Received 30 January 2026
Revised 22 February 2026
Accepted 22 February 2026

An elevator failure mode and effects analysis method based on retrieval augmented generation

Qinfeng Tong
Ningbo Hosting Elevator Co., Ltd, Ningbo, China
Jianwei Chen
China Jiliang University, Hangzhou, China
Yi Zhong and Wei Zhou
Ningbo Hosting Elevator Co., Ltd, Ningbo, China, and
Zuwei Zhou
Zhong'ao Elevator Co., Ltd, Huzhou, China

Abstract

Purpose – To address the limitations of traditional failure mode and effects analysis (FMEA) methods in elevator fault analysis, including heavy reliance on human experience, limited use of large scale heterogeneous text data, static analysis results and insufficient interpretability, this study aims to develop an intelligent FMEA method for the elevator domain.

Design/methodology/approach – An elevator FMEA method based on retrieval augmented generation (RAG) is proposed. An external knowledge base is constructed by integrating a knowledge graph (KG) with a vector database. During retrieval, a multi-route retrieval strategy is adopted to obtain candidate documents. A reranking model named CapsGCN-Rank based on a graph convolutional capsule neural network is designed to perform fine-grained filtering and reranking of candidate documents. The reranked documents are then combined with a large language model to generate structured fault analysis results.

Findings – Experimental results show that the proposed method outperforms several baseline methods in context precision, context recall, as well as the relevance and correctness of generated answers. The method effectively improves the accuracy and completeness of elevator FMEA.

Originality/value – The proposed approach introduces structured semantics from the KG, a multi-route retrieval strategy and dynamic routing of capsule neural networks into the RAG framework. It enables fine-grained document reranking and interpretable fault analysis for the elevator domain, providing an effective solution for FMEA in complex industrial scenarios.

Keywords Elevator, FMEA, Large language model, RAG

Paper type Research article

1. Introduction

Failure mode and effects analysis (FMEA) is a typical risk assessment technique that systematically identifies potential failure modes and evaluates them according to severity and likelihood, thereby supporting maintenance decision-making (Schmitt and Pfeifer, 2015). At present, FMEA has been widely applied in many fields, including mechanical manufacturing (Ervural and Ayaz, 2023), marine engineering (Ceylan and Memiş, 2025) and aerospace engineering (Filz *et al.*, 2021).

Journal of Intelligent Manufacturing and Special Equipment
Vol. 7 No. 2, 2026
pp. 190-206
Emerald Publishing Limited
e-ISSN: 2633-660X
p-ISSN: 2633-6596
DOI 10.1108/JIMSE-01-2026-0004

Funding: This work is supported by Ningbo Innovation Challenge Project of China (NO. 2025YZQ090005).

However, traditional FMEA methods face many challenges in modern elevator maintenance scenarios. In the evaluation process, these methods assign equal weights to failure occurrence probability, detection probability and severity, while such a balance is difficult to achieve in real operating conditions (Filz *et al*., 2021). Meanwhile, maintenance records are often large in volume and poorly structured. Difference in personal experience among maintenance personnel results in inconsistent recording standards, which further reduce data completeness and consistency and hinders systematic assessment (Lu *et al*., 2025). Moreover, traditional FMEA results are typically presented as static tables, lacking reasoning capability and association analysis, which limits the identification of key patterns across failure modes and hinders the provision of clear guidance (Bahr *et al*., 2025).

In elevator scenarios, existing studies have shown that related data often exhibit imbalanced distributions and high complexity. Such data characteristics limit the performance of intelligent fault analysis (Xiao *et al*., 2024). Although some deep learning-based fault diagnosis models alleviate these issues through methods such as feature extraction (Wang *et al*., 2024) and data augmentation (Jiawei *et al*., 2025), their analytical performance still relies heavily on domain knowledge, with limitations in knowledge integration and result interpretation.

In recent years, the emergence of large language models (LLMs) has provided new directions for FMEA. LLMs show strong capabilities in semantic understanding and text summarization, which enables them to extract key information from large scale and structurally complex industrial documents. At the same time, the introduction of retrieval augmented generation (RAG) has further improved the accuracy of LLM-based FMEA. RAG stores knowledge in external databases and performs retrieval and generation during inference, thereby reducing hallucination in LLMs and improving results interpretability (Wu *et al*., 2025). However, existing RAG methods still face difficulties in identifying highly relevant information. Especially in elevator scenarios, data such as failure modes, fault locations and maintenance texts exhibit clear, structured semantic relationships. In practice, these data are often distributed across heterogeneous and fragmented documents, which makes it difficult for retrieval modules to accurately identify core information that is truly related to specific failure modes. Knowledge graph (KG) can organize this dispersed information into a unified semantic network in the form of triples (Lu *et al*., 2024). With the support of graph database query languages, specific entities and their contextual relationships can be precisely retrieved, providing comprehensive and high-quality knowledge support for the reasoning module (Wan *et al*., 2025). By adopting a knowledge graph as the external database, RAG can retrieve information that is more closely aligned with the query intent and perform deeper reasoning based on structured knowledge during generation, thereby producing more complete, reliable and interpretable results for FMEA.

Therefore, we propose an elevator FMEA method based on RAG. The main contributions are as follows:

(1) A multi-route retrieval strategy that integrates KG and vector database is proposed. By combining keyword retrieval based on the KG with similarity retrieval based on the vector database, the relevance and coverage of candidate documents are improved.

(2) A reranking model named CapsGCN-Rank based on a graph convolutional capsule neural network is developed. Multi-scale document capsule representations are constructed by integrating graph convolutional networks with a query-aware attention mechanism to achieve effective alignment between document features and user queries.

(3) Dynamic routing mechanism is introduced to perform fine-grained filtering and reranking of candidate documents, which further improves the contextual quality during the RAG stage.

2. Related work

2.1 Traditional FMEA

The basic procedure of the traditional FMEA is illustrated in Figure 1. First, the scope of FMEA is defined through system analysis, and the system architecture and functions are identified. Second, potential failure modes, failure causes and their effects are analyzed. Then, different failure modes are evaluated from the perspectives of failure occurrence, detection probability and severity. Finally, the risk priority number (RPN) is calculated, and failure modes are ranked according to the RPN to formulate maintenance actions for high-risk failures. For example, Suwankanit (2019) used the FMEA method to analyze the installation process of an elevator, identifying failure modes, their effects and corresponding failure causes during installation.

2.2 Retrieval augmented generation-based FMEA

Traditional FMEA methods are difficult to conduct efficiently in modern industrial systems that involve large volumes of data. To address this issue, many studies have attempted to introduce LLMs into FMEA and have proposed RAG-based FMEA methods. For example, Alenjareghi *et al*. (2026) proposed an LLM-enhanced FMEA method and applied it to safety risk analysis in human–robot collaboration scenarios. Bahr *et al*. (2025) further introduced a KG into the RAG framework, where structured knowledge was used to reduce hallucination in LLMs and thus improve the analytical performance of FMEA.

Although above methods improve the intelligence level of FMEA to some extent, existing studies mainly focus on introducing knowledge at the retrieval stage and lack further filtering and reranking of candidate documents. To address this limitation, we introduced a multi-route retrieval strategy to improve the coverage completeness of candidate documents and further designed a reranking model to enhance the relevance between candidate documents and user queries.

3. Method

To address intelligent fault analysis in elevator systems, we propose an elevator FMEA method based on RAG (KG-CapsGCN-RAG). The overall procedure is shown in Figure 2. First, an external database is constructed by converting the collected text data into vector

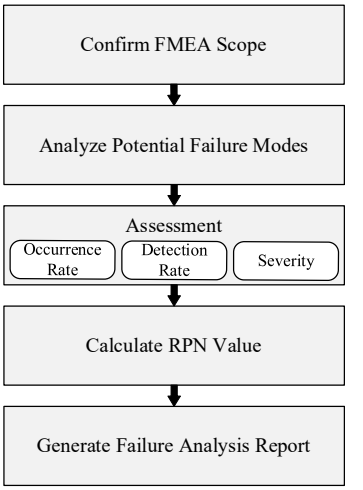

Figure 1. The procedure of the traditional FMEA

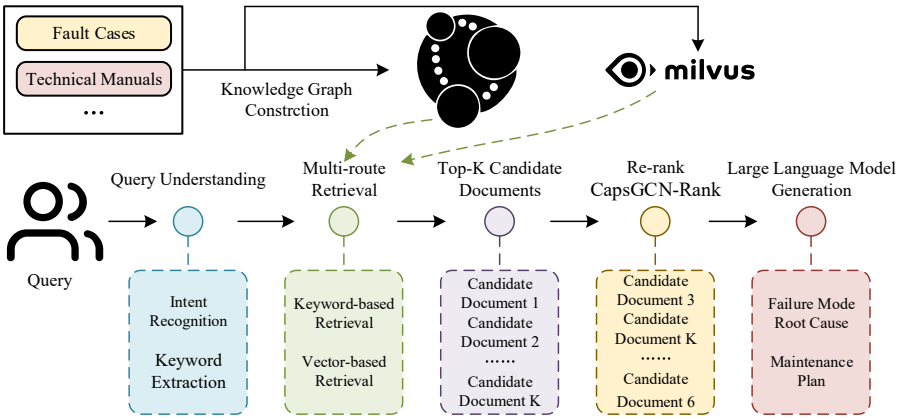

Figure 2. KG-CapsGCN-RAG framework

representations and storing them in a vector database, while a domain KG is built in parallel. Second, within the RAG module, the query is processed through keyword extraction and vector representation. A multi-route retrieval strategy is then adopted, where keyword retrieval is performed on the KG and vector similarity retrieval is conducted in the vector database to obtain candidate documents related to the query. Next, the proposed reranking model CapsGCN-Rank is applied to perform filtering and reranking of the candidate documents. Finally, the reranked documents and the query are jointly fed into an LLM, which performs contextual reasoning and summarization to generate the final FMEA results.

3.1 Knowledge graph embedding

The text data used in this study are expressed in natural language form. Before model inference, the text corpus needs to be embedded by converting text into vector representations, so that semantic information can be captured and similarity retrieval can be performed.

The Moka massive mixed text embedding model (M3E) is trained on a large-scale Chinese sentence pair dataset with tens of millions of samples, and it provides high-quality Chinese text embeddings with strong accuracy in semantic matching and retrieval tasks. Experimental results on multiple Chinese benchmark datasets show that its Normalized Discounted Cumulative Gain at 10 (NDCG@10) scores are higher than those of most baseline models. In addition, the M3E model contains only 110M parameters, which makes it suitable for efficient deployment. In the elevator scenario, most of the processed texts are in Chinese, with only a small amount of English content. Therefore, the M3E text embedding model is selected to embed both the textual data and user queries.

3.2 Multi-route retrieval

The accuracy of candidate documents is a key factor affecting the results of elevator FMEA. Accordingly, we adopt a multi-route retrieval strategy for candidate documents acquisition by combining keyword retrieval based on the KG with similarity retrieval in the vector database. Through the joint use of these two retrieval approaches, the overall performance of the retrieval process is improved.

(1) Keyword retrieval

In the query understanding module, the LLM is used to analyze the user query and extract key entities. The keyword retrieval formulates graph database query statements to retrieve all key

entities from KG and then outputs the corresponding triples. The retrieved triples can be represented as R_{kg}.

(2) Vector similarity retrieval

We evaluate the similarity between the query vector and vectors stored in the vector database using cosine similarity. The core idea is to measure the degree of similarity by computing the cosine value of the angle between two vectors in the vector space.

When a query is received, the M3E text embedding model is used to embed the query and obtain the query vector v_q. The similarity computation between the query vector and vectors in the vector database can be expressed as

$$Similarity(v_q, v_i) = \frac{v_q \cdot v_i}{\|v_q\| \times \|v_i\|} = \frac{\sum_{j=1}^{n} v_q^j \times v_i^j}{\sqrt{\sum_{j=1}^{n} \left(v_q^i\right)^2} \times \sqrt{\sum_{j=1}^{n} \left(v_q^i\right)^2}} \quad (1)$$

where $v_q \cdot v_i$ denotes the dot product between vectors; $\|v_q\|$ and $\|v_i\|$ represent the L2 norms of v_q and v_i, respectively. The output represents the similarity degree between the text and the query.

Finally, based on the similarity scores, the top K most relevant vectors are returned as the output and represented as R_{vec}. The results of keyword retrieval and vector similarity retrieval are then integrated to obtain a set of candidate documents $T = \{t_1, t_2, \ldots, t_K\}$.

3.3 CapsGCN-rank

Because candidate documents obtained at the retrieval stage often exhibit weak relevance and semantic redundancy, this can significantly limit the reasoning quality of the LLM during the generation stage (Xiong *et al.*, 2025). To improve the overall performance of KG-CapsGCN-RAG, reranking of candidate documents is essential. The objective is to further filter and rank documents obtained from multi-route retrieval, extracting more informative and query-relevant content. This study proposes a reranking model based on a graph convolutional capsule neural network (CapsGCN-Rank) to enable effective filtering and accurate reranking of candidate documents. As shown in Figure 3, the model consists of three components. First,

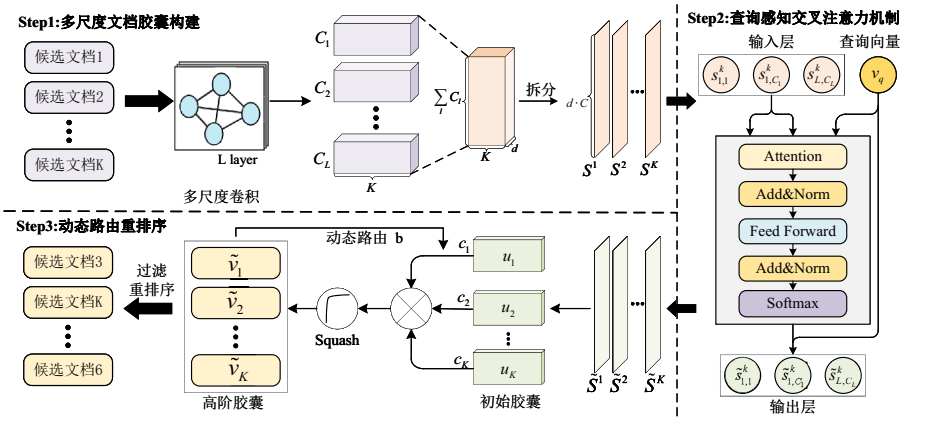

Figure 3. CapsGCN-Rank framework

GCN is used to construct multi-scale document capsules, where candidate documents are taken as input to produce multi-channel semantic representations. Second, a query-aware cross-attention mechanism is introduced to fuse features across different channels by incorporating the query vector, thereby generating primary document capsules. Finally, the dynamic routing reranking is applied to iteratively construct higher-level capsules, and the vector length of the final capsule is used as the similarity criterion between documents and the query, enabling candidate document filtering and reranking.

3.3.1 Multi-scale document capsule construction. In the RAG process, the user query is represented as v_q, and the K candidate documents are denoted as $T = \{t_1, t_2, \ldots, t_K\}$. Their embeddings are given by $V_T = \{v_1, v_2, \ldots v_K\}$, with dimensionality $R^{K\times d}$. In this section, filtering and reranking are performed based on the similarity between the user query and the candidate documents.

Since relations may exist among candidates, documents and GCN can effectively capture such internal features by incorporating semantic graph structures. GCN is applied to process the document embeddings to obtain initial document features:

$$Z^{l+1} = tan\,h\left(\widetilde{D}^{-\frac{1}{2}}(A+I)\widetilde{D}^{-\frac{1}{2}}Z^{l}W^{l}\right) \tag{2}$$

where $W^l \in R^{d\times C_l d}$ is a learnable weight matrix and C_l denotes the number of channels in the lth layer. A is the adjacency matrix, $\widetilde{D}$ denotes the degree matrix corresponding to the adjacency matrix and I is the identity matrix. $Z^l \in R^{K\times(C_l\cdot d)}$ represents the embedding of candidate documents at the lth GCN layer, with the initial value set to V_T.

To more effectively capture interactions among documents, an adaptive adjacency matrix construction method is further introduced, where the adjacency matrix is dynamically built based on document embeddings:

$$A = softmax\left(\left(Z^{l}\mathrm{g}\left(Z^{l}\right)^{T}\right)\Big/t\right) \tag{3}$$

where t is a temperature coefficient used to control the smoothness of the similarity distribution. This adjacency matrix construction method enables the graph structure to be dynamically learned according to semantic similarity among documents, thereby enhancing the capability of GCN to model semantic relationships.

After obtaining the outputs of each GCN layer, they are reorganized into multi-scale document feature representations. For the lth GCN layer, the output is denoted as Z^l, where each channel corresponding to a d dimensional document vector. To construct the initial document capsule, the output of each layer is split by channel into individual channel vectors:

$$s_{l,c}^{k} \in R^{d}, c = 1, 2, \mathrm{K}\ldots C_1\mathrm{K}\ldots, C_l \tag{4}$$

After integrating channel features from all layers, a total of $C = \sum_{l=1}^{L} C_l$ channels are obtained.

3.3.2 Query aware cross-attention mechanism. In GCN, vectors from different layers and different channels describe document features at multiple scales and in different representation spaces. However, the importance of these vectors is not uniform. Some vectors may contain irrelevant semantics or noise, which can affect the performance of subsequent dynamic routing, and moreover, they are not compared with the query vector in terms of similarity.

To address this issue, a query-aware cross-attention mechanism is introduced to apply weighted scaling to each channel vector. This mechanism allows the model to highlight key information related to the query semantics while suppressing redundant components. For the kth candidate document, vectors from all layers and channels are concatenated to form a complete document embedding representation:

$$S^k = concat\left(s^k_{1,1}, \ldots s^k_{1,C_L}, \ldots, s^k_{L,C_L}\right) \tag{5}$$

$$S = \left\{S^1, S^2, \ldots S^K\right\} \tag{6}$$

To distinguish the importance of different channels and their correlation with the query vector, we perform computation using a cross-attention mechanism. Specifically, for the *k*th candidate document, the query vector and the key vector are constructed as

$$Q = W_q v_q, K^k = W_k S^k \tag{7}$$

where W_q and W_k are learnable weight matrices. Q is only related to the query and remains the same for all candidate documents. K^k is determined by the specific candidate document.

Then, based on the query vector and the key vector, the attention weights are computed as follows:

$$\alpha^k = softmax\left(\frac{Q\left(K^k\right)^T}{\sqrt{d_a}}\right) \tag{8}$$

the computed $\alpha^k \in R^C$ contains attention weights of the C channels. According to these attention weights, each channel vector is scaled as

$$\widetilde{s}^k_{l,c} = \alpha^k_c \cdot s^k_{l,c} \tag{9}$$

After concatenating all attention results, a complete document embedding is obtained $\widetilde{S}^k = concat(\widetilde{s}^k_{1,1}, \ldots, \widetilde{s}^k_{1,C_1}, \ldots \widetilde{s}^k_{L,C_L})$, and all candidate documents can be represented as $\widetilde{S} = \{\widetilde{S}^1, \widetilde{S}^2, \ldots \widetilde{S}^K\}$. Each element has dimensionality $R^{C \cdot d}$.

3.3.3 Dynamic routing Re-ranking. Subsequently, the document embeddings are further updated by incorporating the dynamic routing mechanism. First, the document embeddings are expanded to generate an initial document capsule set $U = \{u_1, u_2, \ldots u_K\}$, where $u_k \in R^{(C \cdot d) \times m}$ and m denote the capsule dimension. Then, prediction vectors are computed through an affine transformation:

$$\widetilde{u}_{j|k} = W_{jk} u_k \tag{10}$$

where $\widetilde{u}_{j|k}$ represents the prediction vector of the *k*th document capsule with respect to the *j*th higher-level capsule. $W_{jk} \in R^{m \times m}$ is a learnable weight matrix. After aggregating all prediction vectors, the higher-level capsule can be obtained as

$$\widetilde{v}_j = squash\left(\sum_{k=1}^{K} c_{jk} \widetilde{u}_{j|k}\right) \tag{11}$$

where $squash(\cdot)$ is an activation function. c_{jk} denotes the coupling coefficient, which is used to describe the contribution of a document capsule to a higher-level capsule. For each document capsule u_k, the sum of its coupling coefficients with all higher-level capsules are initialized to 1. The coupling coefficient c_{jk} is computed through the dynamic routing mechanism as follows:

$$c_{jk} = \frac{\exp(b_{jk})}{\sum_{n=1}^{K} \exp(b_{jn})} \tag{12}$$

where the initial value of b_{jk} is set to 0 and is dynamically updated during the routing iterations according to the computation results of the higher-level capsules. The update process is realized through the accumulation of $\widetilde{u}_{j|k} \cdot \widetilde{v}_j$.

Through this dynamic routing mechanism, document capsules and higher-level capsules are able to maintain feature consistency during the iterative process. This enables the model to strengthen document capsules with higher contribution degrees while suppressing noise, thereby generating higher-level document semantic representations for ranking.

Finally, the loss function is designed to minimize the cosine distance between the query vector and the higher-level capsule, so as to constrain the semantic direction of the higher-level capsule to be consistent with the query:

$$L = 1 - \frac{v_q \cdot \widetilde{v}_j}{|v_q| \cdot \left|\widetilde{v}_j\right|} \tag{13}$$

3.4 Reranking

In capsule neural networks, the length of a higher-level capsule vector is usually used to represent the probability strength of the features expressed by the capsule. In reranking tasks, the direction of the higher-level capsule vector is more capable of reflecting the relevance between candidate documents and the query. Therefore, the importance of candidate documents with respect to the query is determined according to the direction of the higher-level capsule, and reranking is performed accordingly.

For each candidate document t_k, a higher-level capsule vector $\widetilde{v}_k$ is computed. Based on the cosine similarity defined in the loss function, the similarity between $\widetilde{v}_k$ and the query vector v_q is computed as follows:

$$score(t_k) = \frac{v_q \cdot \widetilde{v}_k}{|v_q| \cdot \left|\widetilde{v}_k\right|} \tag{14}$$

A higher score indicates a stronger relevance between the document and the query. Subsequently, candidate documents are filtered and reranked according to a threshold, and the final document set is obtained as

$$\widetilde{T} = rank(T, score, \gamma) \tag{15}$$

where $\widetilde{T}$ denotes the final document set and γ represents the threshold. Documents with scores lower than the threshold are filtered out. In the generation stage, the reranked documents are integrated with the query to jointly form the contextual input to the LLM, which performs deeper integration and outputs structured standard answers.

4. Experiments

4.1 Evaluation metrics

To evaluate the accuracy of the conclusions of elevator FMEA, the experiments adopt the RAGas framework for performance evaluation. RAGas is a framework specifically designed for evaluating the performance of RAG systems, which involves multiple metrics:

Context precision (CP): CP is used to measure the relevance of retrieved context fragments to the answers of a given question. Its formulation is expressed as follows:

$$Context\ Precision@k = \frac{\sum precision@k}{total\ number\ of\ relavent\ items\ in\ the\ top\ K\ results} \tag{16}$$

$$Precision@k = \frac{true\ positives@k}{(true\ positives@k + false\ positives@k)} \tag{17}$$

where *Precision@k* represents the proportion of positive results among the top *k* retrieval results.

Context recall (CR): CR is used to measure the consistency between the retrieved context and the ground truth provided by humans. Its formulation is expressed as follows:

$$context\ recall = \frac{|GT\ sentences\ that\ can\ be\ attributed\ to\ context|}{|number\ of\ sentences\ in\ GT|} \tag{18}$$

where GT denotes the ground truth. In this formula, the numerator represents the number of sentences in the retrieved context that are relevant to the ground truth, while the denominator represents the total number of sentences in the ground truth.

Answer relevancy (AR): AR is used to measure the degree of association between the generated answer and the user query. To compute this metric, RAGas invokes LLM to infer possible questions in reverse based on the generated answer and then calculates the average cosine similarity between all inferred questions and the user query. The core idea is that if the generated answer can respond to the user query, the inferred questions and the user query will exhibit a relatively high similarity.

Answer correctness (AC): AC is used to measure the similarity between the generated answer and the ground truth. To compute this metric, RAGas allows LLM to read both the generated answer and the ground truth simultaneously and then evaluates the degree of semantic consistency and factual consistency between them. The final AC score is obtained through weighted computation.

In this study, CP and CR are used to evaluate the candidate documents obtained from retrieval and reranking, so as to measure the relevance and coverage completeness of the candidate documents. At the same time, AR and AC are used to evaluate the conclusions of elevator FMEA obtained in the generation stage, measuring the quality of the generated answers from the perspectives of semantic consistency and factual correctness. By combining these four metrics, the overall performance of the RAG framework across retrieval and generation processes can be comprehensively evaluated.

4.2 Dataset construction

To evaluate the accuracy of FMEA, we construct a question answering evaluation dataset that conforms to the RAGas framework. For question design, the dataset covers multiple types of questions, including simple question answering and complex reasoning, so as to ensure diversity in question types. At the same time, all questions and ground truth are specifically designed based on the actual data used in RAG. To further improve data quality, all proposed questions and ground truth undergo multiple rounds of screening and are validated by domain experts to ensure their accuracy and reliability.

Finally, based on the above procedure, an evaluation dataset consisting of 100 question answering samples is constructed. Some examples are shown in Figure 4.

```
{
    "question": "导靴的靴衬磨损后不及时更换会导致什么问题?",
    "grading_notes": "导靴的靴衬磨损后不及时更换会引起电梯的水平位移，导靴润滑不良、变形、锈蚀靴衬磨损:导靴与导轨配合间隙过小会导致电梯运行有摩擦产生噪声，间隙过大则会导致轿厢振动、晃动。严重时会导致电梯平层故障。"
},
{
    "question": "制动轮表面损伤的检验主要包括哪些方面？",
    "grading_notes": "制动轮表面损伤的检验主要包括四个方面：一是外观检查，以肉眼近距离观察制动轮表面是否存在划痕、沟槽、磨损区域或剥落等缺陷，并注意这些损伤是否由异物进入或闸瓦异常摩擦引起；二是测量制动轮直径，使用专业量具在制动轮圆周多个位置测量并与原始设计尺寸对比，判断直径偏差是否超出规定范围从而识别磨损情况；三是检查制动轮表面硬度，利用硬度测试设备在不同部位测量硬度值，评估材料性能是否均匀可靠；四是检测制动轮的圆度和圆柱度是否达标，以避免闸瓦与制动轮接触不良，在制动时产生振动和噪声，进而影响制动的平稳性和可靠性。"
},
{
    "question": "若制动轮直径偏差超出规定范围，将对制动性能产生什么影响？",
    "grading_notes": "当制动轮直径偏差超出规定范围时，说明制动轮已经发生明显磨损，这会改变制动轮与闸瓦之间的接触状态，使接触压力和摩擦力发生变化，从而削弱制动效果，可能导致制动力不足或不稳定，进而严重影响电梯制动系统的安全性和可靠性。"
},
{
    "question": "电梯制动系统中，电磁铁故障的检验主要应从哪些方面进行？",
    "grading_notes": "电磁铁故障的检验主要包括外观检查、电气性能测试以及动作状态观察等内容。首先应查看电磁铁外壳是否存在破损、变形，连接部位是否松动，因为外壳损坏会影响防护性能，而连接松动会导致接触不良。随后需测量线圈电阻，将测量值与额定电阻对比判断是否存在短路或断路，短路会导致电流异常增大烧毁线圈，而断路则使电磁铁无法正常通电工作。在电梯运行中还需观察电磁铁吸合和释放动作是否及时顺畅，若存在迟缓或卡顿应排查原因，同时还需手动操作以感受其动作灵活性，判断是否存在内部卡滞问题，并检查铁芯表面是否光滑、有无磨损或锈蚀，因为这些缺陷会削弱磁力，导致制动效果变差。通过上述全面检查可及时发现电磁铁隐患，确保制动系统安全可靠运行。"
},
{
    "question": "若电磁铁线圈电阻偏离额定值，会造成哪些影响？",
    "ground_truths": "当电磁铁线圈电阻与额定值偏差较大时，通常意味着线圈发生短路或断路。短路会导致电流异常增大，使线圈过热甚至烧毁；断路则会使电磁铁无法正常通电，无法顺利吸合或释放，从而导致制动装置不能按要求动作，影响电梯制动系统的安全性和可靠性。"
},
{
    "question": "制动弹簧疲劳或断裂的检验主要包括哪些方面？",
    "ground_truths": "制动弹簧的检验应从外观检测、尺寸测量、性能测试、无损检测以及运行状态观察等方面进行。首先需借助强光和放大镜查看弹簧表面是否存在裂纹、锈蚀或明显变形，因为裂纹是断裂隐患，锈蚀会削弱强度，变形说明内部结构已发生改变。其次测量弹簧自由高度并与设计标准比对，偏差超限可能表明弹簧已疲劳；同时通过专业设备测试弹簧刚度，若与规定值不符则表明弹性性能发生变化，存在疲劳风险。对于关键部位弹簧，还需采用磁粉或超声波探伤等无损检测技术检查其内部缺陷。最后在电梯运行中观察弹簧伸缩动作是否顺畅，听是否有异常声响，若出现卡顿或异响则可能存在问题，需要进一步检查，以确保制动系统可靠运行。"
},
{
    "question": "若制动弹簧的自由高度或刚度与设计标准不符，通常说明什么问题？",
    "ground_truths": "当制动弹簧的自由高度或刚度偏离设计标准时，通常说明弹簧已经出现疲劳或弹性性能下降。自由高度超出允许偏差可能表明弹簧经历长期载荷后发生永久变形，而刚度不符合规定则说明其弹性特性发生改变，无法提供正常的制动力度。这些变化意味着弹簧存在潜在疲劳风险，可能影响制动系统的可靠性，需要及时评估并更换。"
},
{
    "question": "制动闸瓦磨损的检验需要从哪些方面进行？",
    "ground_truths": "制动闸瓦磨损的检验应从外观观察、厚度测量、磨损均匀性判断以及连接稳固性检查等方面展开。首先通过直接观察闸瓦表面是否存在磨损沟壑、剥落等现象以初步判断磨损程度。随后使用卡尺等精密量具测量闸瓦厚度，并与厂家规定的原始厚度及最小允许厚度进行对比，若接近或低于最小允许厚度则需及时更换。还应关注闸瓦磨损的均匀性，通过查看制动状态下闸瓦与制动轮接触面摩擦痕迹的分布情况，若出现局部颜色差异或磨损深浅不一致，则可能存在不均匀磨损，需要进一步排查安装是否正确或制动轮表面是否平整。最后检查闸瓦制动衬垫与固定部件的连接是否牢固，防止因磨损导致松动影响制动效果。只有全面严格检查这些要点，才能及时发现潜在隐患，确保电梯制动系统安全可靠运行。"
},
```


Figure 4. Dataset examples

4.3 Experimental environment and parameter settings

In this study, m3e-base was used as the text embedding model to encode both queries and documents. Deepseek-v3.2 was selected as the baseline LLM of KG-CapsGCN-RAG to perform question understanding and answer generation. During the training of the CapsGCN-Rank reranking model, the multi-route retrieval was configured to retrieve 10 relevant documents for each query. The number of training epochs was set to 10, learning rate was set to 5e−5 and Adam was selected as the optimizer. In addition, the number of GCN layers was set to 3, with the number of channels C_l in each layer set to 2. The number of dynamic routing iterations in the capsule neural network was set to 3. The experimental environment and the remaining parameter settings in this section are shown in Table 1.

Finally, the following RAG frameworks are selected for comparison in this study:

NaiveRAG (Gao *et al.*, 2023): NaiveRAG is a standard baseline model of existing RAG frameworks. This method is based on vector retrieval, where the original documents are sliced and encoded to obtain similar document information.

GraphRAG (Edge *et al.*, 2024): GraphRAG is a KG–based RAG method. It summarizes document content by constructing graph level and node level community reports and performs RAG by incorporating the community reports.

LightRAG (Guo *et al.*, 2024): LightRAG is a KG-based RAG method. It adopts a two-level retrieval architecture that combines lower level specific entity retrieval with higher level topic retrieval, thereby improving retrieval coverage.

4.4 Results

4.4.1 Baseline Study. The overall experimental results are shown in Figure 5. The results indicated that, as the RAG method gradually shifts from a retrieval strategy that relies only on vector similarity to a strategy that integrates a KG, the overall performance of RAG is clearly improved. GraphRAG, LightRAG and KG-CapsGCN-RAG all introduced a KG in the retrieval stage and achieved better performance compared with NaiveRAG, which relied only on vector similarity retrieval. Furthermore, by introducing a multi-route retrieval strategy, LightRAG and KG-CapsGCN-RAG can retrieve candidate documents with broader coverage, thereby improving the overall quality of the retrieved context and enhancing the quality of the generated answers. Their overall performance was superior to that of GraphRAG, which relied on a single graph retrieval strategy. On this basis, KG-CapsGCN-RAG further introduced a reranking model. For the task of elevator FMEA, CapsGCN-Rank was designed by combining capsule neural networks with GCN, which performs fine-grained filtering and reranking of candidate documents, thereby improving question answering performance in complex engineering scenarios.

Table 1. Experimental environment and parameter settings

Configuration	Parameter
CPU	Intel i9 14900K
GPU	NVIDIA RTX 4090
Memory	128 GB
System	Windows 11
Python	3.10
RAGas	0.3.6
Graph Database	Neo4j 5.26.0
Embedding Model	m3e-base
Embedding Dimension	768

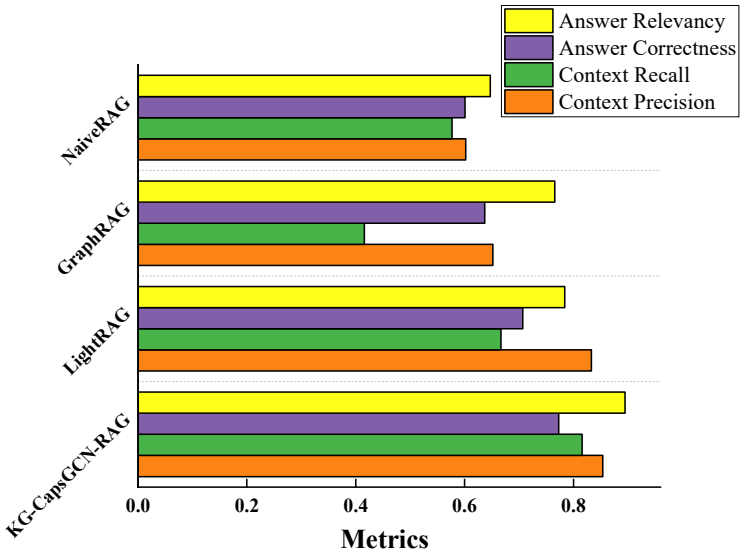

Figure 5. Baseline study

4.4.2 Ablation study. In this section, a series of ablation experiments were conducted to verify the impact of each component in the proposed method. The experiment consists of the following control groups: (1) proposed KG-CapsGCN-RAG; (2) KG-CapsGCN-RAG without keyword retrieval, denoted as w/o KG; (3) KG-CapsGCN-RAG without vector similarity retrieval, denoted as w/o Vector; (4) KG-CapsGCN-RAG without CapsGCN-Rank, denoted as w/o Rerank.

The experimental results are shown in Table 2. It can be observed that all components in KG-CapsGCN-RAG contribute positively to the improvement of RAG performance. First, when the CapsGCN-Rank was removed, the performance of RAG showed a clear decline. This result indicated that although the multi-route retrieval can provide diverse candidate documents, it inevitably included many redundant documents and low relevance information, which directly affect the generation performance of LLM. Both w/o KG and w/o Vector retained the CapsGCN-Rank, and their performance was clearly better than that of w/o Rerank, which further demonstrated that the proposed CapsGCN-Rank plays a crucial role in the overall RAG process.

Further, both w/o KG and w/o Vector rely on a single route retrieval. As a result, the candidate documents obtained by these methods show weaker coverage of relevant information compared with the multi-route retrieval strategy, leading to lower results in *AC* and *AR* than the complete method. The results of these two experiments further

Table 2. Ablation study

Method	AC	AR
KG-CapsGCN-RAG	0.773	0.895
w/o KG	0.722	0.767
w/o Vector	0.709	0.746
w/o Rerank	0.679	0.759

indicated that vector similarity-based retrieval plays a more important role than KG-based keyword retrieval, as it can match a wider range of relevant documents, while the latter can provide more accurate triple information based on the KG structure. By combining these two retrieval methods, more sufficient and precise candidate documents can be obtained.

4.4.3 LLM comparative study. In the generation stage, the performance of the LLM used is a key factor affecting the accuracy of the final answers. A stronger model can accurately understand the question based on the prompts and effectively integrate the retrieved candidate documents. Therefore, Qwen3-Max, DeepSeek-V3.2, KiMi-K2 and GLM-4.6 were selected as baseline models for comparison. In addition, since the number of model parameters is an important indicator for evaluating LLM performance, we further investigated the impact of parameter scale on RAG performance by using different parameter versions of the same model. The objective of this experiment was to maintain RAG accuracy while adopting models with smaller parameter scales for inference, so as to reduce token consumption. In the experiments, DeepSeek-R1 was used as the baseline model, and its full version with 671B parameters, as well as versions with 32B, 14B, 7B and 0.5B parameters, was evaluated. All models were invoked through the Alibaba Cloud Bailian Application Programming Interface (API).

Figure 6 presented the comparison results of different models in AC, AR and average inference time. As shown in Figure 6(a), DeepSeek-V3.2, Qwen3-Max, KiMi-K2 and GLM-4.6 exhibited similar performance on AC and AR. These four models were full-parameter versions of their respective architectures and possessed strong capabilities in semantic understanding and text generation. With appropriate prompts and sufficient and accurate candidate documents support, the models can effectively integrate retrieved information, summarize the given text and produce accurate responses. Regarding inference time, DeepSeek-V3.2, Qwen3-Max and GLM-4.6 can complete inference within a relatively short

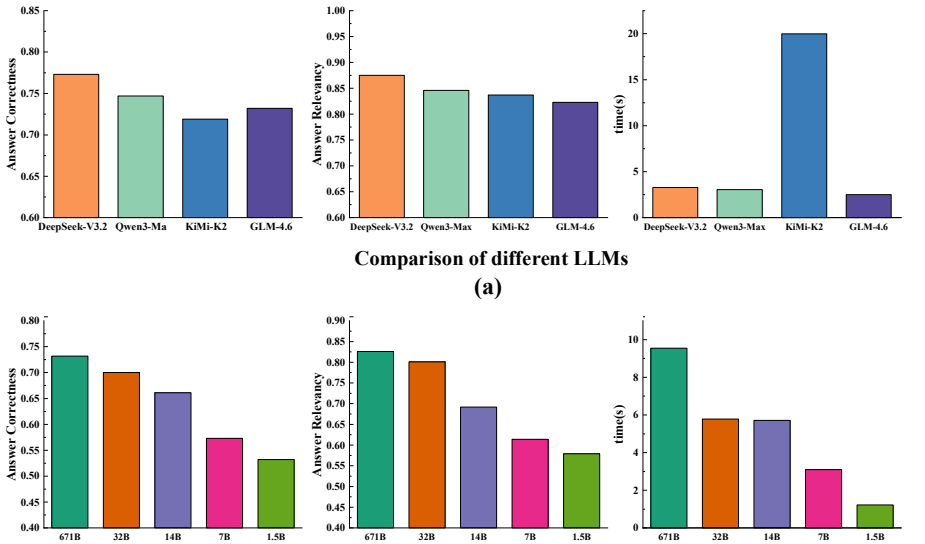

Figure 6. LLM comparative study

time, whereas KiMi-K2 required a longer inference time. This was because KiMi-K2 incorporated a deep reasoning mechanism during inference. However, in the application scenario considered in this study, the task mainly focused on text understanding and summarization, and the deep reasoning capability did not lead to a clear advantage in generation performance.

Figure 6(b) shows the experimental results of DeepSeek-R1 under different parameter scales. The results indicated that, as the number of model parameters decreased, the performance on AC and AR in the generation stage exhibited a gradual decline. Although reducing the model scale led to some reduction in inference time, the improvement was limited and insufficient to compensate for the performance degradation caused by the decrease in parameter scale. These results indicated that, in elevator scenario, models with larger parameter scales can better exploit semantic understanding and produce more accurate FMEA results.

4.4.4 Visualization analysis. Finally, to present the performance of KG-CapsGCN-RAG in a more intuitive manner, a visualization analysis was conducted on its question answering results in practical scenarios. Taking the question "Which aspects are mainly involved in the inspection of elevator brake spring fatigue or fracture?" as an example, the proposed method was compared with commonly used general LLMs, including Qwen3-Max, DeepSeek-V3.2 and GPT5.1. The *AC* and *AR* of the answers generated by different models were calculated based on the reference answer.

As shown in Figure 7, KG-CapsGCN-RAG achieved the best performance on both metrics. Its generated answers showed advantages in content coverage, completeness and conciseness. This was mainly because the reference knowledge was processed by different methods and stored in the KG and the vector database, which enabled the retrieval stage to query relevant knowledge around the user question and provide contextual support for the generation stage.

The other three models also achieved relatively high AR values, indicating that their responses can answer the question and identify key inspection aspects such as visual inspection, dimensional measurement and performance testing. However, from the perspective of AC, the answers generated by these models still exhibited varying degrees of information omission and detail deviation compared with the ground truth. For example, the response generated by Qwen3-Max did not mention non-destructive testing. Although DeepSeek-V3.2 and GPT5.1 mentioned non-destructive testing, they did not further specify inspection techniques such as ultrasonic testing, leading to differences in key implementation details. In addition, none of the above models mentioned abnormal noise during spring operation. These differences ultimately resulted in lower AC values.

5. Conclusions

This study proposed KG-CapsGCN-RAG, an elevator FMEA method based on RAG. The overall framework of the proposed method consisted of four key steps: (1) construction of KG and vector database, (2) retrieval of candidate documents by combining a multi-route retrieval strategy, (3) filtering and ranking of candidate documents using the proposed CapsGCN-Rank reranking model and (4) generation of structured fault analysis results based on LLM.

In KG-CapsGCN-RAG, the CapsGCN-Rank reranking model constructed primary document capsules by integrating multi-scale convolution and a query aware cross-attention mechanism and evaluated document similarity through the dynamic routing mechanism. This process enabled fine-grained filtering and reranking of candidate documents. Experimental results showed that KG-CapsGCN-RAG outperformed existing RAG methods such as GraphRAG and LightRAG. In addition, the CapsGCN-Rank reranking model provided high-quality candidate documents that were highly relevant to

Question	Example Which aspects are mainly involved in the inspection of elevator brake spring fatigue or fracture?	Ground Truth	The inspection of brake springs should include visual inspection, dimensional measurement, performance testing, non-destructive testing, and observation of operating conditions. First, strong lighting and a magnifying glass should be used to check whether cracks, corrosion, or obvious deformation are present on the spring surface, since cracks indicate potential fracture risks, corrosion reduces mechanical strength, and deformation suggests changes in the internal structure. Next, the free height of the spring should be measured and compared with the design specifications, as excessive deviation may indicate spring fatigue; meanwhile, specialized equipment should be used to test spring stiffness, and values that do not meet the specified requirements indicate changes in elastic performance and potential fatigue risk. For springs at critical positions, non destructive testing methods such as magnetic particle testing or ultrasonic testing should be applied to detect internal defects. Finally, during elevator operation, the extension and compression behavior of the spring should be observed to assess whether the motion is smooth, and abnormal noise should be monitored; if jamming or unusual noise occurs, potential problems may exist and further inspection is required to ensure reliable operation of the braking system.
KG-CapsGCN-RAG	Key inspection points for elevator brake springs with fatigue or fracture mainly include the following aspects. Visual inspection involves using strong lighting and a magnifying glass to examine whether there are cracks, corrosion, or obvious deformation on the spring surface, where micro cracks may become fracture risks and corrosion can weaken material strength. Dimensional measurement requires measuring the free height under no load conditions and comparing it with the design specifications, while performance testing includes measuring spring stiffness to evaluate changes in elastic performance. Non-destructive testing applies techniques such as magnetic particle testing and ultrasonic testing to detect hidden internal defects. Observation of operating conditions focuses on whether abnormal noise occurs during operation and whether the extension and compression of the spring are smooth. Answer Correctness : 0.778 Answer Relevancy : 0.798	Qwen3-Max	The inspection of elevator brake spring fatigue or fracture mainly included the following aspects: visual inspection, observing whether cracks, deformation, corrosion, wear, or fracture traces were present on the spring; dimensional measurement, checking whether the free height, wire diameter, and other dimensions met the original manufacturer specifications to determine whether permanent deformation had occurred due to fatigue; performance testing, examining whether the compression and recovery of the spring during brake operation were smooth and sufficiently strong, and ensuring that the braking torque met safety requirements; material property evaluation (when necessary), assessing whether material aging or degradation had occurred through hardness testing or metallographic analysis; operation record analysis, evaluating the service life and accumulated fatigue of the spring by combining elevator maintenance records and fault history. The inspection should be conducted in accordance with the Elevator Supervision Inspection and Periodic Inspection Rules and relevant manufacturing standards, and any abnormalities detected should be addressed by timely replacement to prevent braking failure and related safety accidents. Answer Correctness : 0.669 Answer Relevancy : 0.792
DeepSeek V3.2	The inspection mainly covered the following aspects. Visual inspection focused on observing whether cracks, corrosion, permanent deformation, or fracture traces were present on the spring surface, with particular attention paid to the end coils and contact areas with supports. Dimensional measurement and performance testing involved measuring the free height, compressed height, and stiffness variation to evaluate whether the spring met the design requirements for load and deformation. Non-destructive testing was applied to critical springs using magnetic particle testing or penetrant testing to detect surface and near surface micro cracks. Monitoring of operating conditions examined the opening and closing clearance of the brake and its response time, where abnormalities often indicated degradation in spring performance. Disassembly and in depth analysis were conducted for failed springs, in which fracture analysis, such as scanning electron microscopy, was used to determine the fracture mode, including fatigue, overload, or corrosion. It was recommended to combine inspection results with routine maintenance records, and to conduct focused inspection and preventive replacement for springs that had reached the designed number of load cycles or had experienced abnormal operating conditions. Answer Correctness : 0.612 Answer Relevancy : 0.603	ChatGPT（GPT5.1）	The inspection of elevator brake spring fatigue or fracture mainly included the following aspects:(1) Visual inspection: observing whether the spring exhibited fatigue related features such as cracks, corrosion, deformation, or loosening of end coils.(2) Dimensional measurement and deformation assessment: measuring the free length and working length of the spring to determine whether permanent deformation or abnormal compression had occurred, and comparing the results with manufacturer specifications.(3) Performance testing of elastic force and stiffness: using testing equipment to measure the load displacement characteristics of the spring in order to evaluate whether the elastic force had decreased and whether stiffness had degraded, thereby assessing the degree of fatigue.(4) Non destructive testing: when necessary, applying methods such as magnetic particle testing or penetrant testing to detect micro cracks or hidden damage and to determine potential fracture risks.(5) Installation condition inspection: confirming whether the spring was correctly installed in the brake mechanism and whether eccentric loading or uneven wear was present, so as to prevent stress concentration that could accelerate fatigue. These inspections allowed a comprehensive evaluation of spring performance degradation and enabled timely identification of fatigue damage or potential fracture hazards. Answer Correctness : 0.597 Answer Relevancy : 0.795

Figure 7. Visualization

the query, which verified the effectiveness of the proposed method for elevator FMEA tasks.

References

Alenjareghi, M.J., Ghorbani, F., Keivanpour, S., Chinniah, Y.A. and Jocelyn, S. (2026), "Proactive safety reasoning in human-robot collaboration in disassembly through LLM-augmented STPA and FMEA", *Robotics and Computer-Integrated Manufacturing*, Vol. 98, 103162, doi: 10.1016/j.rcim.2025.103162.

Bahr, L., Wehner, C., Wewerka, J., Bittencourt, J., Schmid, U. and Daub, R. (2025), "Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis", *Journal of Industrial Information Integration*, Vol. 45, 100807, doi: 10.1016/j.jii.2025.100807.

Ceylan, B.O. and Memiş, S. (2025), "Fuzzy parameterized fuzzy soft matrices-based failure mode and effects analysis (FPFS-FMEA) with ship lubricating oil system risk assessment", *Ocean Engineering*, Vol. 342, 123049, doi: 10.1016/j.oceaneng.2025.123049.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. and Larson, J. (2024), "From local to global: a graph rag approach to query-focused summarization", *arXiv Preprint*, arXiv:2404.16130.

Ervural, B. and Ayaz, H.I. (2023), "A fully data-driven FMEA framework for risk assessment on manufacturing processes using a hybrid approach", *Engineering Failure Analysis*, Vol. 152, 107525, doi: 10.1016/j.engfailanal.2023.107525.

Filz, M.-A., Langner, J.E.B., Herrmann, C. and Thiede, S. (2021), "Data-driven failure mode and effect analysis (FMEA) to enhance maintenance planning", *Computers in Industry*, Vol. 129, 103451, doi: 10.1016/j.compind.2021.103451.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J. and Wang, H. (2023), "Retrieval-augmented generation for large language models: a survey", *arXiv Preprint*, arXiv:2312.10997 2(1).

Guo, Z., Xia, L., Yu, Y., Ao, T. and Huang, C. (2024), "Lightrag: simple and fast retrieval-augmented generation", *arXiv Preprint*, arXiv: 2410.05779.

Lu, J., Li, J., Li, W., Song, J. and Xiao, G. (2024), "Heterogeneous propagation graph convolution network for a recommendation system based on a knowledge graph", *Engineering Applications of Artificial Intelligence*, Vol. 138, 109395, doi: 10.1016/j.engappai.2024.109395.

Jiawei, L., Zhang, W., Lu, C., Xiao, G. and Wang, Q. (2025), "A multi-scale convolution capsule network with data augmentation and attention mechanisms for elevator fault diagnosis", *ISA Transactions*, Vol. 167, pp. 1873-1887, doi: 10.1016/j.isatra.2025.09.041.

Lu, J., Chen, H., Chen, J., Xiao, Z., Li, R., Xiao, G. and Wang, Q. (2025), "Temporal knowledge graph fusion with neural ordinary differential equations for the predictive maintenance of electromechanical equipment", *Knowledge-Based Systems*, Vol. 317, 113450, doi: 10.1016/j.knosys.2025.113450.

Schmitt, R. and Pfeifer, T. (2015), *Qualitätsmanagement: Strategien–Methoden–Techniken*, Carl Hanser Verlag GmbH Co KG, München.

Suwankanit, T. (2019), "The identification of failure modes in the elevator installation process of a case company in Thailand by FMEA", *London Journal of Research of Engineering Research*, Vol. 19 No. 4, pp. 21-28.

Wan, Y., Chen, Z., Liu, Y., Chen, C. and Packianather, M. (2025), "Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing", *Advanced Engineering Informatics*, Vol. 65, 103212.

Wang, Q., Chen, L., Xiao, G., Wang, P., Gu, Y. and Lu, J. (2024), "Elevator fault diagnosis based on digital twin and PINNs-e-RGCN", *Scientific Reports*, Vol. 14 No. 1, 30713, doi: 10.1038/s41598-024-78784-7.

Wu, W., Wang, H., Li, B., Huang, P., Zhao, X. and Lei, L. (2025), "Multirag: a knowledge-guided framework for mitigating hallucination in multi-source retrieval augmented generation", *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, IEEE.

Xiao, G., Gu, H., Dong, J., Wang, Q. and Lu, J. (2024), "Simulation data-driven migration diagnosis method for guide rail faults in long-term service elevator", *China Mechanical Engineering*, Vol. 35 No. 01, p. 125.

Xiong, Y., Tu, X. and Zhao, W. (2025), "AFR-Rank: an effective and highly efficient LLM-based listwise reranking framework via filtering noise documents", *Information Processing and Management*, Vol. 62 No. 6, 104232, doi: 10.1016/j.ipm.2025.104232.

Corresponding author
Jianwei Chen can be contacted at: cjw137118@163.com