This is the Pre-Published Version.
The following publication K. Sun, Y. Han, Z. Zhao and G. Q. Huang, "Enhancing Large Language Models for Fashion Smart Manufacturing via Dynamic Collaborative Routing-Based Retrieval Reranking," in IEEE Transactions on Cybernetics, vol. 56, no. 3, pp. 1282-1295, March 2026 is available at https://doi.org/10.1109/TCYB.2025.3624234.

Enhancing Large Language Models for Fashion Smart Manufacturing via Dynamic Collaborative Routing-Based Retrieval Reranking

Kexin Sun, Yujie Han, Zhiheng Zhao*, *Senior Member, IEEE*, George Q. Huang, *Fellow, IEEE*

***Abstract*—Enhancing large language models (LLMs) with external knowledge base retrieval in the fashion manufacturing industry can provide more reliable technical support and decision-making assistance, significantly improving process control and boosting intelligent production efficiency. However, the field of fashion manufacturing involves highly specialized terminology, logically complex technical knowledge, and intricate query tasks. Existing simple query-matching techniques often return a large number of contextually loose and redundant document chunks, severely impacting the model's understanding and response quality. To address this issue, this paper proposes a retrieval optimization framework based on Dynamic Capsule Routing Network with embedded Semantic Graph (SGDCR), which models semantic relations among multiple retrieved documents by simulating a team collaboration mechanism. Specifically, the framework consists of two steps: filtering and reranking. First, a capsule routing mechanism embedded in a semantic association graph dynamically captures complex contextual relationships among coarse-grained document blocks, learns contribution scores for multiple documents, and filters irrelevant or redundant documents based on ranking. Subsequently, the filtered documents are matched with the query through deep semantic similarity measurement, and the documents are reranked by integrating relevance scores and contribution scores, generating efficient, accurate and contextually coherent document prompts. Experimental results on publicly available dense open-domain QA datasets and a constructed fashion manufacturing process QA dataset demonstrate the effectiveness and superiority of the proposed method over existing reranking approaches in the fashion manufacturing knowledge QA system.**

***Index Terms*—Fashion Manufacturing, Large Language Models, Retrieval-Augmented Generation, Retrieval Reranking, Dynamic Routing.**

I. Introduction

THE fashion production and manufacturing process is highly complex and multi-staged, encompassing various phases such as raw material selection, design prototyping, cutting and sewing, quality inspection, and final product packaging [1]. Each stage requires strict technical specifications and meticulous operational procedures to ensure the quality and consistency of the final product. Moreover, fashion manufacturing is influenced by multiple external factors, including market demand, fashion trends, and supply chain management, necessitating production control and decisions that integrate a vast amount of dynamic information. Against this backdrop, the introduction of large language models (LLMs) with advanced language understanding and generation capabilities for question-answering and decision support in fashion manufacturing presents significant value [2]–[4]. LLMs, leveraging natural language processing techniques, can rapidly comprehend and generate text related to production and manufacturing, assisting managers in real-time decision-making, issue diagnosis, and process optimization, as shown in Fig. 1. However, fashion manufacturing involves a substantial amount of domain-specific terminology, technical workflows, material properties, and industry standards, leading to notable limitations in the domain-specific knowledge of existing LLMs. To address this challenge, integrating an external fashion knowledge base in combination with Retrieval-Augmented Generation (RAG) [5] emerges as a promising enhancement

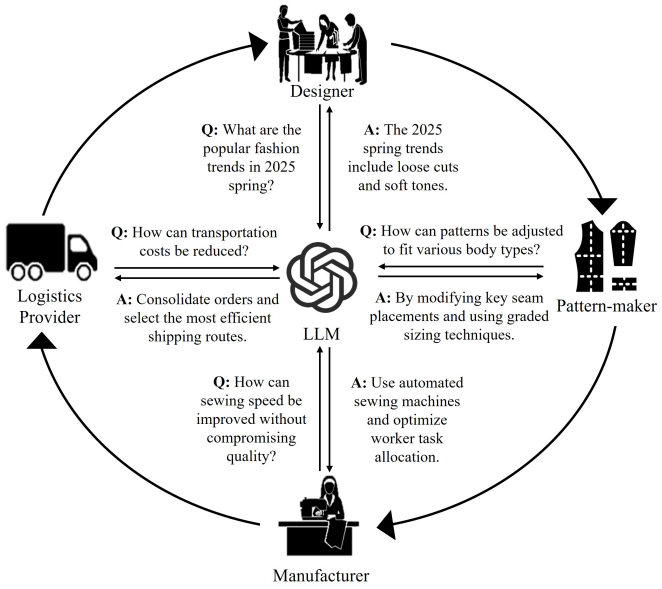


Fig. 1: Application of LLMs to the fashion manufacturing life cycle.

The source code is available at https://github.com/polyu-sun0130/SGDCR.

[1]Kexin Sun is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China kexin130.sun@connect.polyu.hk

[2]Yujie Han is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China yujiejenny.han@connect.polyu.hk

[3]Zhiheng Zhao is with the Department of Industrial and Systems Engineering and the Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University, Hong Kong, China zhiheng.zhao@polyu.edu.hk

[4]Geroge Q. Huang is with the Department of Industrial and Systems Engineering and the Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University, Hong Kong, China gq.huang@polyu.edu.hk

approach. The external knowledge base systematically consolidates expert knowledge in fashion manufacturing. Through the RAG framework, LLMs can dynamically retrieve relevant information from the external knowledge base during response generation, ensuring that their outputs are not only grounded in general language understanding but also closely aligned with specialized domain knowledge. This integration ultimately provides the industry with more efficient and reliable technical support and decision support.

To enhance retrieval efficiency within large-scale knowledge bases, existing RAG methods [6] typically rely on simple similarity matching techniques, such as cosine similarity or keyword matching, to retrieve a large number of knowledge documents related to a given query. However, the fashion manufacturing process involves a vast amount of complex domain knowledge, including specialized terminology, production workflows, and industry standards, making existing methods suffer from significant limitations. On the one hand, given the specialized and complex knowledge in fashion manufacturing, basic similarity matching techniques struggle to accurately model the semantic relevance between queries and multiple documents, often resulting in retrieved documents being disorganized and lacking logical structure. Studies [7], [8] have shown that when the most relevant documents are positioned in the most easily attended locations, typically at the forefront of the input, performance in question-answering tasks improves significantly. On the other hand, due to the continuous evolution of techniques in fashion manufacturing, knowledge bases often contain redundant records, leading to retrieved documents with excessive knowledge redundancy. This redundancy not only results in lengthy input prompts for LLMs but also disrupts contextual coherence, thereby compromising both the quality and efficiency of answer generation.

To address these challenges, document reranking and filtering to refine retrieval results emerges as a potential solution for enhancing retrieval precision and model performance. Recently, pretrained language models such as BART [9], RoBERTa [10], and BGE [11] have been widely applied to document reranking tasks, with their core approach focused on optimizing ranking performance by estimating relevance scores between queries and documents. These models are typically built on encoder stacks within the Transformer architecture, enabling them to capture deep semantic representations and model the intricate relationships between queries and documents. While these methods have demonstrated significant improvements in reranking tasks across general domains, they still present certain limitations when applied to question-answering tasks in the fashion manufacturing domain. Specifically, question-answering tasks in fashion manufacturing often exhibit a densely open-ended nature, requiring reasoning across multiple documents. However, existing methods rank documents solely based on their individual relevance to the query, disregarding contextual relationships among multiple documents. This single-dimensional ranking strategy may lead to redundant documents being placed at the top, causing LLMs to overlook important information that may exist in other positions. Moreover, the fashion manufacturing domain involves a vast number of specialized terms and intricate production workflows, making precise semantic understanding crucial. Relying solely on shallow semantic matching without considering inter-document relationships can lead to relevance misjudgments, ultimately affecting the performance of the question-answering system. Therefore, incorporating document complementarity into the retrieval ranking process is essential for providing LLMs with a refined and contextually coherent set of documents.

It should be noted that the contribution of each document to a query is not solely determined by its individual relevance, but also influenced by the content of other documents. This resembles a dynamic team collaboration mechanism, where all members (”documents”) are dynamically evaluated based on their capabilities and assigned different levels of importance to collaboratively solve the task (”query”). In this process, redundant members with overlapping functions are eliminated. Building on this insight, we propose an innovative document reranking framework based on dynamic capsule routing network with semantic graph. This framework consists of a filtering module that models document complementarity and a reranking module that measures query-answer similarity. In the filtering module, we introduce a capsule-coordinated dynamic routing mechanism embedded within a semantic association graph to dynamically capture complex semantic relationships between documents. This mechanism learns multi-document coupling coefficients with the query, serving as a measure of each document’s contribution. Documents with the lowest contribution scores are filtered out based on their rankings. The remaining filtered documents are then processed using cross-attention modeling to obtain deep semantic feature embeddings for similarity measurement. Finally, documents are reranked based on a combined consideration of relevance scores and contribution scores. The reranked documents are then concatenated to form a coherent, efficient, and precise document prompt for the LLM. Experiments conducted on both publicly available densely open-domain question-answering datasets and a constructed fashion manufacturing question-answering dataset demonstrate that the proposed method outperforms existing reranking approaches, showcasing its effectiveness and superiority in knowledge-driven question-answering systems for fashion manufacturing. The contributions of this paper are as follows:

- To address complex fashion manufacturing QA tasks, we developed a retrieval-augmented optimization framework with a dynamic cooperative routing network to enhance general-purpose LLMs’ applicability in apparel production domains.
- We embed a semantic association graph into the dynamic routing mechanism, capturing knowledge redundancy by modeling the contextual structural relationships between documents. This enables dynamic weighting of routing coupling coefficients to filter out redundant document information.
- We set the reranking criteria as a comprehensive consideration of both individual document relevance and collaborative contribution, forming an efficient, precise, and contextually coherent document prompt for the LLM.

This improves the reliability and practicality of the fashion manufacturing question-answering system.

II. Related Works

The manufacturing industry is rapidly advancing toward intelligent transformation [12]–[14], and as the most versatile technological paradigm in the field of AI, LLMs demonstrate significant enabling potential. As an emerging approach to enhancing the performance of LLMs in specialized domains, RAG is characterized by its ease of deployment and strong performance, demonstrating significant application value and development potential. Consequently, it has garnered increasing research interest from scholars. In this section, we primarily review the applications of RAG in the manufacturing industry and the corresponding retrieval reranking methods. Finally, we summarize the limitations of existing research and present the motivation for our study.

A. RAG for Manufacturing

By combining the generative capabilities of LLMs with structured and unstructured domain-specific retrieval processes, RAG-based systems have emerged as a promising approach to critical challenges related to dynamic problem-solving, contextual reasoning, and automation in manufacturing environments. Existing research demonstrates that RAG systems improve several aspects of manufacturing, particularly in supply chain management [15], [16], real-time troubleshooting [17], and structured decision support [18], [19]. Specifically, studies such as [20] and [21] highlighted how RAG enhances supply chain visibility and operational efficiency by integrating external datasets and knowledge graphs (KGs). Research such as [17] and [22] examined RAG for intelligent question-answering in industrial documentation workflows, facilitating fast and context-aware retrieval of technical information. The integration of these systems with digital twins [23] and multi-agent frameworks [18], [19] allowed adaptive production planning, data-driven decision making, and autonomous task execution. Furthermore, [24] explored how RAG copilots help in complex equipment selection, demonstrating practical benefits in initial planning and operational efficiency. Another significant area of application is worker assistance and knowledge sharing, where LLMs act as cognitive agents to enhance factory operations and learning environments [25], [26]. These systems enable conversational AI interactions for operators, allowing them to quickly retrieve maintenance instructions, resolve complex troubleshooting queries, and reduce reliance on fixed documentation sources.

While these studies confirm the usefulness of RAG in industrial contexts, they also identify several limitations. Effective retrieval is highly dependent on high-quality structured knowledge bases, which may be lacking in many industrial settings [27], [28]. Furthermore, the extensive retrieval of highly specialized long-form content poses significant challenges to the information processing capabilities of LLMs. To enhance the application effectiveness of LLMs in the fashion manufacturing sector, in-depth research on retrieval optimization techniques is imperative to improve the model's robustness and output reliability.

B. Retrieval Reranking

A crucial component of RAG optimization is reranking, which refines the set of retrieved documents before they are passed to the generator. Reranking methods [29]–[35] aimed to optimize document ordering to maximize downstream generation quality, addressing challenges such as noisy retrieval, relevance misalignment, and query ambiguity. Traditional reranking methods focused on semantic matching, often leveraging cross-encoders or contrastive learning models to improve ranking precision [7], [36]. However, in RAG systems, maximizing retrieval relevance alone does not always correlate with optimal generation outcomes, necessitating utility-driven reranking approaches that incorporate task-specific considerations such as factuality, informativeness, and coherence [31], [37], [38]. Several studies proposed hybrid models that integrate retrieval, reranking, and generation into end-to-end frameworks. For instance, Re2G [7] integrated a reranking mechanism into a retrieval and generation pipeline via knowledge distillation, optimizing all components jointly. The effectiveness of these end-to-end models demonstrates the importance of aligning reranking with generative objectives rather than treating retrieval relevance in isolation. The above independent similarity metric methods are prone to correlation misjudgment due to redundant noise, recent approaches have enhanced ranking accuracy through the integration of document contextual relationship modeling. Graph-based reranking, exemplified by G-RAG [39], employed graph neural networks to model inter-document relationships, leveraging Abstract Meaning Representation (AMR)-based graphs to improve context selection. Similarly, ListConRanker [36] enhanced listwise encoding through contrastive training objectives, allowing for holistic ranking of multiple retrieved documents in a single pass.

Despite significant progress in reranking for RAG, key challenges remain. Evaluation misalignment between retrieval relevance metrics (e.g., Recall@K, NDCG) and generation quality metrics (e.g., EM, F1) complicates objective ranking evaluation [37], [40]. Moreover, existing graph-based reranking methods fail to account for the dynamic nature of inter-document relationships that vary with query content, resulting in limited accuracy and adaptability of the ranking results.

C. Research Gap and Motivation

The current RAG frameworks are evolving toward greater precision and efficiency. However, due to the reliance on coarse-grained matching methods in the initial retrieval stage, the retrieved results often contain a large number of redundant documents and suffer from relevance misjudgments. Although existing studies have attempted to improve retrieval quality through reranking techniques based on deep semantic feature matching—placing the most relevant retrieval blocks at the peripheral positions where LLMs may focus—these methods exhibit significant limitations when applied to the fashion manufacturing domain. Documents in this domain typically possess complex logical structures and highly specialized content. Traditional ranking methods fail to account for the synergistic relationships between documents and their overall contribution to a given query. This one-dimensional ranking

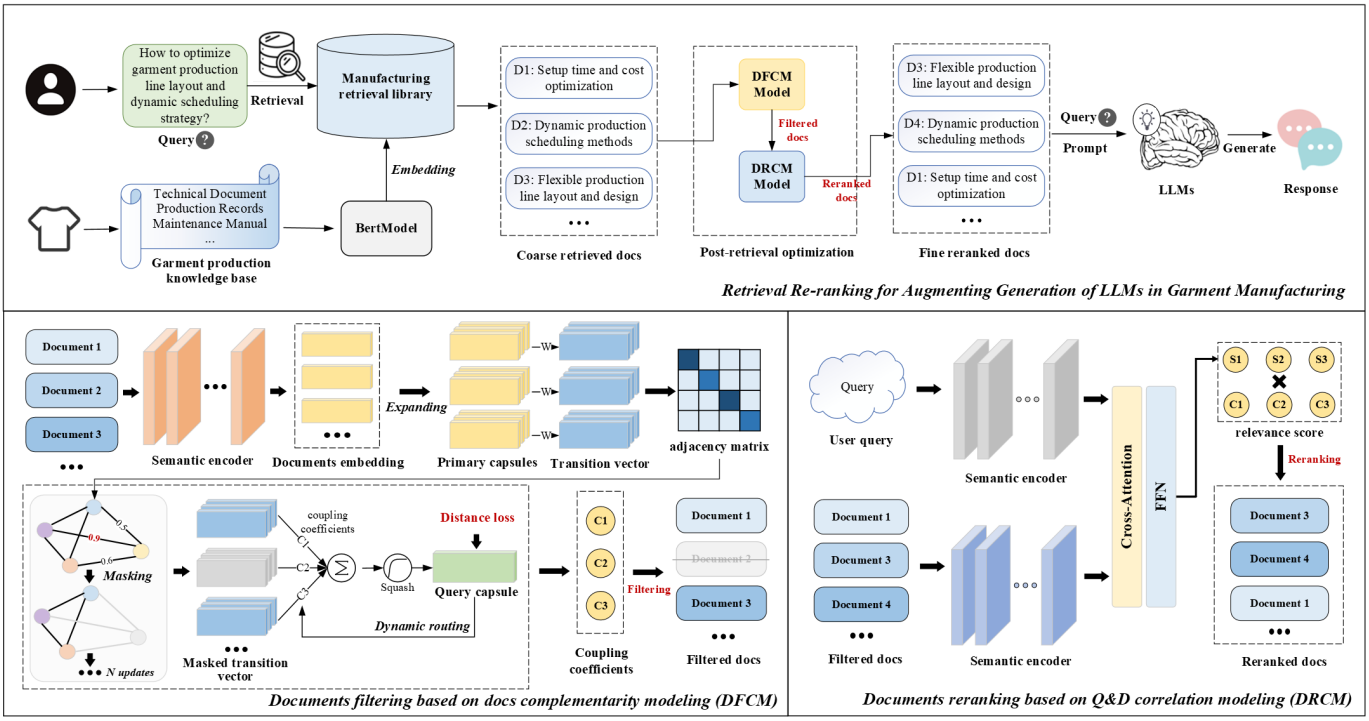


Fig. 2: The overall framework of SGDCR. The input is the user's query during the fashion manufacturing process. After retrieving from the database, a set of coarse-grained retrieval documents is obtained. This set is then filtered through the DFCM module and reranked by the DRCM module, resulting in a fine-grained retrieval document set. Finally, this set is concatenated with the query and input into the LLM to generate the output response.

strategy may lead to several issues. First, certain documents may exhibit high local semantic similarity to the query but contribute minimally in practice, potentially introducing redundant information. Second, the contextual dependencies among documents remain underutilized, resulting in a lack of global consistency in the ranking, which negatively impacts the accuracy and reliability of LLM-generated responses.

To address these limitations, modeling the complementarity among documents emerges as an effective solution. This requires that document blocks be aware of each other, evaluate semantic overlaps or differences, and form a unified understanding that aligns with the query. This process is similar to a collaborative team in which each member provides partial knowledge to improve the overall answer quality. Traditional attention mechanisms mainly focus on the relevance between each document and the query individually. They struggle to model the cooperation among documents effectively, which may lead to redundant information being overused or important complementary content being ignored. In contrast, the dynamic routing mechanism in capsule networks simulates an aggregation process from parts to a whole. It is well suited to capture semantic complementarity among multiple documents. Each document block is represented as a vector capsule, which encodes both the existence and the semantic orientation of features. This allows the model to understand not only how relevant a document is but also the direction and nature of its contribution within a broader context. By modeling the inverse mapping from document capsules to the query capsule, which means predicting the most semantically aligned query representation based on document interactions, the model iteratively updates the routing coefficients through mutual agreement. This enables the model to infer the level of contribution each document makes toward answering the query. Therefore, the dynamic routing mechanism based on capsules offers structural support and expressive capability for capturing document collaboration, making it a well-suited choice for our task.

III. Methodology

Fig. 2 illustrates the complete process of optimizing LLM-based QA tasks in fashion manufacturing through retrieval reranking. First, the system collects data from textual knowledge related to terms, technical manuals, workflows, equipment maintenance records, etc., involved in the fashion production process, and constructs a knowledge retrieval corpus for the fashion manufacturing domain using vector embedding techniques. Next, for the queries input during production, the system performs initial matching in the retrieval corpus to obtain k relevant documents $D = \{d_1, d_2, ..., d_k\}$. These coarse-grained retrieval results D are then input into the filtering module based on document complementarity modeling (DFCM) and the reranking module based on QA similarity measurement (DRCM) for redundancy filtering and reranking, generating fine-grained retrieval results. Finally, the system concatenates the documents with the query in the re-ranked order and inputs them into the LLM to generate accurate and contextually coherent answers. The specific methods will be detailed in this section.

A. Fashion Manufacturing Document Retrieval

Various data sources involved in the fashion production process, such as glossaries, technical manuals, workflow documents, equipment maintenance records, etc., contain both structured and unstructured text, covering all stages from raw material selection to final product packaging. To ensure data quality, the raw text is first preprocessed, including tokenization, removal of stopwords, standardization of terminology expressions, and handling of special symbols. After preprocessing, the text is transformed into high-dimensional vector representations using a pre-trained Sentence-BERT [41] language model, capturing the semantic information of the text and mapping it into a continuous vector space:

$$v_i = \text{SentenceBERT}(d_i) \tag{1}$$

where n is the dimension of the embedding vector. The vector representations of all documents are stored in an efficient vector database, forming the knowledge retrieval corpus for the fashion manufacturing domain. When a user inputs a query q, the same preprocessing and vector embedding operations are performed on the query to generate its vector representation $\mathbf{v}_q$. Then, to improve retrieval efficiency in the large fashion manufacturing corpus, an approximate nearest neighbor search algorithm is used for fast matching, finding the k most similar document vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$ to the query vector $\mathbf{v}q$. The corresponding original document set $D = d_1, d_2, ..., d_k$ is the initial retrieval result. Since current retrieval techniques primarily rely on shallow semantic matching and lack a deep understanding of context and domain knowledge, the preliminary retrieval result Dcoarse may contain redundant documents or documents with low relevance, necessitating further filtering and reranking optimization.

B. Documents Filtering Based on Complementarity Modeling

Modeling the relationship between documents to identify redundant information is crucial for effective filtering. Although some existing studies have recognized the importance of modeling document relationships during retrieval reranking, they focus solely on the relationships within multiple documents and neglect the importance of the query as the target. In a sense, the interrelationships between documents are not fixed but depend on the specific query content, and this dynamic nature is an important factor to consider in retrieval ranking tasks. This is analogous to the division of labor within a team, which is not static but requires dynamic task allocation based on different objectives. Therefore, it is essential to model the complementarity of retrieval documents in relation to the query. Theoretically, when multiple retrieval documents are obtained based on a specific query, the potential query can be predicted in reverse from these documents. Building upon this insight, we aim to establish an inverse mapping process from multiple documents to the query, learning the contribution of different documents to the prediction of the query and modeling their importance in handling the query. Inspired by capsule networks used for computer vision tasks, we propose a dynamic routing network embedded with semantic graph capsules to model the inverse mapping from the retrieved document set D_{coarse} to the corresponding query q. The document information is dynamically adjusted during the iterative process based on the contribution of other documents, thereby capturing the collaborative relationships between documents. Specifically, D_{coarse} is first passed through a semantic encoder (using the pre-trained BART encoder [42]) to obtain deep semantic representations:

$$V_d = \text{BART-Encoder}(D_{\text{coarse}}) \tag{2}$$

where $V_d = \{v_{d_1}, v_{d_2}, \ldots, v_{d_k}\}$ represents the deep semantic representation matrix of the document set, and $v_{d_k} \in \mathbb{R}^d$ is the semantic vector of the i-th document, with d being the vector dimension. Subsequently, we expand the dimensionality of the deep semantic representation vectors V_d to form the initial document capsule set $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$, where $\mathbf{u}_i \in \mathbb{R}^{d \times m}$ and m is the capsule dimension. The prediction vectors are obtained through an affine transformation:

$$\hat{\mathbf{v}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i + \mathbf{b}_{ij} \tag{3}$$

where $\hat{\mathbf{v}}_{j|i}$ represents the prediction vector of the i-th document capsule for the j-th high-level capsule, $\mathbf{W}_{ij} \in \mathbb{R}^{m \times m}$ is a learnable weight matrix, and $\mathbf{b}_{ij} \in \mathbb{R}^m$ is the bias term. The prediction vector $\hat{\mathbf{v}}_{j|i}$ is used to obtain the high-level capsule representation $\mathbf{s}_j$ through multiple dynamic routing iterations, calculated as follows:

$$\mathbf{s}_j = squash(\sum_{i=1}^{k} c_{ij}\hat{\mathbf{v}}_{j|i}) \tag{4}$$

where $squash()$ is an activation function, c_{ij} is the coupling coefficient, representing the contribution of the i-th document capsule to the j-th high-level capsule. It is iteratively updated through the dynamic routing mechanism:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_{l=1}^{k} \exp(b_{il})} \tag{5}$$

where b_{ij} is the log prior probability, initialized to 0 and updated in each iteration based on the agreement between the high-level capsule and the prediction vector:

$$b_{ij} \leftarrow b_{ij} + \hat{\mathbf{v}}_{j|i} \cdot \mathbf{s}_j \tag{6}$$

Through this dynamic routing mechanism, the contribution of document capsules can be dynamically adjusted based on their agreement with high-level capsules, enabling precise reranking of the document set. The routing coupling coefficients are obtained through N dynamic iterations, representing the varying contributions of document capsules to the prediction of high-level capsules. Subsequently, we minimize the cosine similarity distance between the query q and the high-level capsule representation $\mathbf{s}_j$, achieving a precise inverse mapping from the document set to the query. The query capsule mapping loss is defined as follows:

$$L_{\cos} = 1 - \frac{V_q \cdot \mathbf{s}_j}{|V_q| \cdot |\mathbf{s}_j|} \tag{7}$$

where, V_q = BART-Encoder(q). It should be noted that, while document information can be dynamically adjusted based on the contributions of other documents during the iterative process, identifying redundant information remains challenging. To address this, we introduce a mask mechanism based on a semantic correlation graph in the routing iteration process. This mechanism enables each document to enhance its representation by aggregating contextual information from its semantic neighbors. A mask score is computed for each document based on its similarity to other documents. If a document exhibits high similarity with at least one other document, it is considered potentially redundant, and its contribution is down-weighted by applying a soft mask. Accordingly, the dynamic routing coefficients for each document are adjusted using these mask scores, thereby reducing the influence of redundant documents during the iterative routing process. Specifically, the initial document vector $V_d^{(l)}$ is firstly embedded into a Graph Convolutional Network (GCN) node and updated by aggregating information from neighboring nodes:

$$V_d^{(l+1)} = GCN(V_d^{(l)}, A) \tag{8}$$

where A is the adaptively learned adjacency matrix: $A = \text{Softmax}((V_d^{(l)} \cdot (V_d^{(l)})^T)/t)$, t is the temperature coefficient. Next, we traverse the document set and compute the similarity between each document and others. Taking document d_i as an example, the mask score F_{mask_i} is computed based on the updated document representation:

$$F_{mask\_i} = \begin{cases} 1 - \max\left(S_{ij}\right), & \text{if } \max\left(S_{ij}\right) > \lambda_m, \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

where $S_{ij} = \text{sim}(V_{di}^{(l+1)}, V_{dj}^{(l+1)})$, sim() is defined as cosine similarity. λ_m is the mask threshold. Through the aforementioned masking mechanism, documents that are initially highly similar to others in the set will be filtered out, while those with lower similarity will retain their information. After traversing all documents and computing the mask scores F_{mask_i}, we apply them to the dynamic routing coefficients to further filter redundant information in the process of modeling document contribution:

$$\begin{aligned} b_i^{(1)} &= b_i^{(0)} \cdot F_{mask\_i}, \\ C_i^{(1)} &= \text{softmax}(b_i^{(1)}) \end{aligned} \tag{10}$$

Thus, the final contribution score is obtained as $F_{c_i} = C_i^{(1)}$. By setting a threshold λ_f, documents with low contribution scores are filtered out, resulting in the final document set D_f. The general algorithmic flow of the DFCM module is shown in Algorithm 1.

Algorithm 1 Algorithmic flow of the DFCM Module

Input: Document set D_{coarse}, Query q
Step 1: Documents and query embedding
$V_d \leftarrow$ BART-Encoder(D_{coarse})
$V_q \leftarrow$ BART-Encoder(q)
Step 2: Initialize document capsules
$\mathbf{U} \leftarrow expand(V_d)$
Step 3: Compute mask scores using GCN and semantic similarity
$A \leftarrow \text{Softmax}\left((V_d^{(l)} \cdot (V_d^{(l)})^T)/t\right)$
$V_d^{(l+1)} \leftarrow \text{GCN}(V_d^{(l)}, A)$
for each document d_i in D_{coarse} **do**
  $S_{ij} \leftarrow \text{sim}(V_{di}^{(l+1)}, V_{dj}^{(l+1)})$
  if $\max(S_{ij}) > \lambda_m$ **then**
    $F_{mask\_i} \leftarrow 1 - \max(S_{ij})$
  else
    $F_{mask\_i} \leftarrow 1$
  end if
end for
Step 4: Dynamic routing with semantic graph capsules
for iteration $= 1$ to N **do**
  for each document capsule $\mathbf{u}_i$ in $\mathbf{U}$ **do**
    for each high-level capsule j **do**
      $\hat{\mathbf{v}}_{j|i} \leftarrow \mathbf{W}_{ij}\mathbf{u}_i + \mathbf{b}_{ij}$
    end for
  end for
  for each high-level capsule j **do**
    $\mathbf{s}_j \leftarrow \text{squash}\left(\sum_{i=1}^{k} c_{ij}\hat{\mathbf{v}}_{j|i}\right)$
  end for
  for each $\mathbf{u}_i$ and j **do**
    $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{v}}_{j|i} \cdot \mathbf{s}_j$
    $b_{ij}^{(1)} \leftarrow b_{ij}^{(0)} \cdot F_{mask\_i}$
    $c_{ij}^{(1)} \leftarrow \frac{\exp(b_{ij}^{(1)})}{\sum_{l=1}^{k}\exp(b_{il}^{(1)})}$
  end for
end for
Step 5: Filter documents based on contribution scores
$D_f \leftarrow \{d_i \mid F_{c_i} = C_i^{(1)} > \lambda_f\}$
return D_f, $C_i^{(1)}$

C. Documents Reranking Based on Similarity Modeling

In the document reranking process, we comprehensively consider both the contribution degree and similarity of documents to the query. The contribution score is obtained by modeling the collaborative relationships between documents through a dynamic routing mechanism, which reflects the importance of documents in the overall context. Meanwhile, the similarity is measured through an independent interaction process to ensure local semantic alignment between documents and the query. This dual consideration not only effectively filters redundant documents but also enhances the accuracy and robustness of the ranking results. Therefore, the DRCM module aims to model the similarity between documents and the query, further optimizing the document ranking. Specifically, for the input filtered document set D_f, the documents are first passed through a semantic encoder (using the pre-trained BART encoder [42]) to obtain deep semantic representations:

$$V_{D_f} = \text{BART-Encoder}(D_f) \tag{11}$$

Subsequently, we compute the relevance between each document and the query v_q using a cross-attention mechanism.

The attention scores are obtained by computing the attention weights between the query and document representations, followed by a softmax normalization:

$$A = \text{softmax}\left(\frac{v_q W_q (V_{D_f} W_k)^T}{\sqrt{d}}\right) \tag{12}$$

where W_q and W_k are learnable projection matrices, and d is the dimension of the representations. The attended document representation is first obtained by applying the attention scores to the document embeddings:

$$V'_{D_f} = AV_{D_f} \tag{13}$$

Then, a feedforward neural network (FFN) is applied to transform V'_{D_f} into a scalar relevance score:

$$F_{r\_i} = W_r V'_{D_f} + b_r \tag{14}$$

where W_r and b_r are learnable parameters. Inspired by multi-tasking ranking recommender systems [43], [44], the contribution score $F_{c\_i} = C_i^{(1)}$ and the relevance score F_{r_i} are combined using a product-based approach to compute the final ranking score F_i:

$$F_i = F_{c\_i} \cdot F_{r\_i} \tag{15}$$

Finally, the filtered document set D_f is re-ranked according to the final ranking scores, resulting in a fine-grained retrieved document set D_{fine}. The query q and the re-ranked documents are concatenated and input into the LLM to generate the final answer.

D. Optimization objectives

Existing research predominantly emphasizes document reranking, with performance metrics (such as Mean Reciprocal Rank, MRR, and Hits@10) primarily focused on ranking tasks. In this paper, we place greater emphasis on the performance enhancement effect of the reranking process on the downstream task of LLM-based question answering in the fashion manufacturing domain. To achieve this, we optimize the parameters of the upstream reranking network by leveraging the performance of the downstream generation task. During training, the parameters of the LLM are frozen, and the cross-entropy loss is computed by comparing the generated answers with the ground truth labels, as follows:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{16}$$

where, y_i represents the ground truth label for the i-th token in the answer, $\hat{y}_i$ denotes the predicted probability distribution for the i-th token generated by the LLM, N is the total number of tokens in the answer.

By minimizing the cross-entropy loss $\mathcal{L}_{\text{CE}}$, the reranking network parameters are optimized to improve the quality of the retrieved documents, thereby enhancing the overall performance of the LLM in generating accurate and contextually relevant answers. Finally, Combine the answer cross-entropy loss and the question capsule mapping loss L_{COS} to obtain the final loss function:

$$L_{\text{total}} = L_{CE} + \lambda L_{COS} \tag{17}$$

where λ is the balancing coefficient. The optimization process of the algorithm is shown in Algorithm 2.

Algorithm 2 SGDCR optimization process

Input: Coarse-grained document set D_{coarse}, Query q
Initialize: The parameters Θ_{DFCM} of DFCM module, the parameters Θ_{DRCM} of DRCM module.
Step 1: Documents filtering
$F_{c\_i} \leftarrow \text{DFCM}(D_{\text{coarse}}, q)$
$D_{\text{f}} \leftarrow \{d_i \mid F_{c\_i} > \lambda_f\}$
$F_{r\_i} \leftarrow \text{DRCM}(D_{\text{f}}, q)$
Step 2: Documents reranking
$F_i \leftarrow F_{c\_i} \cdot F_{r\_i}$
$D_{\text{fine}} \leftarrow \text{Sort}(D_{\text{f}}, F_i)$
Step 3: Parameters Optimization
Compute L_{total}: $L_{\text{total}} \leftarrow L_{CE} + L_{COS}$
Update Θ_{DFCM} and Θ_{DRCM} using gradient descent on L_{total}
return D_{fine}

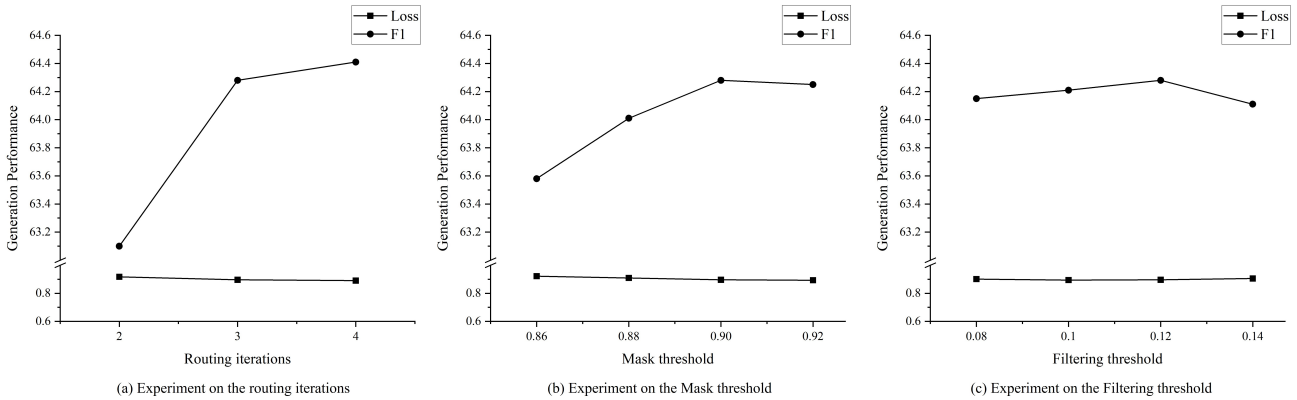
IV. Experimental Results

In this section, considering both the sample size and fairness of the evaluation, we first conducted comparative and ablation experiments on publicly available datasets containing dense open-ended question-answering tasks to validate the effectiveness and superiority of the SGDCR framework in similar complex QA scenarios. Subsequently, we constructed a question-answering dataset specific to the garment manufacturing and performed a case study to assess the potential applicability of the proposed SGDCR framework in this domain.

A. Experimental Setup

Dataset. We conducted comparison and ablation experiments on the multi-hop query datasets SQuAD [45] and HotpotQA [46], Both of them consist of structured, densely packed open-ended question-answer pairs, where questions exhibit high complexity and typically require reasoning across multiple documents to derive the correct answer. This multi-hop reasoning characteristic aligns with the nature of question-answering tasks in the fashion manufacturing domain, particularly when addressing complex production processes, supply chain management, or quality control issues, which often necessitate comprehensive analysis and inference across multiple information sources. In our case study, we conducted comparative experiments using a self-constructed QA dataset on garment manufacturing, benchmarking against the state-of-the-art baseline models trained on public datasets.

Base LLMs. To evaluate the performance of re-ranked documents across language models of different scales, we conducted experiments on multiple mainstream LLMs, including: 1) FLAN-T5-Large (0.78B): A compact model offering efficient inference and reasonable performance, suitable for



(a) Experiment on the routing iterations (b) Experiment on the Mask threshold (c) Experiment on the Filtering threshold

Fig. 3: Hyperparameter experiments on SGDCR.

lightweight RAG evaluations. 2) FLAN-T5-XL (3B): A moderately sized model that balances performance and efficiency, making it suitable for evaluating RAG effectiveness under constrained computational resources. 3) Llama-3.1-Instruct (8B): As a mid-scale instruction-tuned model, Llama-3.1-Instruct excels in multi-task learning and complex instruction comprehension, making it well-suited for evaluating the performance of mid-scale models in multi-hop reasoning tasks. 4) FLAN-T5-XXL (11B): As a larger-scale model, FLAN-T5-XXL demonstrates superior generation quality and contextual understanding, making it ideal for assessing the potential of RAG in larger models.

Retrieval reranking comparison methods. To comprehensively assess the effectiveness of retrieval reranking, we compared multiple approaches, including a no-reranking baseline, reproduced classical methods, and publicly available state-of-the-art reranking models. The specific methods are as follows:

- **No Reranking**: As a baseline method, this approach directly utilizes the retrieved document order without any reranking, serving as a reference to quantify the performance improvement introduced by reranking techniques.
- **BART+Transformer**: A reproduced classical method that encodes retrieved documents using the BART model and applies a Transformer-based architecture for reranking. This approach has demonstrated strong performance in early multi-hop QA tasks.
- **BART+GCN**: A reproduced reranking method based on graph convolutional networks, which encodes documents using the BART model and models inter-document relationships via GCN for reranking. This method is particularly effective in capturing complex dependencies between documents.
- **BART-reranker [42]**: A reranking approach implemented based on publicly available code, leveraging the BART model to directly rerank retrieved documents with strong contextual understanding capabilities.
- **gte-reranker [8]**: A reranker based on publicly available General Text Embedding (GTE), which employs high-quality text embeddings to rerank retrieved documents, making it suitable for open-domain QA tasks.
- **bge-reranker-v2-gemma [47]**: A reranker based on publicly available Bidirectional Generative Embedding (BGE), enhanced with the Gemma model to improve multi-hop reasoning and complex QA performance.
- **jina-reranker-v2 [48]**: A reranker based on publicly available Jina reranking models, leveraging efficient embeddings and reranking techniques tailored for large-scale retrieval scenarios requiring real-time reranking.

Evaluation metrics. To comprehensively assess the enhancement effect of different reranking methods on LLMs, we adopt the following evaluation metrics for comparative analysis: **Exact Match (EM)**, which measures the proportion of generated answers that exactly match the ground truth, providing a strict accuracy assessment; **F1 Score**, which evaluates the overall quality of generated answers by considering both precision and recall, ensuring a balanced assessment of generation performance; and **Time**, representing the reranking time, which measures the duration from initial retrieval to the completion of reranking, excluding the answer generation time by LLMs. Specifically, for the garment manufacturing QA dataset in our case study, we adopted **ROUGE-L** and **BLEU** metrics because the long-text nature of the dataset necessitates evaluating both semantic coherence and fine-grained lexical matching.

Implementation details. The retrieval corpus for the SQuAD and HotpotQA datasets consists of all paragraph texts from the training, validation, and test sets. We employ the Fasis retriever to retrieve 10 relevant documents for each query, because the ground-truth relevant documents per query are typically fewer than 10. To eliminate the influence of retrieval variability on experimental results, all comparative experiments are conducted based on the same set of retrieved documents, ensuring a fair reranking evaluation. The training corpus of our SGDCR is the Wiki dataset. During the training process, the number of training epochs is set to 10, and the optimizer used is AdamW, with a learning rate of 5e-5, ensuring convergence within a reasonable time while achieving optimal performance. Both training and testing experiments are conducted on two Nvidia A6000 GPUs, providing the necessary computational resources to support the efficient training and inference of large-scale models.

Methods	HotpotQA									SQuAD								
	FLAN-T5-Large		FLAN-T5-XL		Llama-3-Instruct		FLAN-T5-XXL		Time	FLAN-T5-Large		FLAN-T5-XL		Llama-3-Instruct		FLAN-T5-XXL		Time
	EM	F1	EM	F1	EM	F1	EM	F1		EM	F1	EM	F1	EM	F1	EM	F1	
No Reranking	39.40	46.09	35.80	56.38	22.80	35.78	37.00	44.72	0.00	29.00	44.96	31.20	49.99	17.00	32.44	27.40	36.84	0.00
BART+Transformer	40.80	47.95	41.60	60.19	23.00	36.30	42.20	49.19	0.046	34.20	48.36	33.80	52.86	18.20	34.05	32.20	41.45	0.045
BART+GCN	41.20	48.13	37.00	57.25	26.20	39.19	43.40	50.43	0.051	34.80	48.87	33.20	52.09	18.80	34.43	30.80	39.97	0.051
BART-reranker	41.60	48.92	41.80	60.25	26.80	40.62	45.80	53.41	0.109	36.40	52.77	38.80	56.69	21.60	36.45	35.40	45.04	0.106
gte-reranker	42.00	49.40	41.20	59.39	26.60	39.61	44.00	50.73	0.049	36.40	54.29	38.60	56.56	22.40	36.94	33.00	43.55	0.051
bge-reranker-v2-gemma	41.80	48.63	41.00	59.44	25.40	39.40	44.80	51.71	0.280	36.00	53.33	38.80	56.72	21.80	36.79	33.60	44.71	0.276
jina-reranker-v2	41.60	48.65	41.40	60.36	26.00	40.29	43.60	50.88	0.044	36.80	52.58	37.20	55.15	21.00	35.27	30.60	40.79	0.044
Ours	43.80	50.21	46.40	62.77	27.60	41.63	46.60	53.12	0.048	38.60	54.17	41.00	59.83	24.20	38.97	38.40	48.22	0.049

TABLE I: Comparative experiments on retrieval reranking enhanced base LLMs on HotpotQA and SQuAD datasets.

Ablation Version	FLAN-T5-XL (3B)		Llama-3-Instruct (8B)		Time
	EM	F1	EM	F1	
Full	46.40	62.77	27.60	41.63	0.048
W/O DFCM	41.20	58.48	22.80	35.54	0.032
W/O Semantic Graph	43.20	60.33	25.60	39.13	0.042
W/O Filtering	43.80	60.50	25.20	37.76	0.048
W/O DRCM	44.60	61.42	26.40	39.20	0.041

TABLE II: Ablation experiments on retrieval reranking enhanced base LLMs on HotpotQA dataset.

B. Hyperparameter Experiments

In the training setup of the SGDCR model, the settings of the dynamic routing iteration count N, the masking threshold λ_m in the semantic association graph, and the document filtering contribution threshold λ_f impact the model's reranking performance. To determine the optimal values for these hyperparameters, we conducted detailed hyperparameter experiments, as shown in Fig. 3. We can obtain the following observations:

(1) The dynamic routing iteration count N determines the depth of information propagation in the dynamic routing process. As N increases, the model's loss gradually decreases, and the F1 score improves. When $N = 3$, the model achieves a good balance between loss and F1 score. Further increasing the iteration count ($N = 4$) results in only marginal performance improvement while increasing computational overhead. Therefore, we set $N = 3$ as the final configuration.

(2) λ_m controls the sparsity of edges in the semantic association graph. Experimental results show that as the masking threshold increases, the model's loss decreases, and the F1 score improves. The model achieves optimal performance when $\lambda_m = 0.90$. Further increasing the threshold leads to a slight decline in the F1 score, indicating that an excessively high threshold may filter out some useful semantic relationships. Therefore, we choose 0.90 as the optimal value for λ_m.

(3) λ_f controls the strictness of document filtering, affecting the quality of the final reranking results. As shown in the table, when $\lambda_f = 0.12$, the F1 score of LLM responses reaches its highest value while the loss remains low. When the threshold is too low or too high, the model's performance declines, indicating that a moderate filtering threshold effectively balances document retention and filtering. Therefore, we select 0.12 as the optimal value for λ_f.

C. Comparison Experiments

We used the Fasis retriever to retrieve 10 relevant documents for each query in the test set and performed an initial deduplication process. The Fasis retriever leverages efficient vectorized retrieval techniques to quickly locate candidate documents that are semantically most relevant to the query from a large-scale document corpus while preventing duplicate documents from entering the subsequent reranking stage through deduplication. All baseline reranking methods are tested using the same set of queries and retrieved documents to ensure fairness and comparability in the experiments. Table I presents the performance comparison of the SGDCR model with baseline reranking methods in enhancing FLAN-T5-Large model, FLAN-T5-XL, Llama-3-Instruct, and FLAN-T5-XXL on the HotpotQA and SQuAD datasets. The results show that SGDCR significantly outperforms other methods in terms of EM and F1 score. For example, on the multi-hop HotpotQA dataset, using the FLAN-T5-XL model, our method improves EM and F1 scores by 11.0% and 4.2%, respectively, compared to the second-best BART-reranker. For the lightweight FLAN-T5-Large model, our method achieves a notable improvement over all baselines. This suggests that our method remains highly effective even when deployed with small-scale models, offering a good balance between performance and efficiency. For the Llama-3-Instruct model, our method achieves EM = 27.60 and F1 = 41.63, surpassing BART-reranker by 3.0% and 2.5%, respectively. On the FLAN-T5-XXL model, our approach also achieves the best performance. Moreover, our method exhibits similar time efficiency to most baseline methods. On the SQuAD dataset, our method also demonstrates a significant performance advantage. In summary, our approach not only significantly improves model performance in multi-hop QA tasks but also maintains high time efficiency, proving its potential for real-world applications.

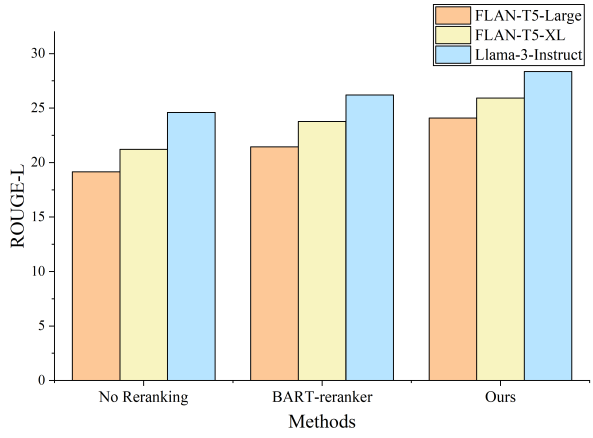
D. Ablation Study

In this section, we conduct comprehensive ablation experiments to verify the effectiveness of each module, along with complexity analysis experiments to provide practical deployment guidelines for different application scenarios and task requirements.

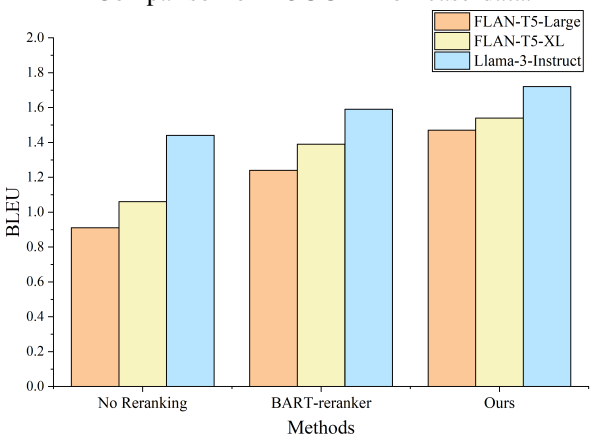
1) Ablation Analysis on Generation Performance: To verify the contribution of each core module in the SGDCR model to reranking performance, we conduct an ablation study by removing or modifying key components of the model to evaluate their impact on overall performance. The specific ablation ver-

Ablation Modules	Counts	FLOPs (M)	Time (s)	F1 Score
Routing iteration	N=2	64.12	0.021	63.10
	N=3	78.35	0.030	64.28
	N=4	92.60	0.042	64.41
	N=5	107.02	0.056	64.37
Semantic graph layers	L=1	61.95	0.026	64.10
	L=2	78.35	0.030	64.28
	L=3	86.17	0.033	64.33

TABLE III: Ablation Study on Computational Complexity of Routing and Semantic graph layers. For routing iteration ablation, the number of semantic graph layers is fixed at L=2; for semantic graph layers ablation, the number of routing iterations is fixed at N=3.



(a) Comparison of ROUGE-L on case data.



(b) Comparison of BLEU on case data.

Fig. 4: Enhanced Performance Comparison of Retrieval Reranking Methods on Case Data.

sions include: 1) Full: The complete SGDCR model, incorporating the dynamic routing mechanism, semantic graph masking mechanism, document filtering module, and similarity modeling. 2) W/O DFCM: Removes the complete Document Filtering and Contribution Modeling (DFCM) module. The reranking score is computed solely based on document similarity, ignoring the dynamic contribution relationships between documents. 3) W/O Semantic Graph: Removes the semantic graph masking mechanism from the dynamic routing process. 4) W/O Filtering: Retains the DFCM module for contribution modeling but does not filter documents, meaning all retrieved documents participate in reranking. 5) W/O DRCM: Excludes document similarity from the reranking score computation, using only the contribution modeling results for reranking.

Table II presents the performance comparison of different ablation versions on the HotpotQA dataset when enhancing FLAN-T5-XL and Llama-3-Instruct. We analyze the results from three dimensions: EM, F1 score, and reranking time to assess the contribution of each module to the model's performance. The results show that the Full SGDCR model achieves the best performance on both FLAN-T5-XL and Llama-3-Instruct. Removing the DFCM module significantly degrades performance for both models, indicating that the DFCM module plays a critical role in modeling the dynamic contribution relationships between documents. Without it, the model fails to effectively capture complex dependencies among documents. Additionally, removing the semantic graph masking mechanism results in performance degradation, demonstrating its effectiveness in identifying and filtering semantically redundant or repetitive relationships. Table II also highlights that document filtering improves the accuracy and efficiency of reranking results. Removing the DRCM module leads to a certain degree of performance decline, validating the effectiveness of ranking strategies that jointly consider contribution and similarity. From the reranking time perspective, the complete model has a slightly higher reranking time than the other versions but remains within an acceptable range, with the DFCM module contributing to the main computational overhead.

2) Ablation Analysis on Computational Complexity: In real-world applications, the computational cost of iterative dynamic routing and graph convolution may pose limitations on model deployment, especially in resource-constrained environments. To analyze this, we conducted dedicated complexity experiments by varying the number of routing iterations and semantic graph layers, as shown in Table III. Results indicate that increasing the routing iterations from 2 to 5 leads to higher FLOPs and runtime, while F1 improves only marginally. Similarly, increasing GCN layers from 1 to 3 results in moderate performance gains but increases latency and computational overhead. These findings suggest that using 2–3 routing iterations and 1–2 GCN layers provides a good trade-off between efficiency and effectiveness. For deployment in large-scale or real-time industrial systems, especially those running on edge devices or limited computing infrastructure, we recommend adopting lightweight settings (e.g., 2 routing iterations, 1 GCN layer), which can significantly reduce latency and resource consumption while maintaining competitive accuracy.

E. Case Study

We conducted a case study in the garment manufacturing domain to further validate the effectiveness of SGDCR. We collected 5,000 document knowledge entries from technical manuals, maintenance records, and other relevant sources to construct a retrieval library. Additionally, we carefully designed 500 question-answer pairs as case study data for

NO.	Question Description	Answer
1	How to avoid thread breakage during machine sewing?	Ways to avoid thread breakage during machine sewing include choosing the right needle and thread, adjusting the right stitch length and thread tension, and ensuring that the fabric is flat and wrinkle-free.
2	How to ensure the straightness of the seam during sewing?	The key to ensuring the straightness of the seam is to keep the fabric flat, control the speed and pressure of the sewing machine, and adjust the stitch length and thread tension in time.

TABLE IV: Examples of garment manufacturing QA task in case data.

Initial Retrieved Docs	Filtered Docs by DFCM	Reranking Time	FLAN-T5-XL		Llama-3-Instruct	
			ROUGE-L	BLEU	ROUGE-L	BLEU
5	4.926	0.029	22.40	1.35	25.94	1.55
10	6.954	0.048	25.92	1.54	28.36	1.72
15	7.232	0.077	26.07	1.52	28.39	1.70
20	7.456	0.101	26.11	1.54	28.57	1.73

TABLE V: Impact of the number of initial retrieved documents on filtering performance of DFCM.

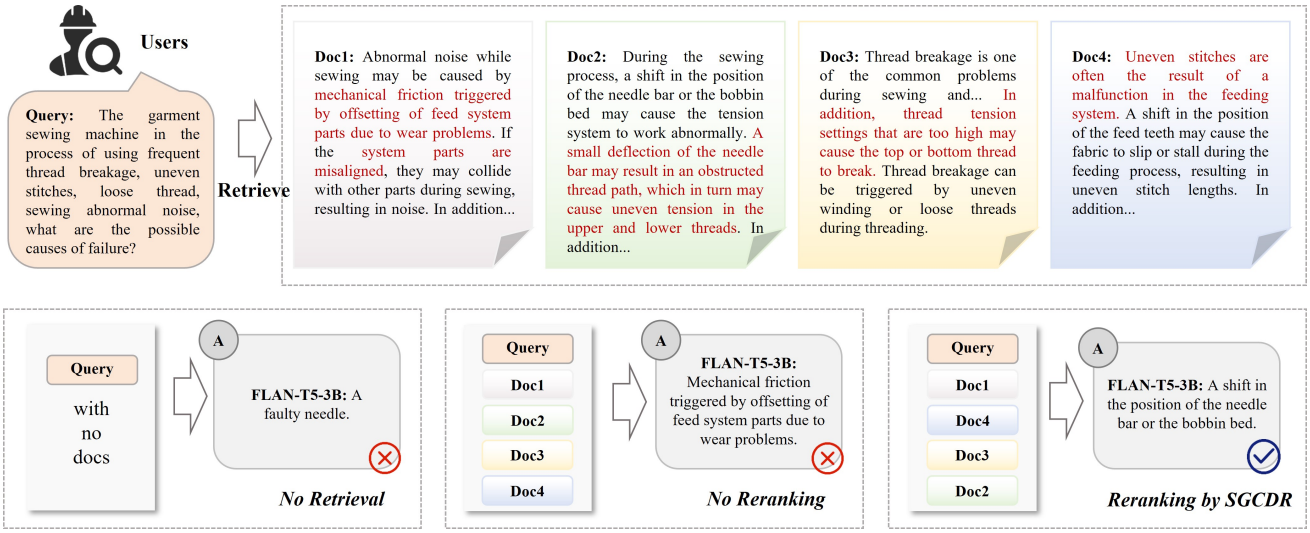


Fig. 5: Visualization of the retrieval reranking-enhanced quizzing task on SGDCR.

Setting	Avg. Score (GPT-4)	Avg. Score (GPT-4o)
No Retrieval	26	31
No Reranking	56	60
Reranking by SGDCR	68	74

TABLE VI: Expert evaluation scores (0–100) under different retrieval settings using GPT-4 and GPT-4o as subjective evaluators.

retrieval-augmented question answering. Examples of these question-answer pairs are illustrated in Table IV. We conducted both quantitative comparison experiments with preset labels and subjective evaluation experiments using an expert model to comprehensively assess the effectiveness of retrieval-augmented generation in question-answering tasks within the fashion manufacturing domain.

1) Case Setup: To further examine the choice of the initial number of retrieved document blocks (D) in our fashion manufacturing case study, we conducted a comparative analysis using $D = \{5, 10, 15, 20\}$. As shown in Table V, we report the average number of filtered document blocks by the DFCM module, re-ranking time, and the final generation performance under both FLAN-T5-XL and LLaMA-3-Instruct models. The results indicate that increasing D leads to only marginal gains in ROUGE-L and BLEU scores beyond D=10, while incurring noticeably higher re-ranking time. For example, with FLAN-T5-XL, the ROUGE-L score improves only slightly from 25.92 at D=10 to 26.07 at D=15, while the reranking time increases by over 60%. Similar trends are observed with LLaMA-3-Instruct. These findings demonstrate that retrieving 10 document blocks is already sufficient to provide highly effective responses to engineering problems in the fashion manufacturing domain, striking a favorable balance between performance and computational efficiency. Therefore, we adopt D=10 as the default setting throughout our case study.

Furthermore, Table V provides additional evidence of the effectiveness of the DFCM module. Even as the size of the candidate pool increases significantly, the number of documents retained by DFCM reaches the saturation point required by the LLM to answer the query effectively. This demonstrates DFCM's ability to efficiently identify useful information from a large set of redundant documents and filter out those with

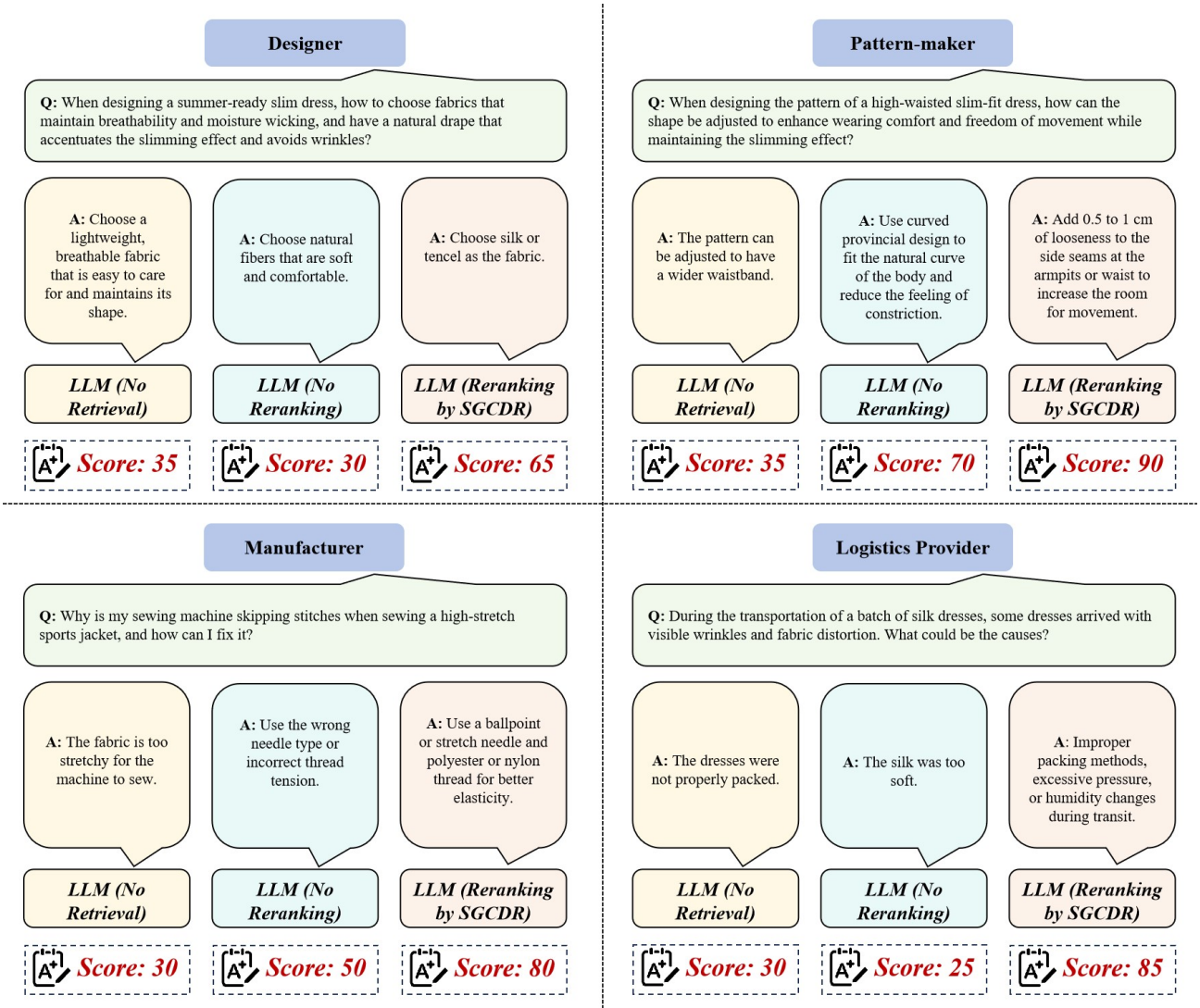


Fig. 6: Expert model GPT-4o's evaluation scores of retrieval-augmented reranking in the fashion manufacturing case study. *Note: Flan-T5-XL is used as the base LLM, and the Score denotes the subjective rating (0–100) given by GPT-4o.*

low relevance to the query.

2) Quantitative Comparison Experiments: We first conducted quantitative comparative experiments. Specifically, we compared SGDCR with BART-reranker, which demonstrated the best performance among the baseline methods on public datasets. The experiments were conducted using FLAN-T5-Large, FLAN-T5-XL and Llama-3-Instruct models. Given the long-text nature of the answers in this case study, we evaluated the results using ROUGE-L and BLEU metrics. The results in Fig. 4 demonstrate that the proposed SGDCR significantly outperforms both the no-reranking baseline and BART-reranker across both metrics and models. For FLAN-T5-Large, our method obtains a ROUGE-L score of 24.09 and a BLEU score of 1.47, showing consistent improvements over the No Reranking baseline and the BART reranker. For FLAN-T5-XL, our method achieves a ROUGE-L score of 25.92 and a BLEU score of 1.54, which are higher than the baseline without reranking and BART reranker. Similarly, for Llama-3-Instruct, our method achieves a ROUGE-L score of 28.36 and a BLEU score of 1.72, surpassing both the baseline without reclassification and the BART reclassifier. These results indicate that SGDCR effectively improves the quality of the retrieved documents, allowing the LLMs to generate more accurate and contextually relevant responses. To further illustrate the effectiveness of our method, we provide a visualization of the retrieval-augmented reranking process in Fig. 5. The visualization demonstrates that the reranked documents form a clear and logical chain of information, which significantly improves the accuracy and coherence of the generated answers. For example, in a case involving troubleshooting a sewing machine, the reranked documents provided a step-by-step guide that aligned perfectly with the query, enabling the LLM to generate a precise and actionable response. In contrast, the contextual logic of documents without reranking is confusing, resulting in less accurate answers.

3) ChatGPT-Based Expert Evaluation Experiment: To further validate the effectiveness of retrieval-augmented generation in the fashion manufacturing domain, we conducted an

expert subjective evaluation using ChatGPT-4 and GPT-4o as the evaluator. The evaluation focused on comparing the quality of answers generated by FLAN-T5-XL under three settings: no retrieval (No Retrieval), retrieval without reranking (No Reranking), and retrieval with reranking using SGDCR (Reranking by SGDCR). For each query in the test set of the case study data, we provided ChatGPT-4 and GPT-4o with a standardized instruction: "This is a common query in the fashion production process, and you are asked to compare the following three responses and rate them based on the comprehensiveness and accuracy of the answers, with a score range of 0-100." The query and corresponding answers from the three settings were then presented for evaluation. The results, reported in Table VI as average scores, highlight the necessity of retrieval augmentation, as both retrieval settings (No Reranking and Reranking by SGDCR) significantly outperformed the no retrieval baseline. More importantly, the answers generated with Reranking by SGDCR received the highest scores, demonstrating a notable improvement in comprehensiveness and accuracy compared to the No Reranking setting. This indicates that SGDCR effectively enhances the quality of retrieved documents, enabling the model to generate more precise and contextually relevant answers. To provide deeper insights, we visualized some evaluation cases of GPT-4o in Fig. 6, covering the entire fashion production cycle, including design, pattern making, production, and transportation. These cases illustrate the answer generated with Reranking by SGDCR provided a step-by-step solution with clear technical details, while the No Reranking setting returned a less structured and partially redundant response. The no retrieval baseline, in contrast, often produced generic or incomplete answers due to the lack of external knowledge support. In conclusion, the ChatGPT-based expert evaluation confirms the critical role of retrieval augmentation in improving answer quality and further validates the superiority of SGDCR in enhancing the logical coherence and accuracy of generated answers.

F. Discussion and Limitations

Despite the demonstrated efficacy of SGDCR in enhancing LLM-powered question answering for fashion manufacturing through retrieval reranking, as evidenced by our comparative experiments and case studies, several limitations warrant discussion. The current implementation's most notable constraint stems from the DFCM module, which intentionally models inter-document collaborative relationships to improve retrieval precision. While this design yields superior accuracy, it inherently limits the system's flexibility by imposing: (1) a fixed capacity constraint on the number of processable documents per training iteration, and (2) reduced adaptability to dynamic document sets compared to more lightweight alternatives. These architectural choices create an accuracy-flexibility trade-off that particularly manifests when handling variable-length industrial documentation. Future work will focus on developing an adaptive document window mechanism to preserve the DFCM's collaborative modeling advantages while eliminating its current scalability constraints.

V. Conclusion

This paper proposes a dynamic capsule routing network with semantic graph to optimize RAG in the fashion manufacturing industry, addressing challenges from specialized terminology and complex queries. SGDCR simulates team collaboration through a two-step process: (1) a capsule co-dynamic routing mechanism in a semantic graph captures complex contextual relationships and filters irrelevant documents, and (2) filtered documents are matched and re-ranked using deep semantic similarity and contribution scores to generate accurate, contextually coherent prompts. Experiments on open-domain QA datasets (HotpotQA, SQuAD) and a fashion manufacturing dataset show SGDCR outperforms existing methods on EM, F1, ROUGE-L, and BLEU scores across multiple LLMs, including FLAN-T5-XL and Llama-3-Instruct. A case study confirms its effectiveness in retrieving coherent document chains and generating precise answers. This work improves RAG in specialized domains and sets the stage for broader industrial applications and real-time optimization.

VI. Acknowledgments

The authors would like to express their sincere thanks to the financial support from the Hong Kong Research Grants Council (No. 15203025, T32-707/22-N, C7076-22GF), National Natural Science Foundation of China (No. 52305557), Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515011930), Research Institute for Advanced Manufacturing (RIAM) of The Hong Kong Polytechnic University (No. CDLU, CDLM, CDJX), Department General Research Fund (No. P0050805)and Intra-Faculty Interdisciplinary Projects (No. P0052206).

References

[1] R. Nayak and R. Padhye, "Introduction to automation in garment manufacturing," in *Automation in garment manufacturing*. Elsevier, 2018, pp. 1–27.

[2] S. Bian, C. Xu, Y. Xiu, A. Grigorev, Z. Liu, C. Lu, M. J. Black, and Y. Feng, "Chatgarment: Garment estimation, generation and editing via large language models," *arXiv preprint arXiv:2412.17811*, 2024.

[3] S. S. Ghidary, D. Chen, F. Mohammadi, A. E. P. Barceló, J. V. S. Luces, and Y. Hirata, "Innovating robotic garment handling through the integration of large language models and behavior trees," in *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2024, pp. 595–600.

[4] L. Ren, H. Wang, and Y. Laili, "Diff-mts: Temporal-augmented conditional diffusion-based aigc for industrial time series toward the large model era," *IEEE Transactions on Cybernetics*, 2024.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[6] A. Salemi and H. Zamani, "Evaluating retrieval quality in retrieval-augmented generation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 2395–2400.

[7] M. Glass, G. Rossiello, M. F. M. Chowdhury, A. R. Naik, P. Cai, and A. Gliozzo, "Re2g: Retrieve, rerank, generate," *arXiv preprint arXiv:2207.06300*, 2022.

[8] X. Zhang, Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang *et al.*, "mgte: Generalized long-context text representation and reranking models for multilingual text retrieval," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2024, pp. 1393–1412.

[9] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[11] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," *arXiv preprint arXiv:2402.03216*, 2024.

[12] D. Guo, R. Y. Zhong, Y. Rong, and G. G. Huang, "Synchronization of shop-floor logistics and manufacturing under iiot and digital twin-enabled graduation intelligent manufacturing system," *IEEE Transactions on cybernetics*, vol. 53, no. 3, pp. 2005–2016, 2021.

[13] L. Ren, J. Dong, L. Zhang, Y. Laili, X. Wang, Y. Qi, B. H. Li, L. Wang, L. T. Yang, and M. J. Deen, "Industrial metaverse for smart manufacturing: Model, architecture, and applications," *IEEE Transactions on Cybernetics*, 2024.

[14] W. Luo, K. Huang, X. Liang, H. Ren, N. Zhou, C. Zhang, C. Yang, and W. Gui, "Process manufacturing intelligence empowered by industrial metaverse: A survey," *IEEE Transactions on Cybernetics*, 2024.

[15] Y. Li, H. Ko, and F. Ameri, "Integrating graph retrieval-augmented generation with large language models for supplier discovery," *Journal of Computing and Information Science in Engineering*, vol. 25, no. 2, 2025.

[16] A. Gezdur and J. Bhattacharjya, "Innovators and transformers: enhancing supply chain employee training with an innovative application of a large language model," *International Journal of Physical Distribution & Logistics Management*, 2025.

[17] A. Narimani and S. Klarmann, "Integration of large language models for real-time troubleshooting in industrial environments based on retrieval-augmented generation (rag)," 2024.

[18] Z. Zhao, D. Tang, H. Zhu, Z. Zhang, K. Chen, C. Liu, and Y. Ji, "A large language model-based multi-agent manufacturing system for intelligent shopfloor," *arXiv preprint arXiv:2405.16887*, 2024.

[19] J. Lim, B. Vogel-Heuser, and I. Kovalenko, "Large language model-enabled multi-agent manufacturing systems," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2024, pp. 3940–3946.

[20] B. Zhu and C. Vuppalapati, "Enhancing supply chain efficiency through retrieve-augmented generation approach in large language models," in *2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*. IEEE, 2024, pp. 117–121.

[21] S. AlMahri, L. Xu, and A. Brintrup, "Enhancing supply chain visibility with knowledge graphs and large language models," *arXiv preprint arXiv:2408.07705*, 2024.

[22] S. Knollmeyer, M. U. Akmal, L. Koval, S. Asif, S. G. Mathias, and D. Groβmann, "Document knowledge graph to enhance question answering with retrieval augmented generation," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2024, pp. 1–4.

[23] 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)*. IEEE, 2023, pp. 1–8.

[24] J. Werheid, O. Melnychuk, H. Zhou, M. Huber, C. Rippe, D. Joosten, Z. Keskin, M. Wittstamm, S. Subramani, B. Drescher *et al.*, "Designing an llm-based copilot for manufacturing equipment selection," *arXiv preprint arXiv:2412.13774*, 2024.

[25] S. Kernan Freire, M. Foosherian, C. Wang, and E. Niforatos, "Harnessing large language models for cognitive assistants in factories," in *Proceedings of the 5th international conference on conversational user interfaces*, 2023, pp. 1–6.

[26] S. K. Freire, C. Wang, M. Foosherian, S. Wellsandt, S. Ruiz-Arenas, and E. Niforatos, "Knowledge sharing in manufacturing using large language models: User evaluation and model benchmarking," *arXiv preprint arXiv:2401.05200*, 2024.

[27] H. Choi and J. Jeong, "A conceptual framework for a latest information-maintaining method using retrieval-augmented generation and a large language model in smart manufacturing: Theoretical approach and performance analysis," *Machines*, vol. 13, no. 2, p. 94, 2025.

[28] I. Fernández, C. Aceta, C. Fernandez, M. I. Torres, A. Etxalar, A. Méndez, M. Agirre, M. Torralbo, A. Del Pozo, J. Agirre *et al.*, "Incremental learning for knowledge-grounded dialogue systems in industrial scenarios," in *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2024, pp. 92–102.

[29] Y. Yu, W. Ping, Z. Liu, B. Wang, J. You, C. Zhang, M. Shoeybi, and B. Catanzaro, "Rankrag: Unifying context ranking with retrieval-augmented generation in llms," *Advances in Neural Information Processing Systems*, vol. 37, pp. 121 156–121 184, 2025.

[30] S. Xu, L. Pang, J. Xu, H. Shen, and X. Cheng, "List-aware reranking-truncation joint model for search and retrieval-augmented generation," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 1330–1340.

[31] S. Wang, X. Yu, M. Wang, W. Chen, Y. Zhu, and Z. Dou, "Richrag: Crafting rich responses for multi-faceted queries in retrieval-augmented generation," *arXiv preprint arXiv:2406.12566*, 2024.

[32] M. Mortaheb, M. A. A. Khojastepour, S. T. Chakradhar, and S. Ulukus, "Re-ranking the context for multimodal retrieval augmented generation," *arXiv preprint arXiv:2501.04695*, 2025.

[33] G. d. S. P. Moreira, R. Ak, B. Schifferer, M. Xu, R. Osmulski, and E. Oldridge, "Enhancing q&a text retrieval with ranking models: Benchmarking, fine-tuning and deploying rerankers for rag," *arXiv preprint arXiv:2409.07691*, 2024.

[34] Y. Wang, R. Ren, J. Li, W. X. Zhao, J. Liu, and J.-R. Wen, "Rear: A relevance-aware retrieval-augmented framework for open-domain question answering," *arXiv preprint arXiv:2402.17497*, 2024.

[35] E. Song, S. Kim, H. Lee, J. Kim, and J. Thorne, "Re3val: Reinforced and reranked generative retrieval," *arXiv preprint arXiv:2401.16979*, 2024.

[36] J. Liu, Y. Ma, R. Zhao, J. Zheng, Q. Ma, and Y. Kang, "Listconranker: A contrastive text reranker with listwise encoding," *arXiv preprint arXiv:2501.07111*, 2025.

[37] K. Kim and J.-Y. Lee, "Re-rag: Improving open-domain qa performance and interpretability with relevance estimator in retrieval-augmented generation," *arXiv preprint arXiv:2406.05794*, 2024.

[38] H. Zhang, J. Song, J. Zhu, Y. Wu, T. Zhang, and C. Niu, "Rag-reward: Optimizing rag with reward modeling and rlhf," *arXiv preprint arXiv:2501.13264*, 2025.

[39] J. Dong, B. Fatemi, B. Perozzi, L. F. Yang, and A. Tsitsulin, "Don't forget to connect! improving rag with graph-based reranking," *arXiv preprint arXiv:2405.18414*, 2024.

[40] F. Tian, D. Ganguly, and C. Macdonald, "Is relevance propagated from retriever to generator in rag?" *arXiv preprint arXiv:2502.15025*, 2025.

[41] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[42] B. Liu, H. Zamani, X. Lu, and J. S. Culpepper, "Generalizing discriminative retrieval models using generative tasks," in *Proceedings of the Web Conference 2021*, 2021, pp. 3745–3756.

[43] X. Ma, L. Zhao, G. Huang, Z. Wang, Z. Hu, X. Zhu, and K. Gai, "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1137–1140.

[44] H. Li, K. Wu, C. Zheng, Y. Xiao, H. Wang, Z. Geng, F. Feng, X. He, and P. Wu, "Removing hidden confounding in recommendation: a unified multi-task learning approach," *Advances in Neural Information Processing Systems*, vol. 36, pp. 54 614–54 626, 2023.

[45] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, and X. Carreras, Eds. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. [Online]. Available: https://aclanthology.org/D16-1264

[46] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *arXiv preprint arXiv:1809.09600*, 2018.

[47] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," 2024.

[48] M. Günther, J. Ong, I. Mohr, A. Abdessalem, T. Abel, M. K. Akram, S. Guzman, G. Mastrapas, S. Sturua, B. Wang *et al.*, "Jina embeddings 2: 8192-token general-purpose text embeddings for long documents," *arXiv preprint arXiv:2310.19923*, 2023.