, *Building and Environment* 287 (2026) 113855. <https://doi.org/10.1016/j.buildenv.2025.113855>.
- [4] B. Mo, K. Yu, J. Kazdan, J. Cabezas, P. Mpala, L. Yu, C. Cundy, C. Kanatsoulis, S. Koyejo, KGGen: Extracting Knowledge Graphs from Plain Text with Language Models, (2025). <https://doi.org/10.48550/arXiv.2502.09956>.
- [5] C. Zheng, S. Wong, X. Su, Y. Tang, A. Nawaz, M. Kassem, Automating construction contract review using knowledge graph-enhanced large language models, *Automation in Construction* 175 (2025) 106179. <https://doi.org/10.1016/j.autcon.2025.106179>.
- [6] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying Large Language Models and Knowledge Graphs: A Roadmap, *IEEE Trans. Knowl. Data Eng.* 36 (2024) 3580–3599. <https://doi.org/10.1109/TKDE.2024.3352100>.
- [7] G. Ciatto, A. Agiollo, M. Magnini, A. Omicini, Large language models as oracles for instantiating ontologies with domain-specific knowledge, *Knowledge-Based Systems* 310 (2025) 112940. <https://doi.org/10.1016/j.knosys.2024.112940>.
- [8] Q. Chen, X. Yin, B. Yuan, Q. Chen, Personalized safety training for construction workers: A large language model-driven multi-agent framework integrated with knowledge graph reasoning, *Computers in Industry* 174 (2026) 104399. <https://doi.org/10.1016/j.compind.2025.104399>.
- [9] D. Zhang, G. Ma, T. Qu, X. Wang, W. Zhou, X. Wang, A knowledge graph-enhanced large language model for question answering of hydraulic structure safety management, *Advanced Engineering Informatics* 66 (2025) 103468. <https://doi.org/10.1016/j.aei.2025.103468>.