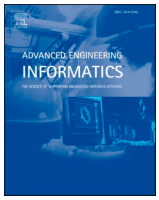
Advanced Engineering Informatics 65 (2025) 103212

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing☆

Yuwei Wan [a], Zheyuan Chen [b,c], Ying Liu [a,*], Chong Chen [d], Michael Packianather [a]

[a] Department of Mechanical Engineering, School of Engineering, Cardiff University, Cardiff CF24 3AA, UK
[b] Guangzhou Institute of Industrial Intelligence, Guangzhou 511458, China
[c] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[d] Guangdong Provincial Key Laboratory of Cyber-Physical System, Guangdong University of Technology, Guangzhou 510006, China

ARTICLE INFO

Keywords:
Smart manufacturing
Large language model
Retrieval-augmented generation
Q&A
Knowledge graph
Additive manufacturing
Design for additive manufacturing

ABSTRACT

Large language models (LLMs) have shown remarkable performances in generic question-answering (QA) but often suffer from domain gaps and outdated knowledge in smart manufacturing (SM). Retrieval-augmented generation (RAG) based on LLMs has emerged as a potential approach by incorporating an external knowledge base. However, conventional vector-based RAG delivers rapid responses but often returns contextually vague results, while knowledge graph (KG)-based methods offer structured relational reasoning at the expense of scalability and efficiency. To address these challenges, a hybrid KG-Vector RAG framework that systematically integrates structured KG metadata with unstructured vector retrieval is proposed. Firstly, a metadata-enriched KG was constructed from domain corpora by systematically extracting and indexing structured information to capture essential domain-specific relationships. Secondly, semantic alignment was achieved by injecting domain-specific constraints to refine and enhance the contextual relevance of the knowledge representations. Lastly, a layered hybrid retrieval strategy was employed that combined the explicit reasoning capabilities of the KG with the efficient search power of vector-based similarity methods, and the resulting outputs were integrated via prompt engineering to generate comprehensive, context-aware responses. Evaluated on design for additive manufacturing (DfAM) tasks, the proposed approach achieved 77.8% exact match accuracy and 76.5% context precision. This study establishes a new paradigm for industrial LLM systems, which demonstrates that hybrid symbolic-neural architectures can overcome the precision-scalability trade-off in mission-critical manufacturing applications. Experimental results indicated that integrating structured KG information with vector-based retrieval and prompt engineering can enhance retrieval accuracy, contextual relevance, and efficiency in LLM-based Q&A systems for SM.

1. Introduction

Recent advances in smart manufacturing (SM) have generated a large amount of complex, domain-specific data from production processes, sensor networks, and technical documentation. Despite the potential of this data to inform decision-making, conventional Q&A systems struggle to utilize the specialized knowledge required for accurate domain-centric responses with the following challenges. Firstly, although the accumulated data has great potential to inform decision-making [1], much of it remains underutilized since it is dispersed across various platforms and exists in heterogeneous formats [2]. Secondly, decision-making in SM is often limited by its dependence on predefined rules or heuristics, and therefore constraints on the flexibility of the Q&A system [3,4]. Lastly, the conventional expert system in SM often lacks deep reasoning capabilities. Therefore, these challenges limit the ability to handle intricate tasks that require logical inference and contextual understanding [5,6].

In this case, cognitive capabilities that integrate and understand the multi-sourced knowledge are essential for SM. In recent advancements, knowledge graphs (KG) and large language models (LLMs) have demonstrated their potential. LLMs excel at interpreting natural language, processing unstructured data, and generating coherent and context-aware responses. Despite their generative and reasoning strengths, LLMs still face limitations in specialized domains, such as

☆ This article is part of a special issue entitled: 'LLM&KG in ProdDes&Mfg' published in Advanced Engineering Informatics.

* Corresponding author.
E-mail address: LiuY81@Cardiff.ac.uk (Y. Liu).

https://doi.org/10.1016/j.aei.2025.103212
Received 8 January 2025; Received in revised form 12 February 2025; Accepted 13 February 2025
Available online 22 February 2025

producing plausible but incorrect responses (referred to as "hallucinations") [7], lacking the latest developments or events beyond their training data [8], failing to properly attribute sources of information [9], and lacking the depth of professional knowledge [10]. Meanwhile, adapting general-purpose LLMs to specific domains through fine-tuning is a widely common approach. However, the fine-tuning method requires substantial resources, including large datasets, significant computational capabilities, and considerable time investment, making it unfeasible for many domain-centric applications. On the other hand, KG offers a structured representation of the specialized knowledge and enables semantic understanding and logical reasoning across interconnected data. In this context, KG can provide the structured domain knowledge necessary for semantically accurate retrieval to compensate for the meticulous and accurate understanding of specialized knowledge required by LLMs [11].

To leverage the complementary strengths of KGs and LLMs, retrieval-augmented generation (RAG) is a promising way to enhance domain-centric Q&A systems in SM. RAG models combine the strengths of retrieval-based methods and generative models to produce contextually accurate and semantically rich outputs [12]. The vast external knowledge sources encapsulated within the KGs can be further accessed and updated dynamically, which is useful when the generative model itself might not contain all the necessary information. Despite their potential, current RAG methods present individual challenges. Commonly, the RAG methods can be classified into vector-based RAG and graph-based RAG [13]. Vector-based retrieval methods calculate semantic similarity between query and document embeddings, which enables them to efficiently retrieve generally related content. However, since these embeddings are typically derived from models pre-trained on broad corpora, they may not capture the specialized vocabulary and fine-grained relationships inherent to SM. As a consequence, although such methods excel in identifying broadly relevant information, they might overlook subtle, domain-specific distinctions that are crucial for accurate technical reasoning. In contrast, by utilizing structured representations such as explicit entity relationships, graph-based approaches can provide precise contextual grounding. However, they often struggle with the variability and unstructured nature of real-world data. Consequently, while each approach has its distinct strengths, neither is sufficient on its own to meet the intricate requirements of domain-centric Q&A systems in SM.

To address these challenges, this study proposes a hybrid KG-Vector RAG approach that combines the strengths of both vector-based retrieval and graph-based reasoning. The proposed hybrid KG-Vector approach bridges the gap between vector similarity and knowledge relevance to enhance the accuracy, relevance, and interpretability of generated responses for intelligent and efficient SM practices. The main theoretical contributions of this study are: (1) a hybrid model, that jointly optimizes structured KG metadata and unstructured vector embeddings, is proposed, (2) a constraint-driven semantic alignment approach to enforce domain-specific logic during retrieval, and (3) a modular KG construction workflow for scalable knowledge integration. In this context, the hybrid approach is proposed to address the precision-scalability trade-off. Moreover, this Q&A system further lowers the barrier to entry for professionals and non-specialists and democratises access to expert-level guidance and insights without requiring extensive background knowledge or specialised training.

The remainder of this study is organized as follows. Section 2 reviews the related works, including domain Q&A in manufacturing, KG and its applications in manufacturing, LLMs, prompt engineering, and RAG. In Section 3, we illustrate the proposed methodology in three aspects: KG construction, semantic alignment, and the proposed hybrid RAG model. Section 4 is the experimental setup. Section 5 includes a case study to demonstrate the practical application and effectiveness of the proposed KG-Vector RAG approach, and details results and discussion. Section 6 concludes this study.

2. Literature review

2.1. Domain Q&A in smart manufacturing

Domain-specific Q&A is designed to retrieve and provide precise answers tailored to the specific needs of a domain. In SM scenarios, Q&A enables quick access to information, supports complex decision-making, and facilitates operations such as production monitoring, fault diagnosis, and supply chain optimization. Unlike general Q&A, domain-specific Q&A is built upon specialized knowledge and is essential for addressing the intricate challenges of SM environments. For example, manufacturing involves a combination of structured and unstructured data, including machine logs, production schedules, and operator manuals [14]. Domain Q&A help professionals quickly retrieve relevant insights without manually filtering through large volumes of information. Conventional Q&A in SM typically involve rule-based methods, information retrieval techniques, or machine learning (ML)-based models. While effective in specific scenarios, these conventional approaches have inherent limitations. Rule-based methods operate based on predefined rules and logic created by experts. They are simple and effective for answering specific types of questions but lack flexibility and struggle to handle new or unexpected queries [15,16]. Information retrieval techniques focus on finding and retrieving relevant documents or text passages based on a query. While they can locate useful information, they often do not provide direct answers or interpret the context of the query [16]. ML-based models in Q&A use algorithms trained on historical data to identify patterns and generate answers, including traditional ML-based models [17] and deep learning-based models [18]. However, they usually lack contextual reasoning capabilities and cannot fully understand the complex relationships in domain-specific knowledge. Despite their potential, conventional Q&A face several challenges in SM, such as handling unstructured data, deep and contextual reasoning, and real-time demands. For instance, manufacturing data often lacks uniformity and is stored in diverse formats such as free text, images, and sensor readings [19]. Extracting meaningful insights from such heterogeneous data remains a significant challenge. Queries in SM require a deep understanding of domain-specific context [14]. Moreover, SM environments are highly dynamic, which requires Q&A systems that can scale to handle growing data volumes and respond to real-time queries with accuracy [20].

As technologies advance, the capabilities of KG and LLMs have been increasingly recognized and demonstrated in SM environments. KG provides a structured way to represent and organize domain knowledge in a unified format, which enables machines to understand and reason over interconnected data. By capturing relationships between entities (e.g., machines, processes, and materials) in a graph format, KGs allow Q&A to answer queries with contextually relevant and accurate information. For example, for fault diagnosis scenarios of hot-rolling lines in the steel industry, a KG can link machine maintenance records with fault diagnosis data, enabling the Q&A system to suggest tailored solutions based on historical patterns and online failure datasets [19]. KGs also support semantic reasoning, which helps Q&A systems interpret complex and multi-faceted queries effectively. Additionally, LLMs excel at understanding natural language and generating contextually coherent-appropriate responses. The capability of LLMs to perform deep reasoning allows them to handle ambiguous or complex queries by leveraging pre-trained knowledge across diverse topics and drawing logical inferences. By interpreting nuanced user queries in SM environments, LLMs can enhance Q&A systems to generate detailed explanations and adapt responses to the specific needs of the domains. In this context, combining KGs with LLMs opens new possibilities for overcoming the challenges faced by conventional Q&A and improves the accuracy and relevance of domain Q&A in SM. LLMs can interact with structured data sources, such as KGs, to provide enriched and context-aware answers.

2.2. KG and its applications in smart manufacturing

A KG is a structured representation of knowledge that captures entities, relationships, and attributes in a graph format. It provides a flexible and machine-readable framework to organise and reason over complicated data in a semantic layer. In the context of SM, the manufacturing field has accumulated extensive data and knowledge from sensor readings and machine logs to production schedules and maintenance records [1]. Although the accumulated data holds significant potential for improving manufacturing, the data is often complex and large-scale, which presents certain challenges in terms of storage, management, and analysis. Moreover, the data is often scattered across different systems, rich in semantic information, heterogeneous in format, and multimodal in nature, including structured data (e.g., tabular data), unstructured data (e.g., text, images), and semi-structured data (e.g., JSON, XML). The conventional data fusion approaches directly concatenate feature vectors to fuse different facets, not considering varying distance metrics across facet boundaries. In this case, one fundamental challenge in fusing disparate facets lies in bridging the semantic gaps among them [21]. The diversity and fragmentation of data further make it difficult to extract meaningful insights.

In this case, KG has been identified as a useful tool to deal with these challenges, which can be opportunities for the advancement of SM. When the data and knowledge generated from various sources are accumulated and properly used, this data can provide invaluable insights regarding SM systems [22]. KG serves as a powerful tool to integrate heterogeneous data sources, facilitate semantic understanding, and enable advanced analytics across SM systems [23]. For instance, KG offers a way to map and interpret the data from multiple sources with rich semantic information for a comprehensive view of the SM environment, regardless of its format or structure, which ensures that potential insights are not lost [24]. Manufacturers can deploy KGs to organize and represent the vast amount of data, which can be used to derive more informed decision-making processes [25]. Meanwhile, by representing data in a graph-based structure, KGs provide more intuitive querying and exploration of knowledge to reveal more complex semantic relationships that might be missed using traditional ontology-based approaches. KGs can be further continuously updated and expanded as new data becomes available, making them well-suited for the dynamic nature of SM. Based on the integration and synthesis of multiple data sources, the mass personalization faced by SM can be achieved to meet individual customer preferences [26]. KG also offers a platform to enhance collaborative interactions between humans and machines with a flexible and machine-readable framework [27]. In addition, a structured and comprehensive knowledge representation represented by KG improves the cognitive aspects of SM, such as self-learning and self-adapting systems [28]. In sum, SM presents numerous challenges that can be addressed by adopting KG-based approaches.

2.3. Large language models and prompt engineering

LLMs are designed to process and generate human language at an advanced level. They have a large number of parameters, often in the billions. Trained on vast amounts of data from various sources, LLMs have developed the ability to understand complex language patterns, including grammar, context, and meaning, and achieve high performances in many aspects, such as comprehending and generating natural language text and providing valuable information and solutions for specific tasks [29]. One of the strengths of LLMs is their ability to generalize across different tasks because of their training on diverse datasets [30]. The generalization allows them to perform well in a variety of applications. Moreover, as LLMs increase in size, their performance improves, making scalability a key advantage of LLMs. However, LLMs also face several challenges. They are resource-intensive by requiring significant computational power and memory, which makes them costly to train and deploy [31]. Additionally, LLMs can inherit biases from their training data and lead to biased outputs, which is a concern in many applications [32]. As known as 'hallucinations', another issue is that LLMs sometimes produce outputs that sound plausible but are factually incorrect or nonsensical [33].

The applications of LLMs are wide-ranging and are used extensively in NLP tasks, such as sentiment analysis, conversational AI, content creation and machine translation. Especially in Q&A systems, LLMs are valuable for understanding and responding to user queries to provide relevant information in a coherent manner [34]. In the context of SM, LLMs hold great potential for enhancing knowledge representation and management. LLMs can be integrated with KGs to improve the retrieval and use of domain-specific knowledge [35]. By translating user queries into structured queries, LLMs help make the interaction with KGs more efficient, leading to better decision-making in manufacturing processes [35].

As a necessary technique to interact with LLMs, prompt engineering is an essential technique for guiding LLMs to produce accurate and relevant outputs in various tasks, where the construction of prompts plays an important role in leveraging the capabilities of LLMs [36]. LLMs perform better when provided with sufficient context and detailed instructions within the prompt. Commonly, prompt engineering contains the design and optimization of prompts (the input of LLMs) to guide LLMs towards the desired output, which is a practice in knowledge engineering [37]. The first and most fundamental strategy in prompt engineering is the clear definition of the task at hand. When interacting with LLMs, ambiguous or poorly defined prompts can lead to irrelevant or incorrect responses. To ensure accurate and meaningful outputs, the prompt must explicitly convey the objective. For instance, in the context of SM, if the task is to generate maintenance instructions based on historical machine performance data, the prompt should specify the data type, machine, and required action clearly [38]. One of the key strategies in prompt engineering is the use of few-shot and one-shot learning, where the model is provided with one or more examples of the desired output format or task structure [39]. By showing the model examples of what a correct response looks like, it can better align its future outputs with user expectations. It is particularly useful in SM where specific query formats or technical jargon are used, such as generating production schedules. Another effective strategy is the use of constraints within prompts to limit the scope of the model's output. The constraints can include setting word limits, specifying the format (e.g., bullet points, tables), or defining the depth of detail required. In SM, constraints ensure that the response is clear, structured, and actionable, rather than being overly verbose or generalized.

Moreover, prompt engineering is often an iterative process, where prompts are refined based on the LLM's responses [40]. The iterative process is crucial in SM, where the complexity of data and the precision required for operational decisions demand high-quality outputs. The initial prompt might produce suboptimal results, which require adjustments to the phrasing, structure, or context. Iteratively refining the prompt helps improve the accuracy and relevance of the output. However, one challenge in prompt engineering is dealing with the inherent ambiguity of natural language [41]. If prompts are not carefully designed, LLMs can generate responses that reflect misinterpretations or biases.

2.4. Fundamentals of retrieval-augmented generation

Although LLMs perform well in processing and generating human language at an advanced level, they can sometimes generate responses that are not fully accurate or grounded in real-world facts [42]. It is the reason why RAG stands out—by using the information retrieved in the first step, the generative model produces a response that combines fluency with factual accuracy. RAG is a model architecture that enhances the capabilities of generative models by integrating retrieval-

based techniques [43]. It combines the strength of retrieval systems, which can search for relevant information from large datasets or knowledge bases, with the language fluency of generative models, such as LLMs. At the core of the RAG model is the retrieval component, which plays an important role in sourcing relevant information from a knowledge base [44]. When a user inputs a query, the model first retrieves relevant documents, facts, or data points from an external source. The external source could be a KG, a document repository, or any large corpus of structured or unstructured data. The retrieved information provides a foundation for the generative model to produce more accurate and contextually relevant responses. Once relevant information is retrieved, the generative component comes into play. A key feature of RAG is the tight integration between retrieval and generation. Instead of relying solely on pre-trained knowledge stored within the model's parameters, RAG retrieves external information dynamically and incorporates it into the response generation process [45]. It ensures that the model is not constrained by outdated or incomplete information, which is a common limitation in independent generative models. In SM, the combination of retrieval and generation enhances the relevance and utility of the output in complex and data-rich environments. Moreover, the capability is particularly valuable in SM, where new data is continuously generated, and decision-making often depends on the most current information available.

In RAG, the ability to combine generative models with retrieval systems results in several key benefits. Firstly, since the model does not need to store all knowledge within its parameters, it can scale more easily. Secondly, the retrieval component searches through large corpora, KGs, or other databases, making RAG models adaptable to vast knowledge domains. Thirdly, in dynamic environments, such as SM scenarios, where situations evolve rapidly, RAG models provide responses that are contextually relevant and tailored to the latest data inputs. Therefore, RAG offers a powerful solution to many of the limitations faced by standalone generative models [45,46]. By integrating retrieval systems with LLMs, RAG combines the ability to retrieve accurate, real-time information with the fluency and reasoning capabilities of generative models. It makes RAG well-suited for knowledge-intensive environments, where accurate and data-driven decisions are crucial. By retrieving and utilizing relevant data from KGs and other external sources, RAG models help ensure that generated responses are linguistically coherent and actually grounded and contextually relevant. In the context of SM, manufacturing processes and systems generate large amounts of data, including sensor readings, maintenance logs, and performance records, which are essential for operational efficiency. By retrieving this data in real-time and combining it with the reasoning capabilities of LLMs, RAG can generate insights that are both actionable and grounded in factual data.

Despite its advantages, RAG also faces a certain challenge, where integrating both retrieval and generation adds complexity to the model [46]. The retrieval component should be fine-tuned to ensure it pulls the most relevant information, while the generative model should accurately incorporate this information into coherent and meaningful responses. The additional step of retrieving information before generating a response can increase the time it takes for the model to produce an output. In real-time applications, optimizing the speed of the retrieval process is crucial. Moreover, the success of RAG depends heavily on the quality and relevance of the information retrieved. If the retriever pulls irrelevant or outdated data, it can negatively impact the quality of the generated response. Therefore, the underlying knowledge base, such as a well-maintained KG, plays a critical role in ensuring the effectiveness of RAG.

3. The proposed hybrid RAG approach

As reviewed in the previous section, vector-based RAG methods in SM lack precision due to unstructured retrieval, while KG-based methods struggle with scalability and dynamic updates. As illustrated in Fig. 1, to bridge this gap, this study proposes the hybrid KG-Vector RAG approach which integrates KGs and LLMs through a hybrid RAG to facilitate domain Q&A in SM. Firstly, a domain-specific corpus was curated from technical publications and industry documents. From this corpus, key metadata was systematically extracted and used to construct a comprehensive KG in Neo4j, capturing essential entities and relationships. Simultaneously, the same corpus was segmented into text chunks that were converted into vector embeddings using a standard embedding model, thereby establishing an unstructured vector store for the following semantic similarity searches. Subsequently, semantic alignment is employed by incorporating domain-specific terminology and applying clear constraints to ensure consistent interpretation and retrieval of relevant design principles. Finally, a hybrid RAG with vector-based and KG-driven indexing is proposed to leverage both unstructured textual sources and structured knowledge to provide insights for question answering. Moreover, the study design explicitly aimed to compare three retrieval strategies (vector-only, KG-only, and a hybrid KG-Vector approach) to determine how best to integrate structured and unstructured data for precise Q&A. Compared with using general LLMs in SM scenarios, the proposed approach can effectively look into both structured explicit knowledge and implicit instructed information by employing this hybrid RAG approach.

3.1. Knowledge and human effort collection for KG construction

To ensure that the RAG pipeline is specifically tailored for SM, domain-centric documents are first collected. As shown in Fig. 2, a core set of specific documents and publications are selected by domain experts. These documents represent the essential knowledge corpus for SM domain-specific question answering, including materials, processes, design heuristics, sustainability considerations, and performance metrics. Once collected, the text data is pre-processed by eliminating noise and standardizing formats to ensure consistency across the corpus.

Subsequently, the text is split into chunks using a character-based splitting strategy. This strategy divides the documents into manageable segments based on character count for facilitating efficient processing and embedding generation. Text vector embeddings are then generated using the text-embedding-3-small model. These vector embeddings are stored in a vector store, which enables rapid retrieval and similarity searches. Lastly, to maintain the truthfulness and relevance of the knowledge base, a human-in-the-loop validation step is conducted.

3.2. Metadata-based KG construction and expert validation

With the extraction of metadata from domain documents, a metadata-based KG is then constructed as the link between vector-based retrieval and KG-based retrieval. With the aim of enhancing precision as well as maintaining retrieval efficiency, this process begins by extracting metadata from domain-specific documents. The metadata includes titles, abstracts (extracted latent features from abstracts), keywords, and document identifiers. Subsequently, these metadata are mapped into directed triplets of head-entity-tail relationships. These triplets are then injected into a Neo4j-based KG to form a structured representation of the domain knowledge as well as the text chunks.

Within this KG, each document node contains its associated metadata so that it has explicit connections between documents, their features, and related attributes. For example, links can be created among documents that share specific keywords, thereby forming semantic relationships in the graph. The aim of constructing this structured layer is to complement vector-based retrieval by providing explicit and accurate identification of domain knowledge. Lastly, expert validation is conducted to confirm the accuracy and relevance of the KG. Domain specialists review the mapped triplets, nodes, and edges to ensure they align with concepts, processes, design heuristics, etc.

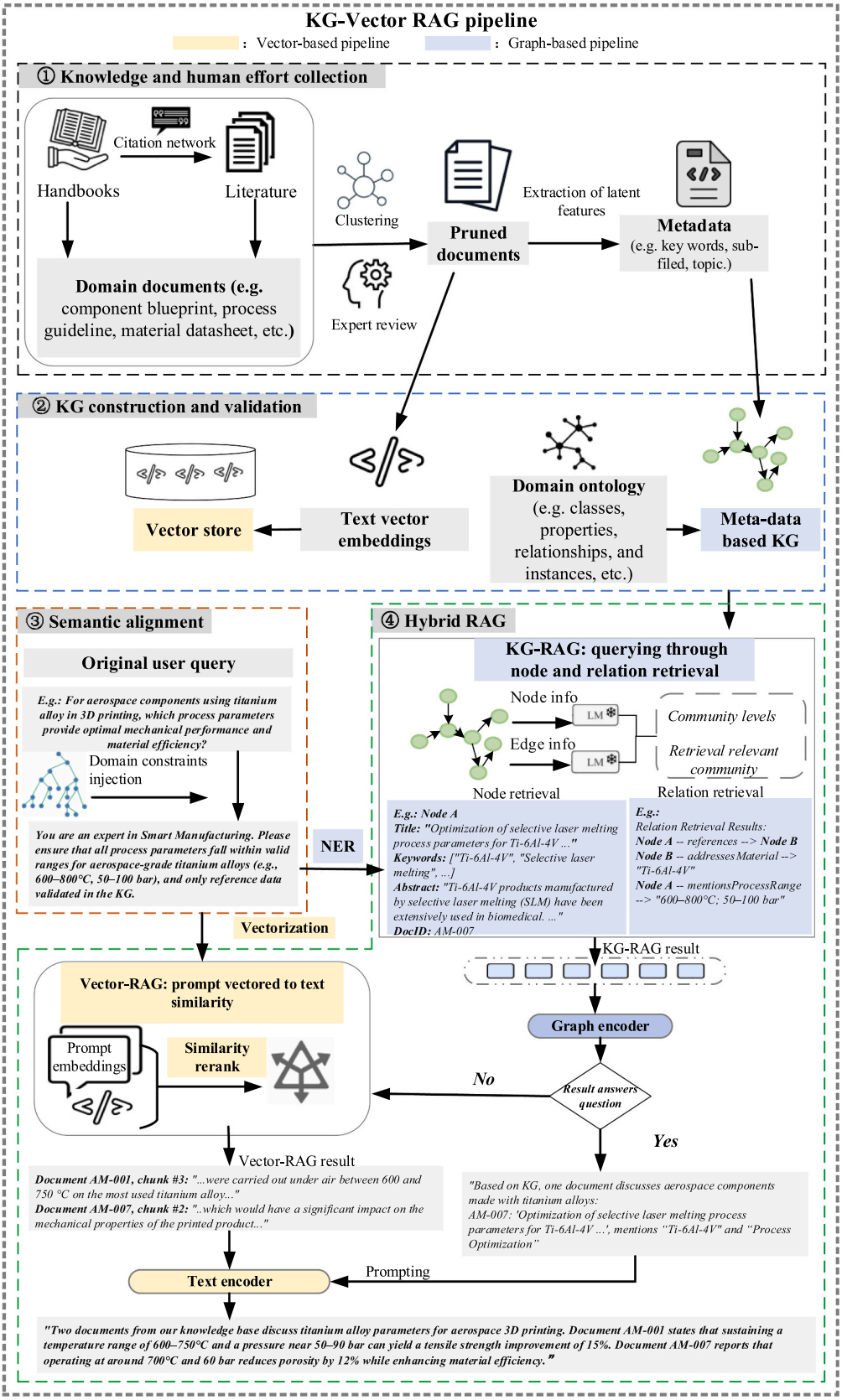


Fig. 1. Illustration of the proposed hybrid RAG approach.

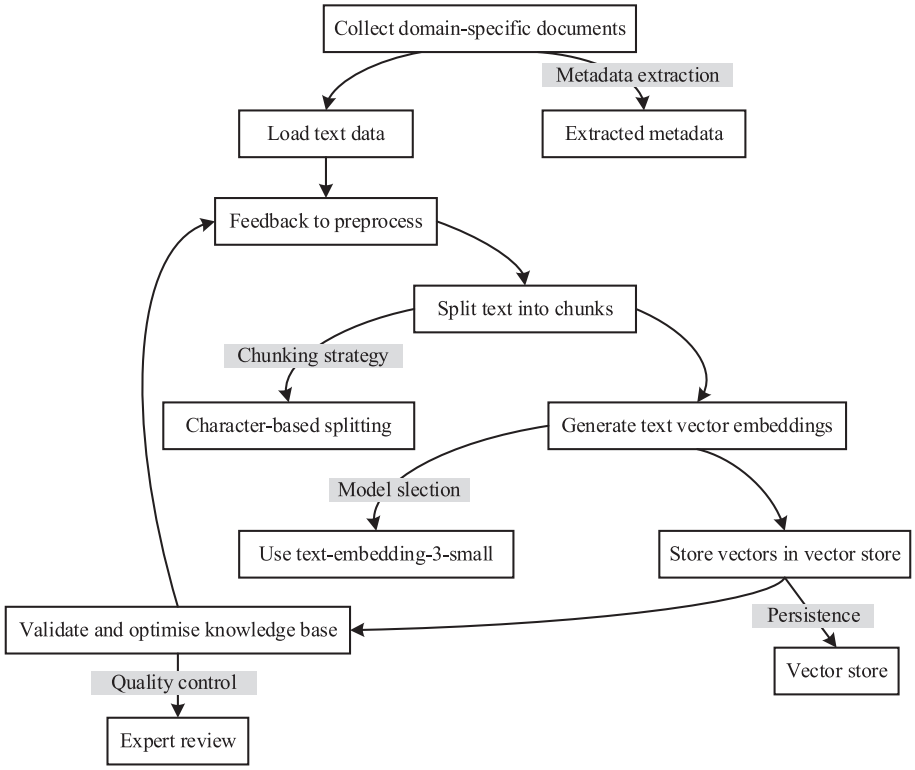


Fig. 2. Illustration of knowledge and human effort collection.

3.3. Semantic alignment through domain knowledge injection and constraints

Before the RAG pipeline, original user prompts are semantically aligned with SM domain-specific knowledge by imposing constraints. These constraints function as explicit rules that the prompts obey. To be specific, there are two types of constraints which are hard constraints and schema constraints. Hard constraints are firm rules that preserve factual correctness and logical consistency. For example, in the case of DfAM, if the DfAM ontology specifies valid temperature or pressure ranges for a particular AM process, the prompts reject any extracted content that deviates from these specifications. In addition to hard constraints, schema constraints define how different classes of entities relate to one another. In a DfAM scenario, "ImprovesPerformance" can link particular process parameters to certain mechanical or sustainability outcomes. By limiting recognized relationships to those declared in the ontology (e.g., "HasProperty," "UsedInProcess," "RequiresResource"), the system filters out spurious matches and retains only semantically valid connections.

In the context of SM, there are multiple granularities of knowledge: from high-level process descriptions (e.g., "Selective Laser Melting for titanium alloys") down to specific numerical parameters (e.g., "700 °C operating temperature"). Injecting domain concepts and enforcing constraints helps reconcile these varying levels of detail. In this case, retrieved passages match the ontology's representations. In the case of DfAM, certain semantic relations become relevant for retrieving and reasoning regarding the AM processes. Table 1 provides a generic example of possible relations. These representative relations illustrate how structured knowledge can be integrated into the retrieval pipeline under DfAM scenarios.

The mechanism of semantic alignment is described using pseudocode and the procedure is provided in Algorithm 1. As a result, semantic alignment by domain knowledge injection and constraints is one of the key elements of the presented methodology. By executing these steps, the system enhances both the precision and utility of the integrated knowledge base, ultimately improving its capacity to support complex reasoning and decision-making in DfAM scenarios.

Table 1
The description of typical semantic relations.

Relation	Symbol	Description
HasProperty	r_1	Links entities to their properties or attributes.
UsedInProcess	r_2	Associates materials or tools with manufacturing processes.
PartOf	r_3	Indicates component relationships within systems or products.
CausesEffect	r_4	Represents causal relationships between actions and outcomes.
RequiresResource	r_5	Specifies resources or conditions needed for operations.
ImprovesPerformance	r_6	Denotes actions that enhance performance metrics.

Algorithm 1. (*The procedure of semantic alignment.*)

Input:
- Q: *User Query* $Q = \{q_1, q_2, \cdots, q_n\}$
- G: *Metadata-driven KG* $G = \{N, R\}$
- D: *Vector store* $V = \{v_1, v_2, \cdots, v_n\}$

Output: A: *hybrid answer*

1.A = set()
2. ***for*** q_i ***In*** Q:
3. Q_constrained = *Inject_domain_constraints*(q_i)
4. Entities = *Extract_entities*(Q_constrained)
5.KG_nodes = *Retrieve_nodes*(G, Entities)
6. KG_relations = *Retrieve_relations*(G, Entities)
7. KG_response = *Combine*(KG_nodes, KG_relations)
8. ***if*** Need_detailed_answer(q_i):
9. Vector_results = *Retrieve_vectors*(V, Q_constrained)
10. A = *Merge*(KG_response, Vector_results)
11. ***else***:
12. A = *Summarise*(KG_response)
13. ***return*** A

3.4. Hybrid KG-Vector RAG

With the construction of a vector store, metadata-based KG and constrained prompts, a hybrid KG-Vector RAG approach is then presented. This approach leverages the explicit relationships captured by the KG and the rich but unstructured information retrieved from the vector store to generate accurate and comprehensive responses.

Following the constrained prompts modified in Section 3.3, named entity recognition (NER) is employed to extract key entities from the prompts for effective knowledge retrieval. The extracted entities are then utilized in the KG-RAG process, which involves both node retrieval and relation retrieval from the metadata-driven KG constructed in Section 3.2. The KG-RAG process retrieves relevant nodes and their interrelations, providing an initial overview of relevant sources related to the query.

Typically, the KG-RAG result directs the user to specific documents or sections within the KG that are most relevant to their query. However, for more detailed answers we need to search the vector store for specific text chunks that contain in-depth information. These text chunks are retrieved based on their vector similarity to the query, ensuring that the content is both relevant and detailed. Finally, the results from both KG-RAG and Vector-RAG are integrated to form a Hybrid Response. This combined answer leverages the structured insights from the KG and the detailed information from the vector store.

4. Experimental setup

4.1. Scenarios and data descriptions

To evaluate the effectiveness of the proposed KG-Vector RAG approach in a practical context, we apply it to the domain of design for additive manufacturing (DfAM). AM holds transformative potential for product design by offering unprecedented capabilities in creating complex geometries and customizing material properties. However, the intricacy of the AM design process presents substantial challenges. Designers are required to synthesize diverse knowledge sources, such as material properties, mechanical performance data, manufacturing constraints etc. The difficulty in retrieving relevant and accurate information from disparate and fragmented sources can severely impede both the efficiency and quality of design decisions within AM. In this context, this domain is particularly suitable due to its complexity and the necessity for integrating extensive domain-specific knowledge with advanced reasoning capabilities.

Initially, two textual handbooks specializing in DfAM were selected by the domain experts in AM to form the core of the datasets [47,48]. The documents contain detailed guidelines, best practices, material properties, process parameters, and design considerations specific to AM technologies. These two handbooks were then expanded to build a dataset through the citation network in two rounds. The resulting collection comprised 69 relevant publications, which serve as a refined corpus to construct a DfAM dataset. Lastly, the text data is obtained by processing the original PDF document using Langchain's PyPDFLoader [49], which extracts and segments the textual content for subsequent processing. After decomposing and analysing the dataset, the post-processing steps were taken to reveal and refine the topic clusters. As summarised in Table 2, 12 topic clusters were revealed to provide the most optimal grouping.

Following the methodology outlined in Ref. [50], a domain-specific KG is constructed from the extracted textual data from this document in Fig. 3, which contains 281 entities and 3536 relations. The KG captures the intricate relationships between entities, such as materials, manufacturing processes, design principles, performance metrics etc.

Table 2
List of topic clusters for DfAM.

ID	Topic clusters	Related documents
1	Additive manufacturing process optimisation	9
2	Material selection	3
3	Geometric design	7
4	Lightweight design	3
5	Design constraints for additive manufacturing	6
6	Topology optimisation	6
7	Multi-material additive manufacturing	4
8	Support structure optimisation	6
9	Thermal and residual stress analysis	8
10	Mechanical performance	9
11	Standards and certification	5
12	Energy efficiency	3

The construction process involves entity recognition, relation extraction, and the structuring of these elements into a coherent graph that reflects the semantic structure of DfAM. Moreover, the experimental setup in this section allows for a direct comparison between vector-based RAG and KG-based RAG methods. By utilizing the same source data and domain context, we can assess the strengths and limitations of each approach in handling domain-specific and complex queries.

4.2. Comparative RAG methods

Based on the documents described in Section 4.1, three RAG methods applied within the context of DfAM are conducted for comparison: vector-based RAG, graph-based RAG, and the proposed KG-Vector RAG approach. Each method employs a different retrieval mechanism to augment LLMs with external knowledge, which aims to improve the accuracy and relevance of generated responses in domain-specific applications.

(1) Vector-based RAG: the vector-based RAG method improves LLMs by integrating vector retrieval techniques with external unstructured textual data. The process begins with a user query related to information contained in external documents not included in the LLM's training data. These documents are segmented into manageable chunks due to the context size limitations of LLMs. Each chunk is converted into a vector representation using an embedding model, and the resulting embeddings are stored in a vector database. As shown in Fig. 4, when a query is issued, it is also converted into a vector embedding. The retrieval component performs a similarity search within the vector database to identify and rank the chunks most relevant to the query based on vector similarity measures. The top-ranked chunks are retrieved and aggregated to provide additional context for the LLM. The LLM then takes the original query along with the retrieved contextual information and generates a response. In this case, the vector-based approach allows the LLM to produce outputs that are grounded in the most recent and relevant external information, which improves the accuracy and contextual relevance of the responses. However, vector-based RAG relies solely on unstructured data retrieval and may struggle with complex queries that require structured reasoning or understanding of domain-specific relationships. Table 3 provides the parameter summary regarding the setup of vector-based RAG.

(2) Graph-based RAG: the graph-based RAG method incorporates structured-specialised knowledge stored in a domain KG during the retrieval process. Similar to vector-based RAG, the process begins with a user query. Instead of performing a vector similarity search, the query is used to traverse the domain KG for relevant entities and relationships. As shown in Fig. 5, the retrieval component extracts a subgraph consisting of nodes (entities) and edges (relationships) related to the query. The extracted subgraph provides a structured context that reflects the

Fig. 3. Illustration of constructed KG for DfAM.

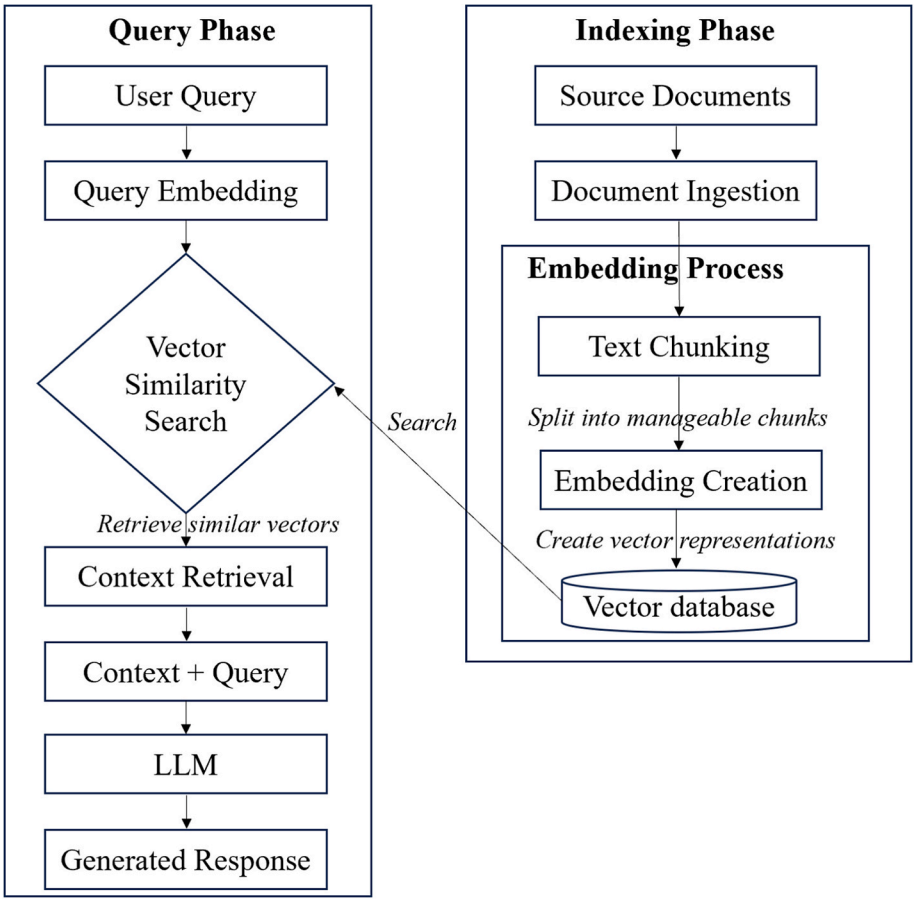


Fig. 4. Illustration of the vector-based RAG.

semantic relationships within the domain. The subgraph is then encoded into a format that the LLM can interpret, such as serialized triples or embeddings. The LLM generates a response by combining its pre-trained knowledge with the structured information from the domain KG. In this context, the graph-based method enhances the accuracy and relevance of the generated responses, particularly for queries that require an understanding of specific domain relationships. However, graph-based RAG may be limited by the completeness and granularity of the KG. If the KG lacks certain information, the LLM may not be able to generate comprehensive responses. Table 4 provides the parameter summary regarding the setup of graph-based RAG.

(3) KG-Vector RAG: the proposed KG-Vector RAG approach combines the strengths of both vector-based and graph-based retrieval methods to overcome their individual limitations. Specifically, the proposed method in this study employs a mutual indexing mechanism between the KG and vector embeddings to enhance retrieval performance and support complex decision-

Table 3
Parameters setup for vector-based RAG.

Parameters	Values
LLM	GPT-4o
Temperature	0
Embedding model	text-embedding-3-small
RAG pipeline	Langchain
Chunk size	1200
Chunk overlap	20
Maximum output tokens	1200
Chunks for similarity algorithm	20

making tasks in DfAM. To begin with, graph-based retrieval and vector-based retrieval are first attempted to retrieve information from different databases. The system first attempts to retrieve information from the KG using the user query. It searches for relevant entities and relationships that can directly answer the query through structured reasoning. As described in Section 3.2, six semantic relations commonly required in DfAM are identified and summarized in Table 5. If the KG does not contain sufficient information to answer the query, the system performs vector-based retrieval on the unstructured textual data. Then, with the incorporation of domain-specific vocabulary, terminology, and concepts, the retrieval and reasoning process is applied to guide the alignment of open information with the domain ontology. This includes specialized terms related to AM processes, materials, design principles, geometric complexities, and performance metrics. Lastly, schema constraints based on the domain ontology are applied to filter and refine the extracted information as described in Section 3.3. This step enforces consistency and validity within the KG by ensuring that only information conforming to the defined entities and relations is included.

In addition to the incorporation of domain-specific vocabulary and semantic relations, relying solely on the aforementioned methods may not be sufficient to achieve optimal results. In other words, without additional adjustments, using RAG directly might not fully capture the complexities of the DfAM domain with the data. Therefore, it is essential to enhance the prompts provided to the LLMs. Fine-tuning prompts, a process known as prompt engineering, is strongly recommended to guide LLMs effectively and improve the quality of the generated responses. Prompt enhancement (PE) involves crafting prompts that are tailored to the specific context and requirements of the domain. By providing clear instructions, context, and constraints within the prompts, we can better align the outputs of LLMs with the desired outcomes. In the DfAM context, enhanced prompts help the model to understand complex queries, utilize the KG effectively, and produce accurate and domain-relevant answers. Several refined prompts are shown in Table 6.

4.3. Evaluation metrics

In this section, two types of metrics are introduced to evaluate the

Table 4
Parameters setup for graph-based RAG.

Parameters	Values
LLM temperature	GPT-4o
Pipeline	Langchain
KG database	Neo4J
Chunk size	1200
Chunk overlap	20
Number of triples	5867
Number of nodes	4993
Number of edges	5815

Table 5
The descriptions of typical semantic relations regarding DfAM.

Relations	Symbols	Descriptions
HasMaterialProperty	r_1	Links materials to their specific properties or attributes.
SuitableForProcess	r_2	Associates materials or designs with compatible AM processes.
HasGeometryFeature	r_3	Links design to their geometric features or complexities.
RequiresSupport	r_4	Indicates that a design requires support structures during printing.
EnhancesPerformance	r_5	Denotes design modifications that improve performance metrics.
CompatibleWithMachine	r_6	Associates materials or designs with specific AM equipment.

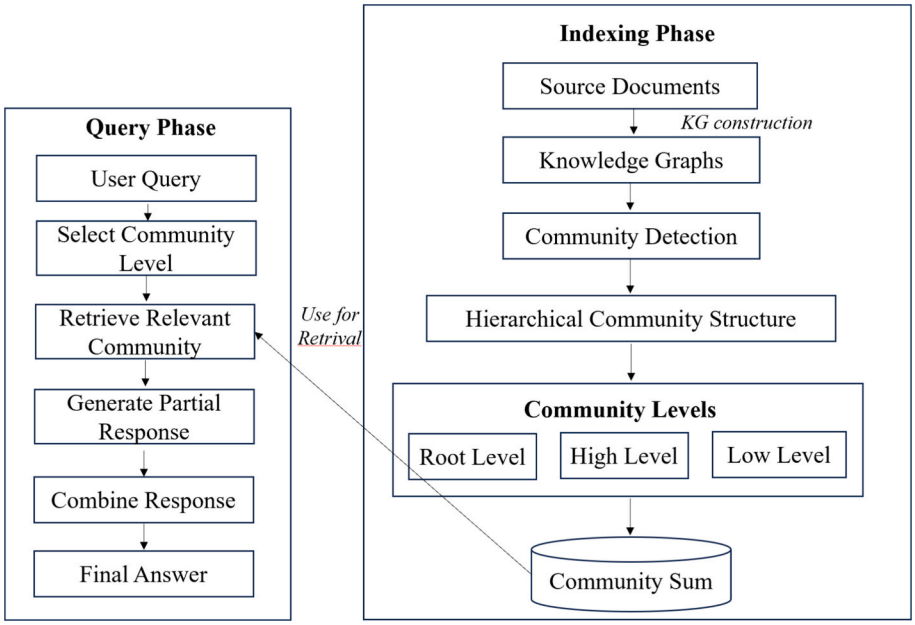


Fig. 5. Illustration of KG-based RAG.

Table 6
Examples for prompts refinement.

Prompts Refine	Descriptions
1	*Using the knowledge graph of additive manufacturing materials and processes, identify the optimal material-process combination for producing a lightweight aerospace component with high thermal resistance. Provide reasoning based on material properties and process capabilities.*
2	*Refer to the DfAM ontology and determine which additive manufacturing process is most suitable for fabricating a component with complex internal geometries. Explain how the selected process addresses geometric complexities.*
3	*Based on the relationships defined in the DfAM knowledge graph, suggest design modifications that could enhance the structural integrity of a 3D-printed component without significantly increasing weight. Justify your suggestions using relevant semantic relations.*
4	*Utilize the domain-specific terminology and concepts in DfAM to evaluate the compatibility of titanium alloys with different additive manufacturing processes. Provide a comparative analysis highlighting key material-process interactions.*

performances of the proposed KG-Vector approach in comparison with different RAG approaches and diverse LLMs, including generic RAG metrics and domain metrics.

(1) Generic RAG metrics: the evaluation of the proposed hybrid KG-Vector RAG approach is conducted under generic RAG metrics using three key metrics: exact match (EM) [51], context precision (CP) [52], and latency [53]. Independent of domain-specific considerations, EM and CP focus on the overall quality of the generated answers, including accuracy and relevance. Latency measures the system's responsiveness, which is crucial for real-time applications. EM evaluates whether the generated answer exactly matches the reference (ground truth) answer. It is a strict metric that assesses the precision of the response at a granular level, which ensures that the LLMs produce accurate outputs. CP measures the proportion of relevant information in the generated answer relative to all the information provided in the answer. It evaluates how accurately the RAG system incorporates retrieved content into the response, which reflects the quality and informativeness of the generated outputs. By combining EM and CP, the insights into both the exactness and the quality of the generated responses can be evaluated. Latency refers to the time taken by the RAG system to complete a response for a single query. It quantifies the system's efficiency by measuring how quickly it retrieves and generates answers. Moreover, latency is defined as the mean time required to process a query, covering both the retrieval phase (searching for relevant knowledge) and the generation phase (producing the final response). It is important for interactive and real-time applications. By integrating EM, CP and latency, this study provides a comprehensive assessment of the RAG system's effectiveness, which not only measures the accuracy and contextual relevance of the generated answers but also evaluates the sensitivity of the response time.

(2) Domain metrics through LLMs as judges: the evaluation employs an LLM to act as an expert judge. The LLM is prompted to evaluate the generated answers based on specific criteria relevant to DfAM. LLMs used here are GPT-4o [54]. The prompt used here is: "*Below is a student's answer to a question based on the provided DfAM document. Please evaluate the answer for accuracy, alignment with domain knowledge, conciseness, and relevance. Identify any issues in the answer, and then provide a score. Score range: 1 to 10.*".

5. Results and discussion

To assess the performance of the proposed KG-Vector RAG approach, we employ both generic and domain-specific evaluation metrics under different LLMs in this section.

5.1. Experimental results under generic metrics

To evaluate the effectiveness of the proposed KG-Vector approach in domain-specific Q&A tasks, a comparative analysis of three different LLMs with four diverse RAG approaches is presented based on their performances on accuracies in terms of EM, CP and latency. Three different LLMs include GPT-4o [54], GPT-4o mini [55] and DeepSeek V2 [55]. Moreover, the comparative RAG strategies are classified into four groups: Group 1 (vector-based), Group 2 (graph-based), Group 3 (KG-Vector), and Group 4 (KG-Vector with PE). Each group represents a distinct RAG approach. Group 1 employs a vector-based retrieval strategy that focuses solely on unstructured textual data. The graph-based RAG method in Group 2 only leverages a domain KG to retrieve relevant information. Group 3 introduces the proposed hybrid KG-Vector RAG approach, which combines the strengths of vector-based and graph-based retrieval. Building on the hybrid KG-Vector RAG method, Group 4 incorporates PE to further refine the interaction with LLMs.

Fig. 6 compares the EM experimental results of three LLMs tested with four RAG strategies. The results demonstrated that performance improves gradually as the RAG strategy transitions from vector-based to graph-based, then to KG-Vector RAG, and finally to KG-Vector RAG with PE. The comparison of the four groups showed the incremental benefits of incorporating structured domain knowledge provided by domain KGs and leveraging hybrid retrieval mechanisms. The inclusion of PE further enhanced performance and reflected the importance of domain-specific prompts in guiding LLMs. Specifically, Group 1 indicated the lowest EM scores among all groups, which reflects that vector-based RAG is limited by its reliance on unstructured data retrieval, which may not fully capture domain-specific relationships. Introducing structured domain knowledge via KGs in Group 2 improved EM scores, which demonstrated the advantage of graph-based reasoning in understanding domain-specific relationships and providing accurate responses. The hybrid KG-Vector RAG strategy in Group 3 further enhanced performance by combining the strengths of vector-based and graph-based approaches. The improved scores highlighted the value of integrating structured and unstructured data retrieval for comprehensive and accurate responses. Group 4 achieved the highest EM scores across all LLMs, with GPT-4o scoring 77.8 %, GPT-4o mini at 75.5 %, and DeepSeek V2 at 74.8 %. The inclusion of PE in Group 4 demonstrated its critical role in enhancing the performances of all tested LLMs. By crafting domain-specific prompts, this suggested that PE is effective when applied in conjunction with the hybrid KG-Vector RAG approach. Additionally, the performance hierarchy among the tested LLMs was consistent across all RAG strategies. GPT-4o consistently achieves the highest scores, followed by GPT-4o mini and then DeepSeek V2. Specifically, GPT-4o showed the largest performance improvement, with EM scores increasing from 63.4 % (Group 1) to 77.8 % (Group 4). It indicated that larger and more advanced LLMs benefit more from the proposed strategies, particularly the hybrid KG-Vector RAG with PE. The performances of GPT-4o mini and DeepSeek V2 also highlighted that even smaller models can benefit from the proposed hybrid RAG framework and PE.

Fig. 7 illustrates the performances of four RAG strategies with three different LLMs on CP. The experimental results of CP scores highlighted a clear progression as the retrieval mechanism moves from vector-based RAG (Group 1) to KG-Vector RAG with PE (Group 4), which reflects the value of integrating structured and unstructured retrieval mechanisms and the additional benefit of domain-specific prompt engineering. Specifically, in Group 1, CP scores were the lowest with the limitations of vector-based RAG, such as its reliance on unstructured data and inability to capture semantic relationships. Introducing graph-based retrieval improved CP scores across all models in Group 2. The structured reasoning capabilities of KGs allowed for more accurate and

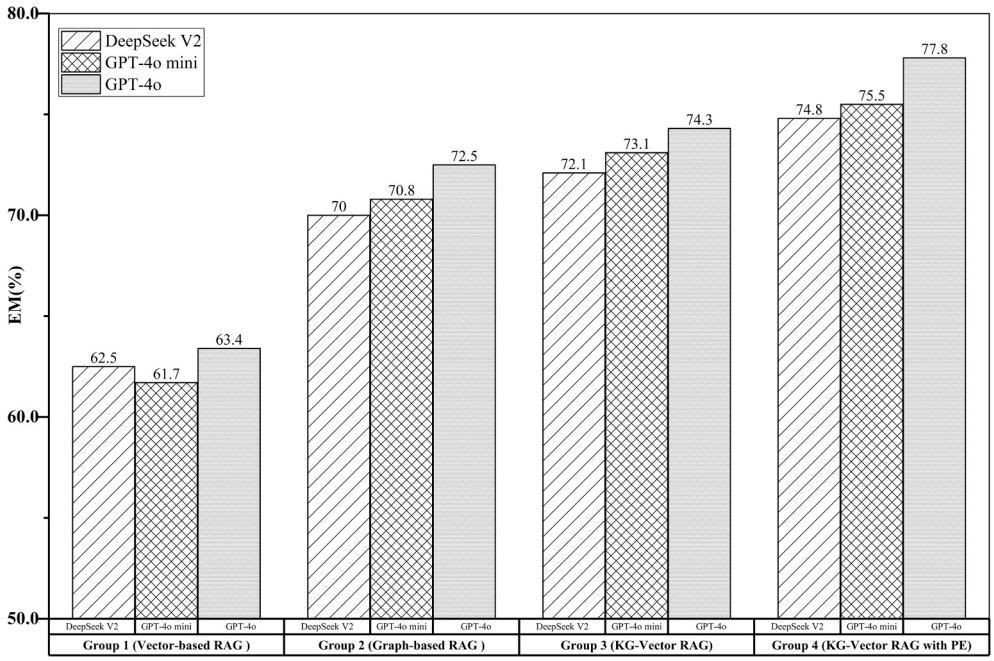


Fig. 6. Performances of four RAG strategies with three different LLMs on EM.

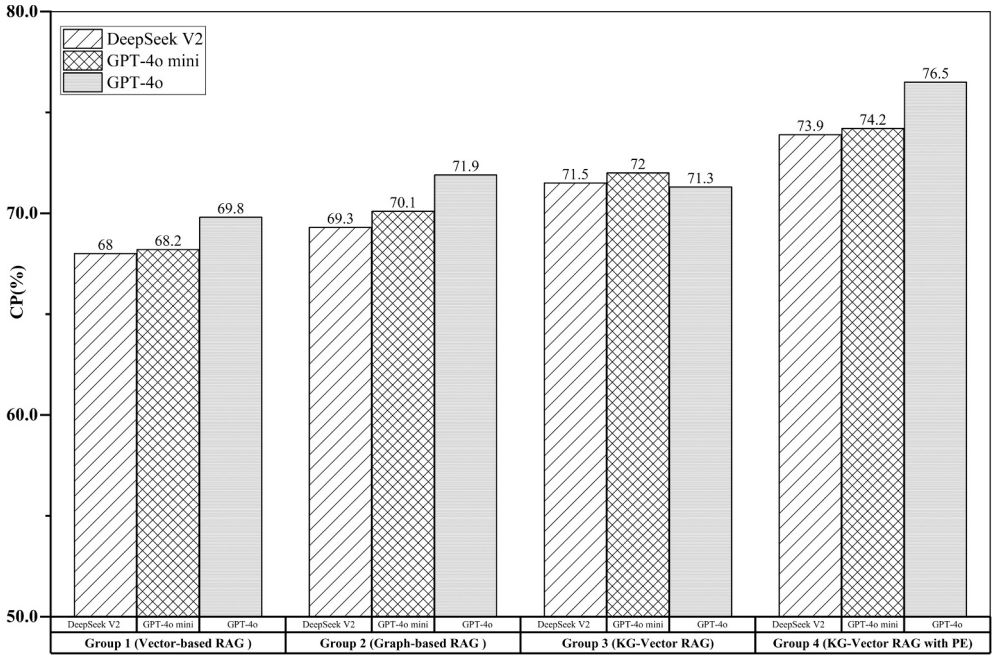


Fig. 7. Performances of four RAG strategies with three different LLMs on CP.

contextually relevant information retrieval, resulting in higher CP scores compared to Group 1. The hybrid KG-Vector RAG approach further enhanced CP performance by combining the strengths of vector-based and graph-based methods. GPT-4o scored 71.3 %, GPT-4o mini achieved 72.0 %, and DeepSeek V2 reached 71.5 %. These results indicated that the hybrid strategy successfully balances the retrieval of unstructured and structured data to improve the quality of responses. The inclusion of PE resulted in the highest CP scores across all models, with GPT-4o scoring 76.5 %, GPT-4o mini at 74.2 %, and DeepSeek V2 at 73.9 %. The prompts tailored by PE helped the LLMs interpret complex queries effectively and utilised domain-specific knowledge from the KG and vector embeddings for precise and context-aware responses. Additionally, as with the CP metric, GPT-4o consistently outperformed GPT-4o mini and DeepSeek V2 across all RAG strategies. GPT-4o exhibited the most notable improvement in CP scores, increasing from 69.8 % (Group 1) to 76.5 % (Group 4). GPT-4o mini and DeepSeek V2 showed similar trends, with the inclusion of PE consistently improving their ability to provide accurate and relevant context in generated responses. It demonstrated that while larger models achieve higher overall CP scores, smaller models also gain substantial performance improvements when using the proposed retrieval strategies. Meanwhile, PE's incorporation of tailored instructions, contextual constraints, and domain vocabulary ensured that the generated answers were not only accurate but also aligned with the specific requirements of the DfAM

domain.

Fig. 8 compares the mean response time of four RAG strategies with three different LLMs. The latency results illustrate the efficiency associated with different RAG approaches across various LLMs. Overall, Group 1 (vector-based retrieval) achieved the lowest latency, while the incorporation of KG-based retrieval (Group 2), hybrid KG-Vector RAG (Group 3), and PE-enhanced KG-Vector RAG (Group 4) progressively increased response times. Group 1 exhibited the fastest response times, with DeepSeek V2 achieving the lowest latency (3.84 s), followed by GPT-4o mini (6.07 s) and GPT-4o (7.01 s). The experimental results align with expectations, as vector retrieval operates on pre-embedded unstructured text, minimizing computational overhead. Group 2 demonstrated a moderate increase in latency due to the additional graph traversal and reasoning steps. The latency values rose to 4.86 s for DeepSeek V2, 8.22 s for GPT-4o mini, and 9.46 s for GPT-4o, which indicates the added complexity of querying a structured KG. Group 3 further increased latency compared to Group 2, with DeepSeek V2 reaching 5.79 s, GPT-4o mini at 9.99 s, and GPT-4o at 10.96 s. It suggests that while the hybrid approach effectively integrates structured KG knowledge with vector-based retrieval, the additional computation required for alignment and reasoning adds latency. Group 4 recorded the highest latency, with DeepSeek V2 at 8.67 s, GPT-4o mini at 14.37 s, and GPT-4o at 16.33 s. The substantial increase is attributed to PE, which introduces additional pre-processing steps, such as more complex prompt structuring and iterative refinement, which leads to improved accuracy at the cost of response time. Larger LLMs (e.g., GPT-4o) exhibited higher latency compared to smaller models (e.g., GPT-4o mini), which reflects the increased computational complexity required for advanced reasoning and generation. Graph-based and hybrid retrieval approaches introduced additional processing time, as structured knowledge extraction and alignment demand greater computational resources. The impact of PE increased latency in Group 4 across all LLMs. While PE improves response quality, it introduces a trade-off by making query processing more time-intensive. Moreover, DeepSeek V2 consistently maintained the lowest latency in all groups, which indicated its efficiency in rapid response generation. However, it may lack the reasoning depth of larger models like GPT-4o.

5.2. Experimental results under domain metrics evaluated by LLMs

To assess the performance of the proposed model within the specialized context of DfAM, domain-specific evaluations were conducted using LLMs as expert judges. The models were evaluated based on criteria such as technical accuracy, adherence to domain knowledge, relevance, and clarity. As mentioned in Section 5.1, four RAG strategies, tested across three LLMs (DeepSeek V2, GPT-4o mini, and GPT-4o), were categorised into four groups.

The experimental results evaluated by GPT-4o are presented. As demonstrated in Table 7, the "Scores" represent the evaluations from three separate trials. The results consistently showed improved scores as retrieval strategies progressed from Group 1 to Group 4, and highlighted the advantages of integrating structured knowledge, hybrid retrieval methods, and PE. Specifically, the evaluated scores in Group 1 are the lowest across all LLM. The reliance on unstructured data limited the ability to address complicated-specialised queries accurately in Group 1. In Group 2, incorporating KGs improved scores, which reflect that the structured knowledge representation in DfAM KGs allows for more accurate retrieval and reasoning, but this strategy still struggled with unstructured data. For Group 3, the hybrid retrieval approach further enhanced performances by combining the strengths of vector-based and graph-based methods. It also demonstrated the benefits of integrating structured and unstructured data retrieval for comprehensive responses. The inclusion of PE yielded the highest scores in Group 4. In this group, PE ensured that LLMs interpret complex queries more effectively and generate responses that are better aligned with domain-specific requirements. Additionally, the hierarchy of LLM performances remained

Table 7
Experimental results of different RAG approaches under domain metric.

LLMs	RAG approaches	Scores (trial 1, 2, 3)
DeepSeek V2	Vector-based	(5.5, 5.0, 5.5)
	KG-based	(6.0, 5.5, 6.0)
	KG-Vector	(6.5, 6.0, 6.5)6.5, 6.0, 6.5
	KG-Vector with PE	(7.0, 6.5, 7.0)
GPT-4o mini	Vector-based	(6.0, 5.5, 6.0)
	KG-based	(6.5, 6.0, 6.5)
	KG-Vector	(7.0, 6.5, 7.0)
	KG-Vector with PE	(7.5, 7.0, 7.5)
GPT-4o	Vector-based	(6.5, 6.0, 6.5)
	KG-based	(7.0, 6.5, 7.0)
	KG-Vector	(7.5, 7.0, 7.5)
	KG-Vector with PE	**(8.0, 7.5, 8.0)**

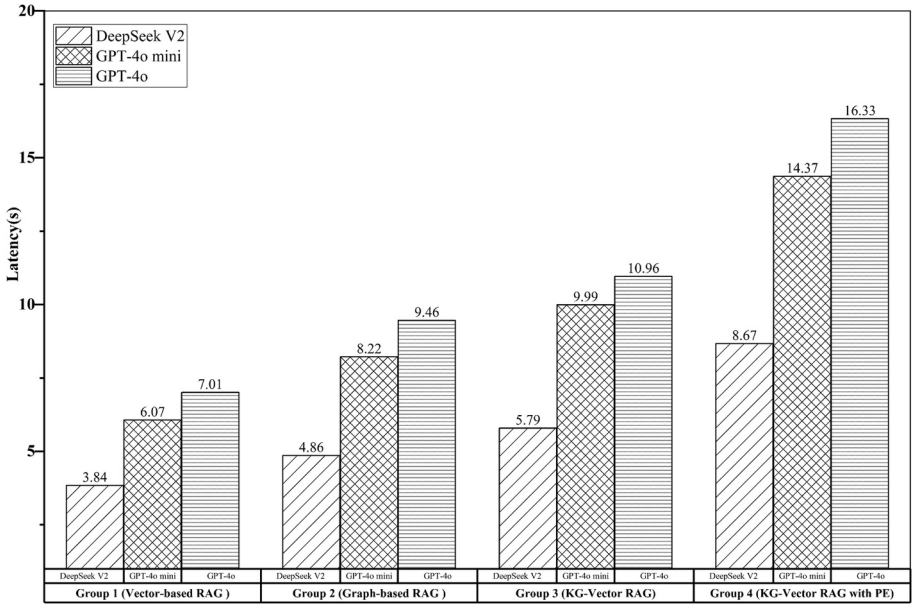


Fig. 8. Performances of four RAG strategies with three LLMs on latency.

consistent across all groups. GPT-4o consistently achieved the highest scores in three separate trials with (8.0, 7.5, and 8.0), followed by GPT-4o mini and DeepSeek V2. It demonstrated that larger, more advanced models benefit more from the proposed RAG strategies. Specifically, three trial scores of GPT-4o improved significantly across groups, which highlights the ability of GPT-4o to leverage complex retrieval mechanisms and tailored prompts for superior domain-specific performance. Ranging from Group 1 to Group 4, three test scores of GPT-4o mini lagged behind GPT-4o and showed that even smaller models can benefit from hybrid retrieval and PE. While achieving lower scores overall, DeepSeek V2 showed improvements in three trials from Group 1 to Group 4, which indicates the robustness of the proposed approach even for less advanced LLMs. The consistent improvement in Group 4 also further highlighted the importance of PE in guiding LLMs.

Similarly, Fig. 9 shows the average scores of different LLMs evaluated by GPT-4o under diverse RAG strategies. The "average" is the mean of these trials. The consistent inclination of the mean score across different RAG models and diverse LLMs further showed the effectiveness of the proposed hybrid RAG approach and the importance of PE incorporation. Notably, the highest average score of 7.83 was achieved by GPT-4o with the KG-Vector approach and PE. It reflected the superior capability of GPT-4o in generating accurate and relevant responses to the Q&A system in the DfAM domain. Albeit slightly lower than GPT-4o, GPT-4o mini with KG-Vector and PE indicated strong performance with an average score of 7.33. From this perspective, the inclusion of PE can guide LLMs to enhance their performances, which demonstrates the effectiveness of fine-tuning prompts. Additionally, the effectiveness of the proposed KG-Vector RAG was highlighted by comparing different RAG methods within the same LLM.

In addition to these numerical metrics, a DfAM Q&A practice is also presented in Table 8. A user posed the query: "*For producing a heat-resistant component with complex geometry in a time-efficient manner, which additive manufacturing process and material should I choose*?" Applying the hybrid RAG framework, the system first attempted KG-based retrieval. It searched the KG for entities such as heat-resistant materials and AM processes supporting complex geometries and examined relations like process capabilities and material properties. The KG might reveal that Inconel 718 is a heat-resistant material, that selective laser melting (SLM) supports complex geometries but has medium production time, and that directed energy deposition (DED) has a shorter production time but limited geometric complexity. However, no direct match satisfied all criteria. The system then proceeded to vector-based retrieval, performing a semantic search to retrieve documents related to AM processes, materials, and production times. Relevant documents might indicate that electron beam melting (EBM) efficiently processes heat-resistant alloys like Inconel 718 and supports complex geometries, offering faster build times compared to SLM. Associative retrieval in chunk space further connected information fragments about EBM's efficiency and Inconel 718′s properties, concluding that EBM with Inconel 718 meets all the user's criteria. Finally, the retrieved information from both the KG and vector retrieval was consolidated, and the LLM generated a comprehensive answer: "*Electron Beam Melting (EBM) using Inconel 718 is recommended for producing a heat-resistant component with complex geometry efficiently. EBM offers faster production times due to its high energy density and effectively processes heat-resistant materials like Inconel 718, which is suitable for your requirements.*" This example demonstrated how the proposed hybrid RAG approach successfully retrieves relevant information not fully available in the KG alone. It also can support complex decision-making by addressing a multifaceted query involving material properties, process capabilities, and production efficiency, and compensate for the KG's sparsity by incorporating external information. The layered retrieval and reasoning process ensures that the most reliable and relevant information is utilized at each step, which enhances both recall and accuracy.

5.3. Comparative analysis and observations

The experimental results under generic metrics revealed a clear hierarchy in the performances of the different models and retrieval approaches. GPT-4o combined with the KG-Vector method and PE achieved the highest EM score of 77.8 % and a CP score of 76.5 %, which indicates better capability in generating accurate and relevant responses within the DfAM domain. It underscores the effectiveness of integrating structured knowledge from KGs with unstructured data through vector retrieval, further enhanced by carefully crafted prompts. GPT-4o mini with the same retrieval approach and PE closely followed, which demonstrates that even smaller models can benefit from the proposed hybrid RAG. The consistent improvement across LLMs with the inclusion of PE highlights the critical role of PE in guiding LLMs to produce more precise and contextually appropriate responses. The latency results demonstrated the trade-off between retrieval complexity and response efficiency. For real-time applications, Group 1 remains the most efficient

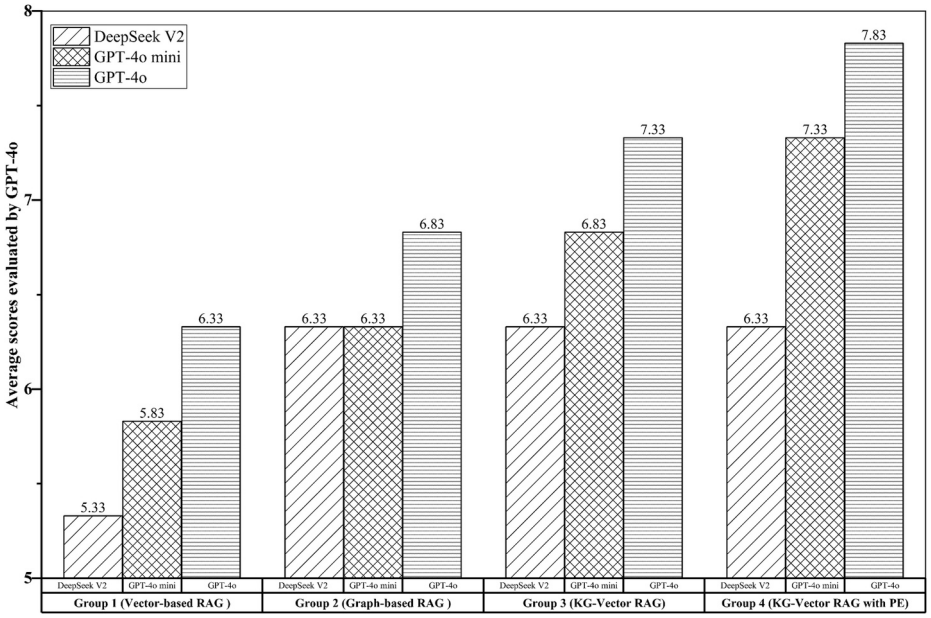


Fig. 9. The average scores of different LLMs evaluated by GPT-4o under diverse RAG strategies.

Table 8
Examples demonstrating different RAG scenarios under the DfAM domain.

Examples demonstrating different RAG scenarios under the DfAM domain
Q1: Identify suitable additive manufacturing processes for topology-optimized lightweight aerospace components. **Graph Retrieval**: *(Topology Optimization, is used in, Aerospace Component Design)* *(Selective Laser Melting, is suitable for, Aerospace Components)* *(Electron Beam Melting, is suitable for, Aerospace Components)* *(Binder Jetting, is less suitable for, High-Strength Components)* **Vector Retrieval**: Not executed. **A1**: Suitable AM processes include Selective Laser Melting (SLM) and Electron Beam Melting (EBM).
Q2: Among these processes, which one minimizes production time? **Graph Retrieval:** *(SLM, has Build Rate, Medium)* *(EBM, has Build Rate, High)* *(EBM, requires Vacuum Environment, Yes)* **Vector Retrieval**: Not executed. **A2**: Electron Beam Melting (EBM) has a higher build rate than SLM, potentially minimizing production time.
Q3: Which materials compatible with EBM maintain structural integrity in aerospace components? **Graph Retrieval**: No direct matches found in KG. **Hybrid Retrieval**: Not executed. **Vector Retrieval** results: Doc1: "*EBM commonly uses titanium alloys like Ti-6Al-4 V, offering high strength and excellent fatigue properties suitable for aerospace applications*." Doc2: "*Aluminium alloys are challenging to process with EBM due to their reflective properties and lower melting points*." Doc3: "*Nickel-based superalloys can be used in EBM for high-temperature aerospace components*." **A3:** Titanium alloy Ti-6Al-4 V is compatible with EBM and maintains structural integrity in aerospace components.
Q4: What is the recommended process and material combination to achieve minimal production time and maintain structural integrity? **Graph Retrieval**: Not directly found. **Hybrid Retrieval**: Consolidating previous findings: **Process**: Electron Beam Melting (EBM) **Material**: Titanium alloy Ti-6Al-4 V **Rationale**: High build rate of EBM minimizes production time; Ti-6Al-4 V ensures structural integrity. **A4**: The recommended combination is using EBM with titanium alloy Ti-6Al-4 V to minimize production time while maintaining structural integrity for the topology-optimized aerospace component.

option due to its minimal latency. In scenarios prioritizing reasoning and knowledge integration, Group 3 balances retrieval complexity and response time effectively. For cases where accuracy and contextual understanding outweigh latency concerns, Group 4 is preferred, despite the computational overhead. Optimization techniques such as query preprocessing and adaptive retrieval strategies could be explored to reduce latency while maintaining retrieval effectiveness. It highlighted that while hybrid and PE-enhanced approaches enhance accuracy and contextual understanding, they introduce computational time that must be carefully considered for practical deployment.

Moreover, as illustrated in section 5.2, the experimental results under domain metrics showed a similar pattern, where the GPT-4o model acts as an expert judge. Obviously, GPT-4o with the KG-Vector approach and PE achieved the best performance and attained the highest average score (7.83). It reflects exceptional performances of the GPT-4o model with the proposed KG-Vector approach and PE strategy in generating accurate, comprehensive, and domain-relevant answers. The incremental improvements observed when moving from vector-based to KG-based and then to KG-Vector approaches confirmed the advantage of combining structured and unstructured data retrieval mechanisms. The performances of LLMs with the inclusion of PE further highlighted the importance of prompt design in enhancing the LLMs' understanding and adherence to domain knowledge. Notably, even though DeepSeek V2 is a less powerful model compared to GPT-4o in the DfAM tasks, its performance significantly was improved with the KG-Vector approach and PE, which indicates the robustness of the proposed method and PE strategy across different model capacities.

These results indicate that the integration of structured knowledge from KGs with unstructured vector-based retrieval enhances both the precision and efficiency of information retrieval in domain-specific Q&A systems. Furthermore, the incorporation of prompt enhancement contributed to additional improvements in output quality, indicating the importance of tailored prompt engineering for refining LLM responses. Broadly speaking, the Q&A case study in DfAM further illustrated the practical effectiveness of the proposed hybrid RAG approach. The system navigated complex queries that required integrating multiple facets of domain knowledge, including material properties, manufacturing processes, performance requirements etc. The initial KG-based retrieval provided foundational information but was insufficient to fully address the query. By proceeding to vector-based retrieval and performing associative retrieval in the chunked data space, the system incorporated external information to compensate for the KG's sparsity. The final consolidated answer was aligned with domain expertise, which demonstrates the system's ability to enhance both recall and accuracy through layered retrieval and reasoning processes. In summary, this study showcases how the hybrid KG-Vector RAG approach can effectively support complex decision-making tasks by leveraging the strengths of both KGs and vector retrieval methods.

6. Conclusions

This study proposed a KG-Vector RAG approach that leverages KG and LLMs to facilitate a domain Q&A, which provides accurate responses tailored to domain-specific requirements in SM. The hybrid KG-Vector RAG approach is proposed to bridge the gap between vector similarity and knowledge relevance for enhancing the accuracy and domain relevance of generated responses. The integration of the specialised vocabulary, semantic alignment, and prompt improvement further enhances the retrieval and generation for complex decision-making tasks. Lastly, a case study of DfAM was conducted to validate

the effectiveness of the proposed KG-Vector RAG approach in handling complex and multifaceted queries. Experimental results demonstrated that the proposed hybrid KG-Vector RAG approach outperformed the baseline models across both generic and domain-specific metrics. The inclusion of PE further improved the accuracy, precision, and relevance of responses, as shown by higher EM and CP scores across different LLMs.

The proposed method is tailored to the SM domain, which means that the initial collection and preparation of specialized knowledge need to be adjusted when applying the method to other domains. Future research will focus on adapting the proposed method for broader domain applications by refining the process of knowledge collection and preparation to fit different domain requirements. Also, improving the timeliness of KGs remains a critical challenge. Based on that, exploring strategies for automated KG updates is interesting, which ensures that emerging domain-specific terms and recent advancements are integrated efficiently. Lastly, further investigation will be conducted on dynamically updating external knowledge sources to improve the system's adaptability and reliability in fast-changing environments.

CRediT authorship contribution statement

Yuwei Wan: Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Zheyuan Chen:** Writing , – original draft, Methodology, Investigation, Conceptualization. **Ying Liu:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization. **Chong Chen:** Writing – review & editing, Methodology, Investigation. **Michael Packianather:** Writing – review & editing, Supervision, Methodology.

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.

Acknowledgment

Our work was supported by the National Natural Science Foundation of China (No. 62302103).

Data availability

The authors do not have permission to share data.

References

[1] S. Ren, Y. Zhang, Y. Liu, T. Sakao, D. Huisingh, C.M. Almeida, A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing: A framework, challenges and future research directions, *Journal of Cleaner Production* 210 (2019) 1343–1365.

[2] A. Schlemitz, V. Mezhuyev, Approaches for data collection and process standardization in smart manufacturing: Systematic literature review, *J. Ind. Inf. Integr.* 38 (2024) 100578.

[3] Z. Wang, X. Liang, M. Li, S. Li, J. Liu, L. Zheng, "Towards Cognitive Intelligence-enabled Product Design: The Evolution, State-of-the-art, and Future of AI-enabled Product Design," *Journal of Industrial Information, Integration* (2024/12/09/ 2024,) 100759, https://doi.org/10.1016/j.jii.2024.100759.

[4] K. Lei, P. Guo, Y. Wang, J. Zhang, X. Meng, L. Qian, Large-Scale Dynamic Scheduling for Flexible Job-Shop With Random Arrivals of New Jobs by Hierarchical Reinforcement Learning, *IEEE Trans. Ind. Inf.* 20 (1) (2024) 1007–1018, https://doi.org/10.1109/TII.2023.3272661.

[5] Y. Xia, M. Shenoy, N. Jazdi, and M. Weyrich, "Towards autonomous system: flexible modular production system enhanced with large language model agents," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 12-15 Sept. 2023 2023, pp. 1-8, doi: 10.1109/ETFA54631.2023.10275362.

[6] C. Picard, et al., "From Concept to Manufacturing, Evaluating Vision-Language Models for Engineering Design," (2023).

[7] 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.* (2024).

[8] Y. A. Yadkori, I. Kuzborskij, A. György, and C. Szepesvári, "To Believe or Not to Believe Your LLM," *arXiv preprint arXiv:2406.02543,* 2024.

[9] N. Harvel, F.B. Haiek, A. Ankolekar, D.J. Brunner, Can LLMs Answer Investment Banking Questions? Using Domain-Tuned Functions to Improve LLM Performance on Knowledge-Intensive Analytical Tasks, *Proceedings of the AAAI Symposium Series* 3 (1) (2024) 125–133.

[10] N. Zhang *et al.*, "A comprehensive study of knowledge editing for large language models," *arXiv preprint arXiv:2401.01286,* 2024.

[11] S. K. Freire, C. Wang, and E. Niforatos, "Chatbots in knowledge-intensive contexts: Comparing intent and llm-based systems," *arXiv preprint arXiv:2402.04955,* 2024.

[12] S. Wu *et al.*, "Retrieval-Augmented Generation for Natural Language Processing: A Survey," *arXiv preprint arXiv:2407.13193,* 2024.

[13] D. Edge *et al.*, "From local to global: A graph rag approach to query-focused summarization," *arXiv preprint arXiv:2404.16130,* 2024.

[14] E. Ruiz, M.I. Torres, A. del Pozo, Question answering models for human–machine interaction in the manufacturing industry, *Comput. Ind.* 151 (2023) 103988.

[15] G.M. Biancofiore, Y. Deldjoo, T.D. Noia, E. Di Sciascio, F. Narducci, Interactive question answering systems: Literature review, *ACM Comput. Surv.* 56 (9) (2024) 1–38.

[16] B. Ojokoh, E. Adebisi, A review of question answering systems, *Journal of Web Engineering* 17 (8) (2018) 717–758.

[17] Q. Xiong, J. Zhang, P. Wang, D. Liu, R.X. Gao, Transferable two-stream convolutional neural network for human action recognition, *J. Manuf. Syst.* 56 (2020) 605–614.

[18] T. Wang, J. Li, Z. Kong, X. Liu, H. Snoussi, H. Lv, Digital twin improved via visual question answering for vision-language interactive mode in human–machine collaboration, *J. Manuf. Syst.* 58 (2021) 261–269.

[19] H. Han, J. Wang, X. Wang, Leveraging Knowledge Graph Reasoning in a Multi-Hop Question Answering System for Hot Rolling Line Fault Diagnosis, *IEEE Trans. Instrum. Meas.* (2023).

[20] P. Renna, Multi-agent based scheduling in manufacturing cells in a dynamic environment, *Int. J. Prod. Res.* 49 (5) (2011) 1285–1301.

[21] W. Zheng, et al., Pay attention to doctor-patient dialogues: Multi-modal knowledge graph attention image-text embedding for COVID-19 diagnosis, *Inf. Fusion* (2021).

[22] Y. Wan, et al., Making knowledge graphs work for smart manufacturing: Research topics, applications and prospects, *J. Manuf. Syst.* 76 (2024) 103–132.

[23] Z. Chen, Y. Wan, Y. Liu, A. Valera-Medina, A knowledge graph-supported information fusion approach for multi-faceted conceptual modelling, *Inf. Fusion* 101 (2024) 101985.

[24] C. Chen, et al., Reinforcement learning-based distant supervision relation extraction for fault diagnosis knowledge graph construction under industry 4.0, *Adv. Eng. Inf.* 55 (2023) 101900.

[25] Z. Huang, X. Guo, Y. Liu, W. Zhao, K. Zhang, A smart conflict resolution model using multi-layer knowledge graph for conceptual design, *Adv. Eng. Inf.* 55 (2023) 101887.

[26] B. Zhou, J. Bao, J. Li, Y. Lu, T. Liu, Q. Zhang, A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops, *Rob. Comput. Integr. Manuf.* 71 (2021) 102160.

[27] X. Li, P. Zheng, J. Bao, L. Gao, X. Xu, Achieving cognitive mass personalization via the self-X cognitive manufacturing network: an industrial knowledge graph-and graph embedding-enabled pathway, *Engineering* (2021).

[28] P. Zheng, L. Xia, C. Li, X. Li, B. Liu, Towards Self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach, *J. Manuf. Syst.* 61 (2021) 16–26.

[29] F. Maibaum, J. Kriebel, J.N. Foege, Selecting textual analysis tools to classify sustainability information in corporate reporting, *Decis. Support Syst.* (2024) 114269.

[30] J. Yang, et al., Harnessing the power of llms in practice: A survey on chatgpt and beyond, *ACM Trans. Knowl. Discov. Data* 18 (6) (2024) 1–32.

[31] M. Xu *et al.*, "A survey of resource-efficient llm and multimodal foundation models," *arXiv preprint arXiv:2401.08092,* 2024.

[32] R. Patil, V. Gudivada, A review of current trends, techniques, and challenges in large language models (llms), *Appl. Sci.* 14 (5) (2024) 2074.

[33] G. Perković, A. Drobnjak, and I. Botički, "Hallucinations in LLMs: Understanding and addressing challenges," in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, 2024: IEEE, pp. 2084-2088.

[34] Q. Guo, S. Cao, Z. Yi, A medical question answering system using large language models and knowledge graphs, *Int. J. Intell. Syst.* 37 (11) (2022) 8548–8564.

[35] J.Z. Pan, et al., "Large Language Models and Knowledge Graphs, Opportunities and Challenges," *arXiv Preprint arXiv:2308.06374* (2023).

[36] K. D. Spurlock, C. Acun, E. Saka, and O. Nasraoui, "ChatGPT for Conversational Recommendation: Refining Recommendations by Reprompting with Feedback," *arXiv preprint arXiv:2401.03605,* 2024.

[37] W. Wang, V.W. Zheng, H. Yu, C. Miao, A survey of zero-shot learning: Settings, methods, and applications, *ACM Transactions on Intelligent Systems and Technology (TIST)* 10 (2) (2019) 1–37.

[38] M. Stewart, M. Hodkiewicz, and S. Li, "Large language models for failure mode classification: an investigation," *arXiv preprint arXiv:2309.08181,* 2023.

[39] Y. Li, "A practical survey on zero-shot prompt design for in-context learning," *arXiv preprint arXiv:2309.13205,* 2023.

[40] J.D. Velásquez-Henao, C.J. Franco-Cardona, L. Cadavid-Higuita, Prompt Engineering: a methodology for optimizing interactions with AI-Language Models in the field of engineering, *Dyna* 90 (230) (2023) 9–17.

[41] M. Cosler, C. Hahn, D. Mendoza, F. Schmitt, C. Trippel, nl2spec: interactively translating unstructured natural language to temporal logics with large language

models, in: *International Conference on Computer Aided Verification*, Springer, 2023, pp. 383–396.
[42] Y. Zhou *et al.*, "Trustworthiness in Retrieval-Augmented Generation Systems: A Survey," *arXiv preprint arXiv:2409.10102,* 2024.
[43] P. Lewis, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, *Adv. Neural Inf. Proces. Syst.* 33 (2020) 9459–9474.
[44] P. Zhao *et al.*, "Retrieval-augmented generation for ai-generated content: A survey," *arXiv preprint arXiv:2402.19473,* 2024.
[45] W. Fan, et al., A survey on rag meeting llms: Towards retrieval-augmented large language models, in: In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 6491–6501.
[46] S. Zeng *et al.*, "The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag)," *arXiv preprint arXiv:2402.16893,* 2024.
[47] O. Diegel, A. Nordin, D. Motte, *A practical guide to design for additive manufacturing*, Springer, 2020.
[48] I. Gibson *et al.*, "Design for additive manufacturing," *Additive manufacturing technologies,* pp. 555-607, 2021.
[49] S. Ghane, R. Sawant, G. Supe, C. Pichad, "LangchainIQ: Intelligent Content and Query Processing," *International Journal of Management, Technology and Social Sciences (IJMTS)* 9 (3) (2024) 34–43.
[50] Y. Wan, Z. Chen, Y. Liu, M. Packianather, R. Wang, in: IEEE, 2023, pp. 1–6.
[51] D. Rau, S. Wang, H. Déjean, and S. Clinchant, "Context Embeddings for Efficient Answer Generation in RAG," *arXiv preprint arXiv:2407.09252,* 2024.
[52] K. Zhu *et al.*, "RAGEval: Scenario Specific RAG Evaluation Dataset Generation Framework," *arXiv preprint arXiv:2408.01262,* 2024.
[53] X. Wang, et al., Searching for best practices in retrieval-augmented generation, in: In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 17716–17736.
[54] M. Ayala-Chauvin and F. Avilés-Castillo, "Optimizing Natural Language Processing: A Comparative Analysis of GPT-3.5, GPT-4, and GPT-4o," *Data and Metadata,* vol. 3, pp. . 359-. 359, 2024.
[55] N. Sinha, V. Jain, and A. Chadha, "Are Small Language Models Ready to Compete with Large Language Models for Practical Applications?," *arXiv preprint arXiv: 2406.11402,* 2024.