

Article

Intelligent Q&A System for Welding Processes Based on a Symmetric KG-DB Hybrid-RAG Strategy

Shuxia Ye ^{1,2} , Liwen Cai ¹, Yongwei Zhang ^{1,2} , Xiaoqi Xin ¹, Bo Jiang ¹ and Liang Qi ^{1,2,*} 

¹ School of Automation, Jiangsu University of Science and Technology, No. 666 Changhui Road, Zhenjiang 212114, China; yeshuxia@just.edu.cn (S.Y.); lwcai@stu.just.edu.cn (L.C.); ywzhang@just.edu.cn (Y.Z.); xqxin@stu.just.edu.cn (X.X.); 202210331115@stu.just.edu.cn (B.J.)

² Jiangsu Shipbuilding and Ocean Engineering Design and Research Institute, Zhenjiang 212100, China

* Correspondence: alfred_02030210@just.edu.cn

Abstract

This paper pioneers the use of the symmetrical Hybrid-RAG strategy in the ship welding process domain, addressing the problems of fragmented, unstructured knowledge storage, as well as the limitations of traditional Retrieval-Augmented Generation (RAG), particularly high retrieval noise and low accuracy when answering complex procedural queries. This study proposes an intelligent three-stage symmetric “Generate–Retrieve–Generate” framework for the ship welding process (SWP-Chat), supported by dual retrieval engines: a Neo4j knowledge graph for symbolic reasoning and a vector database for semantic retrieval. Unlike approaches that rely solely on LLM-based process planning, SWP-Chat uses the LLM to generate a logical form, then executes Cypher queries on Neo4j, enabling transparent traceability, precise entity–relation constraints, and deterministic retrieval. Meanwhile, the vector channel supplements unstructured or contextual welding information to enhance semantic coverage. To further improve efficiency, principal component analysis (PCA) was employed for vector dimensionality reduction, reducing average retrieval latency by 31% while retaining more than 95% variance. In addition, an explainable structural–confidence fusion formula integrates evidence from both engines to produce auditable and trustworthy industrial responses. Experimental evaluation demonstrates that the framework achieves an F1 score of 79.35%, greatly surpassing typical RAG systems.

Keywords: ship welding process; large language model; Neo4j; generate–retrieve–generate; hybrid-RAG



Received: 21 October 2025

Revised: 12 November 2025

Accepted: 13 November 2025

Published: 18 November 2025

Citation: Ye, S.; Cai, L.; Zhang, Y.; Xin, X.; Jiang, B.; Qi, L. Intelligent Q&A System for Welding Processes Based on a Symmetric KG-DB Hybrid-RAG Strategy. *Symmetry* **2025**, *17*, 1994. <https://doi.org/10.3390/sym17111994>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

China has remained the world’s largest manufacturing economy for 15 consecutive years according to the latest statistics from the Ministry of Industry and Information Technology [1]. Intelligent modernization of manufacturing processes is essential for improving efficiency, reducing costs, and ensuring product quality in an increasingly competitive environment. The shipbuilding sector, which is a prime example of high-end equipment production, deals with highly intricate welding procedures. These present difficulties with large amounts of various data and include a variety of welding materials, climatic circumstances, welding parameters, and process specification documents.

The development of intelligent question-answering systems in manufacturing has mainly progressed through two stages aimed at improving the management and utilization of process knowledge. Knowledge engineering techniques such as knowledge graphs

(KGs), which allow for accurate storage and logical reasoning for process parameters and specification clauses, were the mainstay of early systems [2–4]. However, these systems frequently rely on formal query languages like Cypher or manually coded rules, which limits practical human–machine interaction. In addition, they encounter difficulty integrating and using unstructured process documentation, and real-time knowledge updates and maintenance are expensive.

Due to its superior natural language understanding and unstructured text-processing capabilities, RAG technology has become the standard option in industrial Query and Answer (Q&A) as large language models have grown in popularity [5]. RAG greatly improves system interaction and knowledge coverage by embedding extensive process documentation into vector spaces, allowing for effective semantic search.

Despite these efforts, the strictly logical and parameter-coupled vertical industrial scenario of ship welding still faces three major core challenges:

- (1) Conflict between the retrieval paradigm and logical reasoning: The welding process involves complex question-answering tasks that rely on multi-parameter coupled decisions across various material performance indicators. Traditional RAG uses a “retrieve first, then generate” blind recall strategy. This method aggressively retrieves a large number of document fragments, leading to an explosion of retrieval noise. Especially in complex process Q&A that involves selecting multiple parameters based on several material performance metrics, the model’s logical chain easily breaks, and structural information is lost. This makes the output unreliable for high-risk welding decision support, where precision is critical. There is an urgent need for an innovative approach to isolate retrieval noise before the recall step.
- (2) Lightweight and real-time knowledge updating: Industry standards, classification society rules, and internal specifications change rapidly; However, existing GraphRAG methods require recalculation of the embeddings for the entire graph, resulting in prohibitively high update costs ($O(N^2)$). Once a new rule is issued after system deployment, it often necessitates several hours of downtime for knowledge reconstruction [6]. This contradicts the manufacturing sector’s requirement for real-time knowledge and continuous system operation.
- (3) Lack of answer credibility and auditability: For safety-critical applications in the shipbuilding industry, every engineering decision derived from the Q&A system must be fully traceable and justifiable. Current RAG strategies only provide an opaque black-box score based on text similarity. This lack of transparent graph paths and standard clause citations prevents engineers from confidently making judgments, failing to meet crucial auditability and accountability requirements.

In summary, this paper presents SWP-Chat, the first hybrid-RAG intelligent query-answering system for shipbuilding processes based on the “Generate–Retrieve–Generate” symmetric framework. In contrast to conventional RAG systems, the SWP-Chat system uses sophisticated LLM reasoning powers to first produce logical phrases, as shown in Figure 1. This logical form generation drives cypher queries on the knowledge graph, which work jointly with vector retrieval to effectively filter retrieval noise before the recall stage. This system design makes industry-grade advances in real-time knowledge updates and answer dependability, improves retrieval accuracy for complex situations, and successfully lowers noise interference in complex process queries. In complex process contexts, it offers a feasible route for intelligent question answering and decision assistance. The following are this paper’s primary contributions:

- (1) This study builds a symmetric KG+Vector dual engine that supports 11 types of welding entities, making it the first to apply symmetrical design to ship welding technology.

This architecture achieves mirrored balance between the vector channel and knowledge graph channel. This bidirectional symmetry mechanism fundamentally isolates pre-retrieval noise, significantly enhancing the accuracy of complex reasoning and the coherence of question answering, with an F1 score improvement of 40.89%.

- (2) This paper avoids global graph embedding by explicitly adopting Neo4j Cypher symbolic queries. Furthermore, by integrating PCA vector dimensionality reduction technology, we successfully shortened the average retrieval latency by 31%. This achievement enables agile, minute-level updates of industrial knowledge, thereby circumventing the lag and system interruptions caused by the high recalculation cost of GraphRAG's graph embedding and ensuring the continuity of knowledge services.
- (3) This work suggests an explainable fusion process that strikes a balance between source weighting and structural confidence. It offers transparent, auditable reliability proof for every choice made in shipbuilding scenarios, guaranteeing complete outcome traceability and adherence to strict safety regulations.

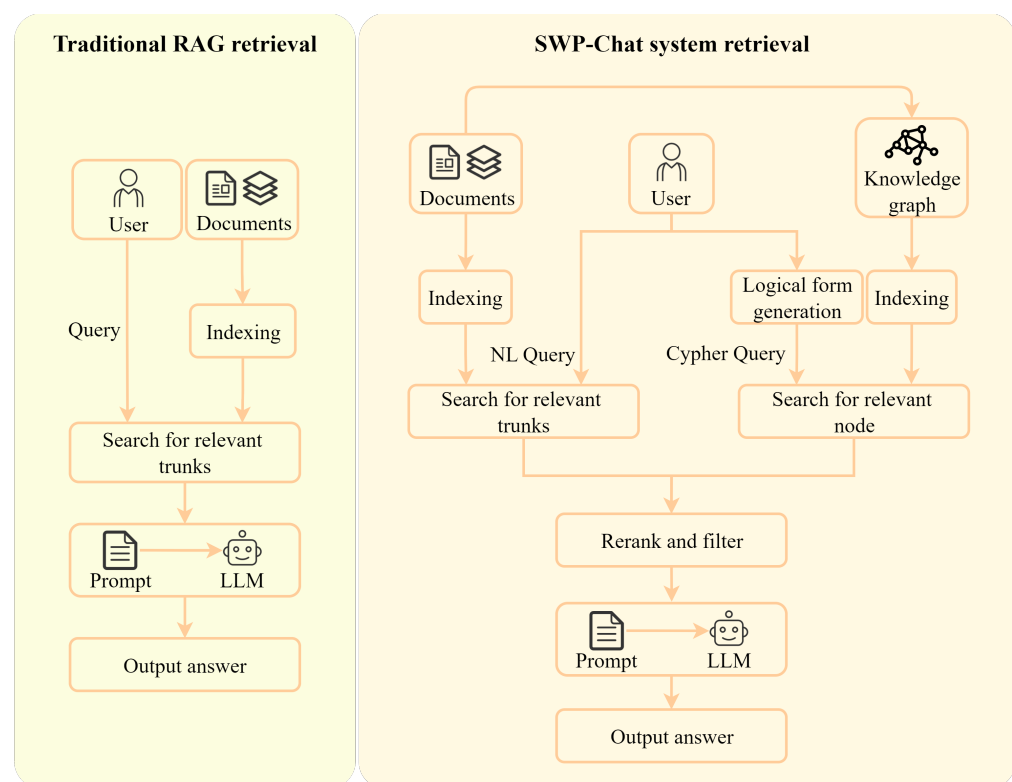


Figure 1. Comparison of retrieval methods between traditional RAG and SWP-Chat systems.

This article's main organizational framework is outlined as follows: Section 2 reviews the related work of process knowledge graph, industrial RAG, and Hybrid-RAG. Section 3 elaborates in detail on the core methods of the proposed Hybrid-RAG intelligent question-answering system, including the system architecture, natural language question parsing and logical generation, and the construction and retrieval mechanisms of the knowledge graph and vector database. Section 4 introduces the experimental design, dataset construction, and evaluation metrics. Section 5 presents the results of ablation experiments, comparative experiments, and subjective evaluations, as well as detailed discussions. Section 6 summarizes the full text and looks forward to future research directions.

2. Related Work

2.1. Process Knowledge Graph

In the manufacturing industry, technological processes include bridge product design and production [7]. As product requirements have diversified, process planning has become increasingly complex and variable. Knowledge graphs are well-suited for modeling complex entities and their relationships and have therefore become an important tool for managing industrial process information.

Early manufacturing knowledge graphs were primarily built using ontology-driven methods. For example, Ford Motor Company began storing production knowledge in ontologies in 2017, enabling intelligent assistance for complex assembly processes. This allowed for intelligent support for complex product assembly processes. Yahya et al. [8] introduced the Generalized Ontological Model (RGOM) ontology framework based on the Resource Description Framework/Web Ontology Language (RDF/OWL) in the manufacturing area, combining elements like workpieces, welding procedures, and welding robots into a single semantic model. To check for process compliance, they used Semantic Web Rule Language (SWRL) rules. Similarly, a top-down strategy of “defining ontologies before populating entities” is emphasized in Meckler’s [9] procedure model for building knowledge graphs for industry applications. After using capability questions to define domain boundaries, it uses OWL constraints on classes, properties, and instances to accomplish controlled evolution of process knowledge graphs. However, these approaches still depend heavily on structured data. Their ability to extract and integrate knowledge is limited when facing semi-structured models, unstructured logs, or maintenance reports. In later levels of reasoning, this increases dependence on contextual interpretation, which often leads to conceptual ambiguity and unstable reasoning outcomes.

To overcome these limitations, researchers have begun combining knowledge graphs with intelligent algorithms, such as deep learning and statistical models, to improve automation and reasoning in process knowledge applications. The BITC deep network was proposed by Guo [10] et al. to automatically extract triples from large amounts of machine-generated free text. These triples were written into Neo4j following cosine similarity-based entity alignment, allowing process knowledge graphs to be constructed from semi-structured data from beginning to end. Liu [11] et al. developed a knowledge graph-based industrial knowledge question-answering system by using a naive Bayes classification model for semantic categorization. In order to increase the efficiency of process creation, Zhang [12] et al. combined deep learning with knowledge graphs for macro-level process planning.

Building upon this foundation, subsequent research increasingly focused on knowledge reasoning and dynamic generation. For ship heterogeneity models, Dong [13] built a multi-element process knowledge graph that allows for the logical creation of production processes and the quick reuse of case knowledge. Some research has moved toward intelligent question answering and assembly process planning as application demands have increased. For example, Zhou [14] created a knowledge network about the assembly process using a hierarchical modeling technique. They produced intelligent assembly process planning by combining distributed graph embedding with a sequence convolutional network, which greatly increased assembly automation and efficiency. In order to enable intelligent inference of product assembly sequences, Shi [15] created a multidimensional integrated assembly resource knowledge graph and combined it with the BERT model.

To summarize, the use of knowledge graphs in the field of current processes has evolved from ontology-driven to deep learning-integrated and from static knowledge organization to dynamic reasoning. In addition, KGs have shown great promise in decision support and manufacturing process planning, particularly in complicated modeling and

knowledge reuse, where they have seen early success. However, current knowledge graph-based industrial systems still have notable limitations that make it difficult to meet the increasing complexity of modern industrial production:

- (1) Insufficient integration of unstructured knowledge: Many industrial process documents are text-based and unstructured. This knowledge is underutilized, which results in incomplete representation within knowledge graphs.
- (2) Weak intelligent question answering and interactivity: Existing KG-based systems focus mainly on retrieval and recommendation. They lack flexible natural language interaction, making it difficult to support engineers' daily needs for quick querying and reasoning.

Therefore, with the breakthroughs of LLMs in natural language processing and reasoning capabilities, leveraging the language understanding ability of LLMs to make up for the shortcomings of KG in unstructured knowledge fusion and natural language interaction has become the key to the development and large-scale application of industrial intelligent question-answering systems.

2.2. Industrial RAG Technology

RAG is an effective method for combining LLMs with external knowledge sources and has been widely used to improve accuracy and reliability in specialized question-answering tasks. In industrial settings, RAG strategies have been applied to tasks such as equipment diagnostics, fault prediction, and path planning [16–18], providing intelligent solutions for the optimization of process parameters, diagnosis of malfunctions, and forecasting of quality outcomes. Compared with fine tuning, RAG reduces training costs by retrieving relevant documents first, then generating responses. It also reduces the risk of LLM hallucination and improves adaptation to domain-specific content. For instance, Werheid [19] et al. applied RAG to a “co-pilot” system for manufacturing equipment selection, providing engineers with high-quality references to address skill shortages and resource constraints. The RAGraph framework by Jiang et al. [20] supports both static and dynamic data, maintaining high performance in manufacturing scenarios without requiring fine tuning for specific tasks. In the sPeCialized Knowledge and Rationale Augmentation Generation (PINKE-RAG) framework, Wang et al. [21] ensure coherent understanding and accurate responses to industrial process texts through a closed-loop system integrating domain-specific knowledge retrieval and LLM reasoning.

However, while RAG enhances text retrieval and single-hop reasoning capabilities, it still faces the following limitations in manufacturing processes:

- (1) Insufficient utilization of structured knowledge: RAG focuses on document fragment retrieval, making it difficult to directly manipulate process entities and relationships, leading to imprecise understanding of parameter associations and process flows.
- (2) Limited complex logical reasoning: For complex queries involving cross-document, multi-table, or multi-parameter dependencies, RAG often misses key reasoning steps, leading to incomplete answers.
- (3) Lagging knowledge updates and evolution: RAG retrieval repositories rely on manual document imports for updates, failing to reflect real-time iterations of new materials and processes.

In this paper, our SWP-Chat system introduces the symmetrical Hybrid-RAG technology to the manufacturing process domain. By integrating the structured advantages of knowledge graphs with the semantic matching capabilities of vector retrieval, it aims to effectively address the aforementioned shortcomings in unstructured knowledge fusion,

complex reasoning limitations, and knowledge update delays. This approach seeks to achieve more efficient, precise, and reliable intelligent question answering.

2.3. Hybrid-RAG Technology

Hybrid-RAG combines the semantic matching of vector databases with the structured reasoning of knowledge graphs, overcoming weaknesses of single-path RAG in complex reasoning tasks. Works like those of Barron [22] and Opoku [23] have confirmed Hybrid-RAG's efficacy in industries like healthcare and finance. Additionally, Hybrid-RAG was used by Russell-Gilbert [24] for adaptive anomaly detection, and Sarmah [25] also suggested unique knowledge graph–vector fusion frameworks to improve the accuracy and efficiency of information extraction.

However, applying Hybrid-RAG in industrial environments remains challenging. First, the majority of the work that has already been done pertains to fields that are more static than manufacturing, such as healthcare and finance, where real-time knowledge updates are less important. There are very few examples of direct industrial application. GraphRAG [26], the current major fusion approach, integrates the complete knowledge structure with text vector databases after encoding it into a continuous vector space using graph embedding. In addition, graph embedding methods introduce a black-box problem and do not meet industrial requirements for traceability and answer auditability. Due to a black-box problem in the answer generation process caused by this graph embedding technique based on numerical similarity, it is challenging to meet industry criteria for answer auditability and reliability. Furthermore, graph embedding has a very high recalculation cost, which significantly restricts real-time knowledge updates.

Given the aforementioned challenges and existing research gaps, the SWP-Chat system proposed in this paper builds upon traditional Hybrid-RAG strategies. It explicitly adopts Neo4j Cypher symbolic queries for knowledge graph retrieval, eschewing the graph embedding technique frequently employed in industrial GraphRAG systems. The significant update costs and black-box problems that come with graph embedding are essentially fixed by this. Although several recent studies have combined knowledge graphs with vector databases for hybrid retrieval in industrial domains [27], these applications mainly target general smart manufacturing or equipment maintenance and do not address the highly domain-specific constraints of ship welding processes. In contrast, the shipbuilding industry lacks an interpretable, auditable, and standards-aligned knowledge reasoning system. The system presents a “Generate–Retrieve–Generate” symmetric framework, which builds on this foundation. It makes use of LLMs' sophisticated reasoning powers to produce logical forms first, then accurately drive dual-engine collaborative retrieval before combining the results to produce outputs in natural language. This efficiently suppresses retrieval noise and greatly improves the accuracy of complex industrial Q&A. The system processes the vector database using PCA vector dimensionality reduction, which substantially lowers query latency and further optimizes performance. In order to achieve industrial-grade trustworthiness and auditability, we simultaneously suggest an explainable fusion formula that integrates source weighting and structural confidence to meet the demanding dependability requirements of industrial scenarios.

3. Methodology

3.1. Overall System Architecture

In addition to the structurally controllable and semantically enriched “Generate–Retrieve–Generate” framework, the SWP-Chat system adopts a symmetric dual-channel fusion design. This design substantially enhances response accuracy, efficiency, and interpretability at an industrial level in ship welding scenarios and the efficiency of the

response and the interpretation of industrial grade in the context of ship welding processes, in addition to successfully solving logical confusion and structural information gaps in complex question–answer scenarios. Figure 2 shows the particular architecture.

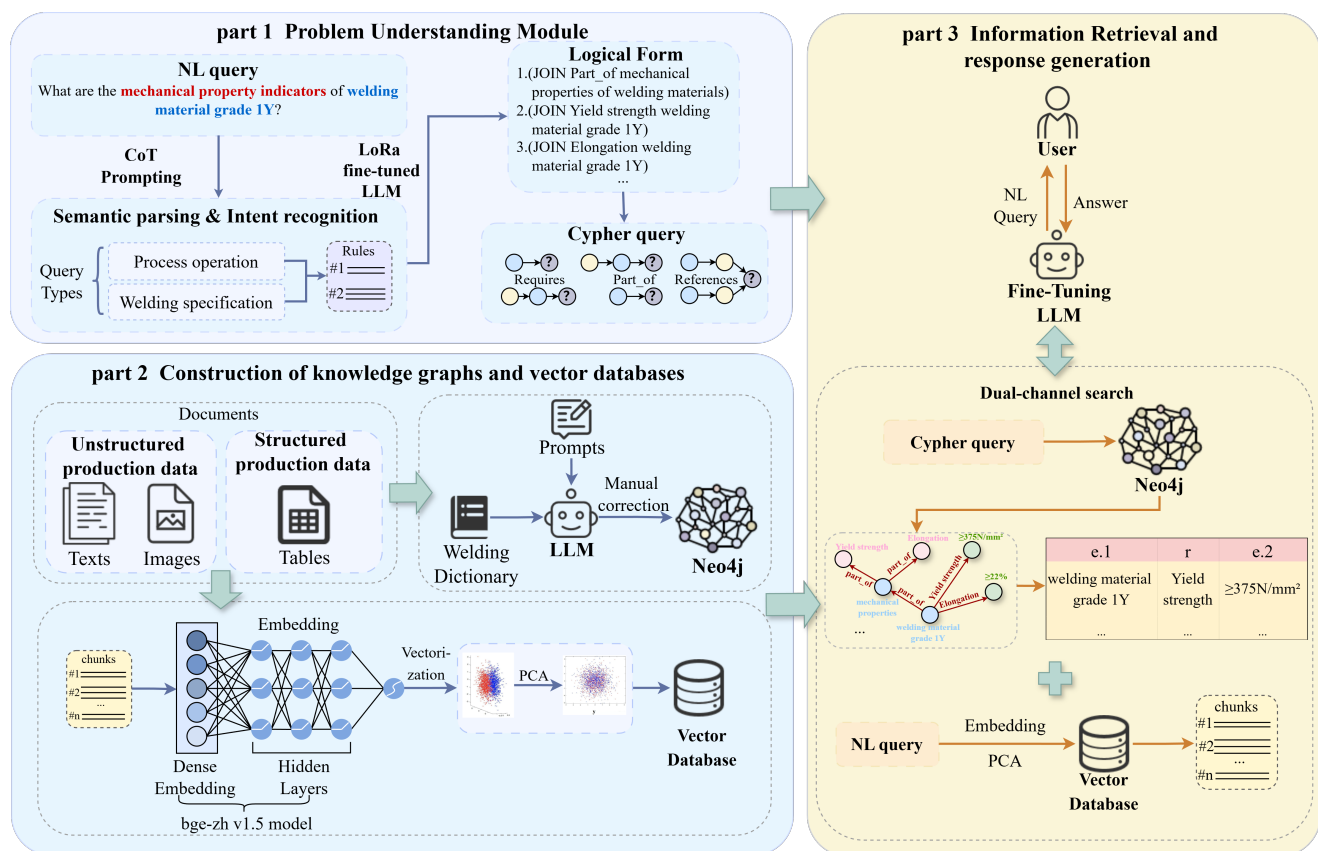


Figure 2. System framework.

The system comprises three core modules: the User Question Understanding Module, the Database Construction Module, and the Information Retrieval and Response Generation Module.

The User Question Understanding Module is responsible for parsing natural language questions and identifying user intent. It provides logical forms using template matching, question classification, and keyword extraction to support downstream retrieval.

The Database Construction Module includes the building of a structured knowledge graph database based on Neo4j and the construction of an unstructured vector database using semantic embedding. It also incorporates PCA algorithms for dimensionality reduction of the vector database to enhance retrieval efficiency.

Under the Retrieval and Response Generation Module, using graph retrieval, the LLM initially establishes the fundamental question path and response structure. It then incorporates additional information from the Faiss vector database to provide contextual explanation and boundary refinement. This strategy guarantees the system's continued robust scalability and accuracy.

This study uses a "semantic parsing-logic generation-structural querying-answer generation" symmetric paradigm of "generation–retrieval–generation" in which the knowledge-graph channel and vector-retrieval channel are driven by the same logical form developed in the first stage. The two retrieval processes run in parallel: the KG channel performs Cypher-based symbolic reasoning with strict entity–relation constraints, while the vector channel retrieves semantically related unstructured knowledge. This

approach produces a domain-specialized Q&A system that maintains structural control and efficient response generation while preserving broad semantic coverage.

3.2. Problem Analysis and Logic Generation

This paper employs CoT-Prompting and a Naive Bayes classifier to rapidly perform semantic analysis and retrieve the intent of user queries. Additionally, it accomplishes phrase segmentation within user queries by constructing a TF-IDF feature extractor based on specialist welding terminology and jieba word segmentation. After segmentation, the findings are sent to an LLM that has been fine-tuned by LoRA to produce logical forms. This method shows more flexibility in addressing real-world issues in industrial contexts, like vague claims, sparse data, and non-standardized vocabulary.

3.2.1. Query Parsing and Intent Recognition

User natural language queries are initially received by large language models. However, because natural language queries are diverse and ambiguous, direct processing often leads to misunderstanding and inefficiency. Therefore, the primary task is to transform them into a more structured intermediate representation. To address this, the system applies a multi-stage parsing workflow, as shown in in Figure 3. It first utilizes the jieba tokenization tool to segment user questions into phrases. This allows for initial identification of candidate entities and relations, assisted by a predefined domain lexicon for similarity matching.

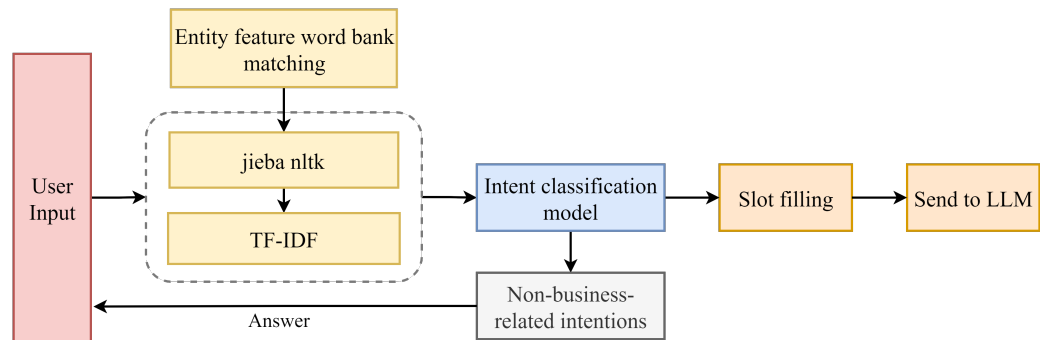


Figure 3. Intent recognition flowchart.

Subsequently, the segmented results are vectorized using the TF-IDF feature extractor [28]. TF represents term frequency, indicating how often a term appears in user queries, while IDF denotes inverse document frequency, reflecting a term's rarity across the entire corpus. The TF-IDF formula is expressed as follows:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

$$IDF_i = \lg \frac{|D|}{1 + |j : t_i \in d_j|} \quad (2)$$

$$T-I_{i,j} = TF_{i,j} \times IDF_i \quad (3)$$

Here, $n_{i,j}$ denotes the number of occurrences of word i in question j and; $\sum_k n_{k,j}$ represents the total number of words in question j ; $|D|$ denotes the total number of documents in the corpus; d_j represents the j -th document in the corpus; $1 + |j : t_i \in d_j|$ indicates the number of questions containing word t_i ; $T-I_{i,j}$ signifies the importance of word i in question j , accounting for its global significance across the entire question set (D).

Statistical analysis shows that multi-hop queries account for only 9.6% of Q&A samples in real welding data. Traditional complexity classification is not applicable to this system

due to the unbalanced distribution of data. Therefore, the system classifies user intent into two categories: welding requirements and process activities. This classification offers significant semantic guidance and is more in line with industry demands.

To achieve this intent classification, the system models question texts using a naive Bayes classifier [29] combined with a constructed keyword feature dictionary. Sample features include core term frequency, contextual collocation phrases, and structural template matching. The classifier is trained on a small, manually annotated dataset of question types and rapidly categorizes user questions during the prediction phase. The naive Bayes implementation of intent classification is expressed as follows:

$$f(x) = \arg \max_{y_k} P(y_k) \prod_i^n P(x_i | y_k), \quad k = 1, 2, \quad (4)$$

where x_i denotes the i -th feature in query x , y_k represents one of two intent categories (process operation/welding specification) present in the query, and $P()$ denotes probability. After completing intent classification, the system performs slot filling to obtain complete structured information. The completed structured queries are then passed to the LLM for logical form generation and answer synthesis.

Furthermore, the intent recognition module conducts a preliminary evaluation for user-posted “boundary questions” that are not covered by the system’s knowledge graph and vector database. The system automatically treats the query as having a “non-business-related intent” and uses the LLM’s own general knowledge to generate a response once it has concluded that retrieving the information from the knowledge base is not an appropriate way to answer it. To ensure auditability, the system explicitly notifies users when an answer is generated solely by the LLM rather than retrieved from vetted knowledge sources. It shows that these responses come directly from the LLM itself, not from organized knowledge sources or other publications, and are, thus, solely offered for reference. By keeping the Q&A process transparent and ensuring that the system may still provide feedback on unknown queries, this technique helps users avoid misinterpreting the origins of the answers.

3.2.2. LLM Logic Generation and LoRA Fine Tuning

To enable LLMs to convert parsed queries into machine-interpretable logical forms, the model is fine-tuned using LoRA. The core of the LoRA algorithm involves introducing a parallel low-rank branch alongside the Transformer layer, performing a dimension reduction followed by dimension expansion. The specific structure is illustrated in Figure 4.

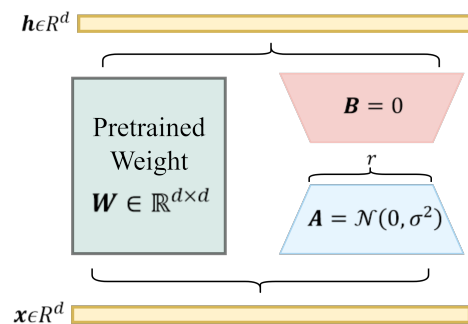


Figure 4. LoRA policy branch.

Only the low-rank matrix **A** (down-projection) and matrix **B** (up-projection) are trained, while the pretrained weights remain frozen. The model’s input and output dimensions remain unchanged. During output, the **BA** matrix is merged with the parameters of the pre-trained model. This approach minimizes trainable parameters and GPU memory

usage while preserving fine-tuning performance. The following formula can be used to calculate the neural network layer's output vector (**h**) during fine tuning:

$$\mathbf{h} = \mathbf{W}\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}, \mathbf{B} \in \mathbb{R}^{d \times r}, \mathbf{A} \in \mathbb{R}^{r \times k}, \quad (5)$$

where **x** represents the input vector before fine tuning; **W** denotes the pre-trained weight matrix; *d* is the output feature dimension of **W**; *r* is the rank of the low-rank matrix; and $r \ll \min(d, k)$, where *k* is the input feature dimension of **W**.

To enable accurate logical form generation, the DeepSeek-V3 base model is fine-tuned using LoRA. In this study, the rank is set to $r = 8$, the scaling factor is set to $\alpha = 16$, and a dropout rate of 0.1 is applied to prevent overfitting. Training is performed using AdamW with a learning rate of 2×10^{-4} for 3 epochs, which provides stable convergence without increasing computational overhead. The full hyperparameter configuration and rationale for these choices are provided in Appendix A.

A logical form is a structured representation of a natural language question. Taking S-expressions as an example, a logical form typically consists of projections and various operators. The projection operation represents a one-hop query on the triplet (e_1, r, e_2) over either e_1 or e_2 , where $(?, r, e_2)$ is represented as $(JOINr e_2)$ and $(e_1, r, ?)$ is represented as $(JOIN(R \ r) e_1)$. Practical application examples are shown in Table 1.

Table 1. Example of generating logical forms.

Query	Logical Forms
What are the mechanical property indicators of welding material grade 1Y?	1. (JOIN Part_of mechanical properties of welding materials) 2. (JOIN Yield strength welding material grade 1Y) 3. (JOIN Elongation welding material grade 1Y)

3.3. Construction of Knowledge Graphs and Vector Databases

3.3.1. Definition of Welding Process Entities

Ship welding process knowledge includes extensive tabular data, images, process resources, and testing information. Therefore, welding process entities are divided into 11 categories, each assigned a corresponding label based on technical standards and literature [30]. Table 2 provides specifics.

Table 2. Welding process knowledge entity labels.

ID	Entity Label	Entity Example
1	Welding materials	Welding rods, flux, and associated gases
2	Welding method	Shielded metal arc welding, CO ₂ gas shielded welding
3	Welding equipment	Torches, automatic gravity feeders, and gas chisels
4	Welding operation	Groove preparation, arc striking, tack welding, and workpiece heating
5	Welding base metal	Steel and aluminum alloys
6	Welded structure	Welding position, groove type, and joint design
7	Welding quality	Weld formation factors, welding defects, and various tests
8	Process parameter	Current, voltage, diameter, and wire feed speed
9	Table	-
10	Image	-
11	Others	-

3.3.2. Graph Construction Based on Large Language Models

LLMs now show a great deal of promise for ontology building. Domain-specific fine-tuning prompts can be used to improve the domain adaptability of these models, as well as to build and refine ontologies. Therefore, the “Materials and Welding Specifications” document published by the CCS in 2024 [31] is the source of pertinent welding process content that is extracted in this work. The text input is preprocessed before being loaded into a sizable language model that is set up with pertinent instructions to extract appropriate symmetric triples. This triplet architecture imposes fundamental structural symmetry on knowledge representation, wherein each relation serves as a bilateral link between two balanced entities. Following manual refinement and expert validation, these are imported into Neo4j to construct a complete knowledge graph, as shown in Figure 5. Ultimately, 1583 valid entities and 2088 triples were extracted from the selected document sections.

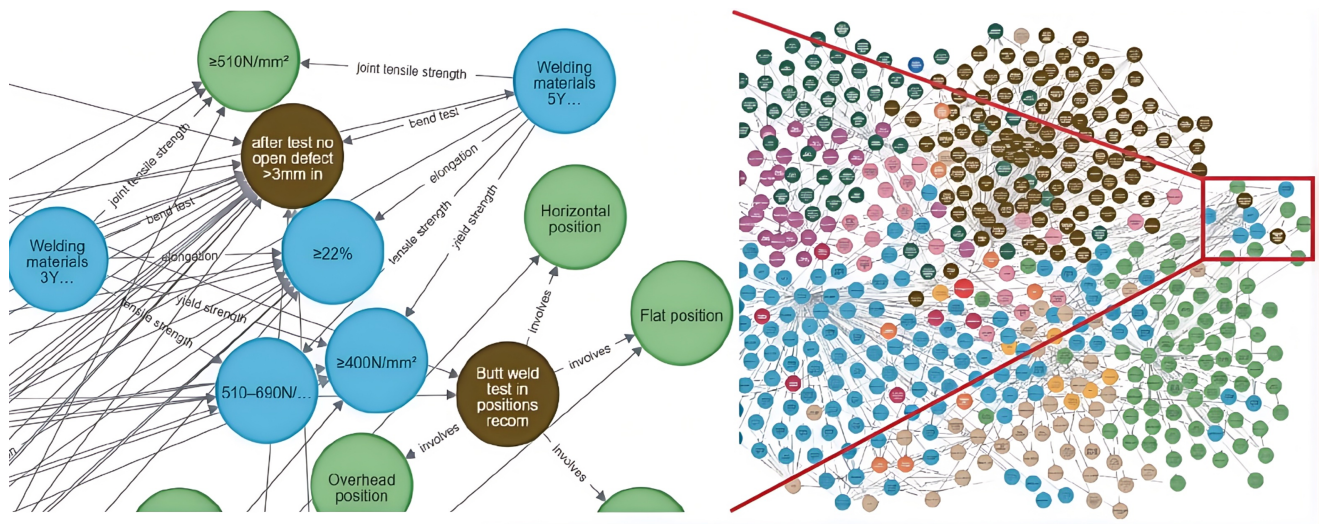


Figure 5. Knowledge graph of ship welding processes (partial).

3.3.3. Document Vector Database Construction

A Faiss vector database is constructed from both structured and unstructured text sources, following a structure-first, semantics-enhanced fusion strategy. Cosine similarity is used for vector retrieval, and Z-score standardization is applied to ensure consistent scaling across embedding dimensions. The Faiss index is implemented using the IVF-Flat structure, with $nprobe = 16$ to balance retrieval accuracy and latency. Therefore, the vector database is built according to the pipeline shown in Figure 6.

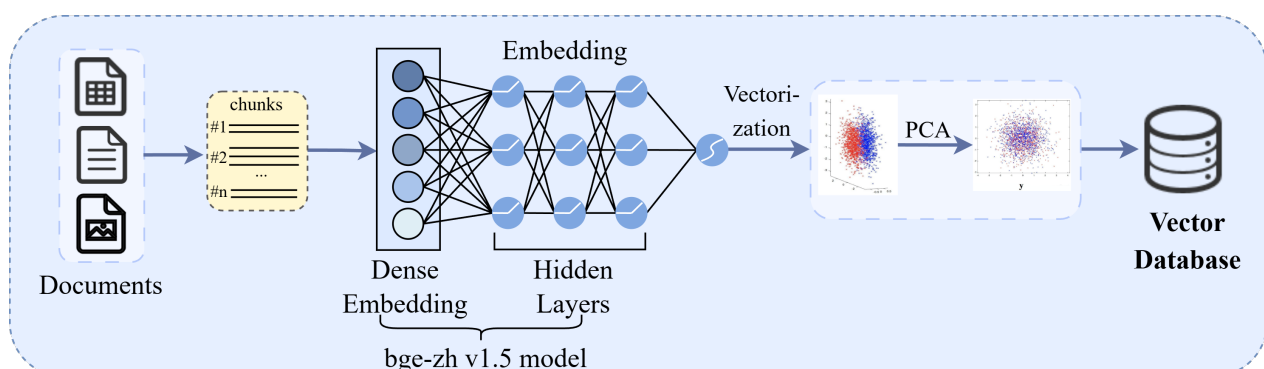


Figure 6. Vector database construction process.

In order to guarantee semantic coherence, this research first uses the Embedding model (bge-zh-v1.5) to divide the CCS “Materials and Welding Specifications” document

into segments of 128–512 tokens at the granularity of “paragraph + figure caption”, with a 20% stride. Semantic vectors with 1024 dimensions are then created from these parts. Finally, the 1024-dimensional embeddings are reduced using PCA. This enables initial coarse retrieval with reduced vectors, followed by top-K refinement using the original 1024-dimensional embeddings, thereby improving retrieval speed while maintaining accuracy.

As a classical linear dimensionality reduction method, PCA effectively removes redundant information and noise while preserving core semantic features. In the specific implementation process, we first applied Z-score normalization to the document embedding vectors to ensure the data had a mean of 0 and a variance of 1. Subsequently, the covariance matrix of the standardized data was calculated, and the eigenvalues and eigenvectors were extracted via eigen-decomposition, as shown in Figure 7. Considering the high process performance requirements in the domain of ship welding technology and based on the cumulative contribution rate of the eigenvalues, we ultimately chose a target dimension that retains 95% of the data variance for the dimensionality reduction transformation. Specifically, the original 1024-dimensional vector database was reduced to 280 dimensions. The corresponding pseudocode for this dimensionality reduction task is provided in Algorithm 1.

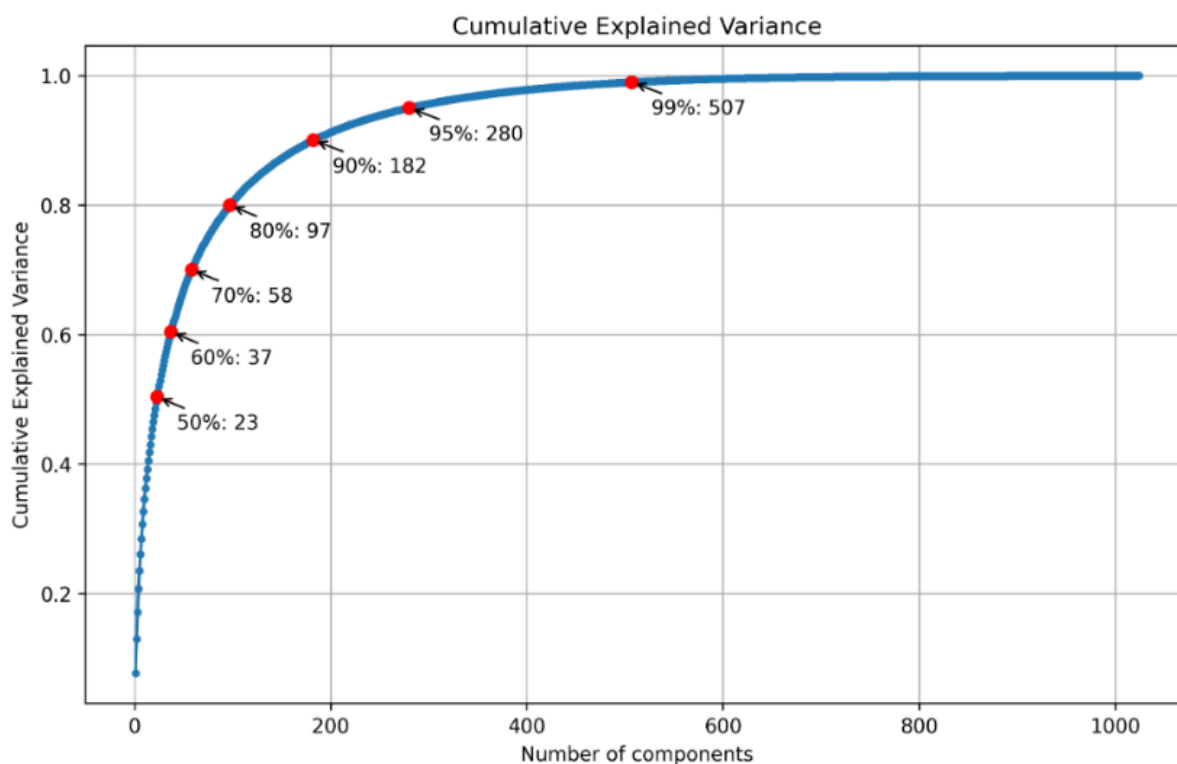


Figure 7. Cumulative explained variance before dimension reduction.

Algorithm 1: PCA-Based Vector Dimension Reduction

Input: Set of original high-dimensional vectors V^h

Output: Set of reduced-dimension vectors V^l

- 1 Compute covariance matrix: $Cov = \text{covariance}(V^h)$;
 - 2 Compute eigenvalues and eigenvectors: $E^{val}, E^{vec} = \text{eigen}(Cov)$;
 - 3 Select principal components: choose the first k eigenvectors based on cumulative explained variance threshold;
 - 4 Data projection: $V^l = V^h \cdot E^{vec}[:, 1 : k]$;
 - 5 **return** V^l ;
-

Analysis of the vector database after dimension reduction reveals relevant parameter metrics, as shown in Figure 8 and 9.

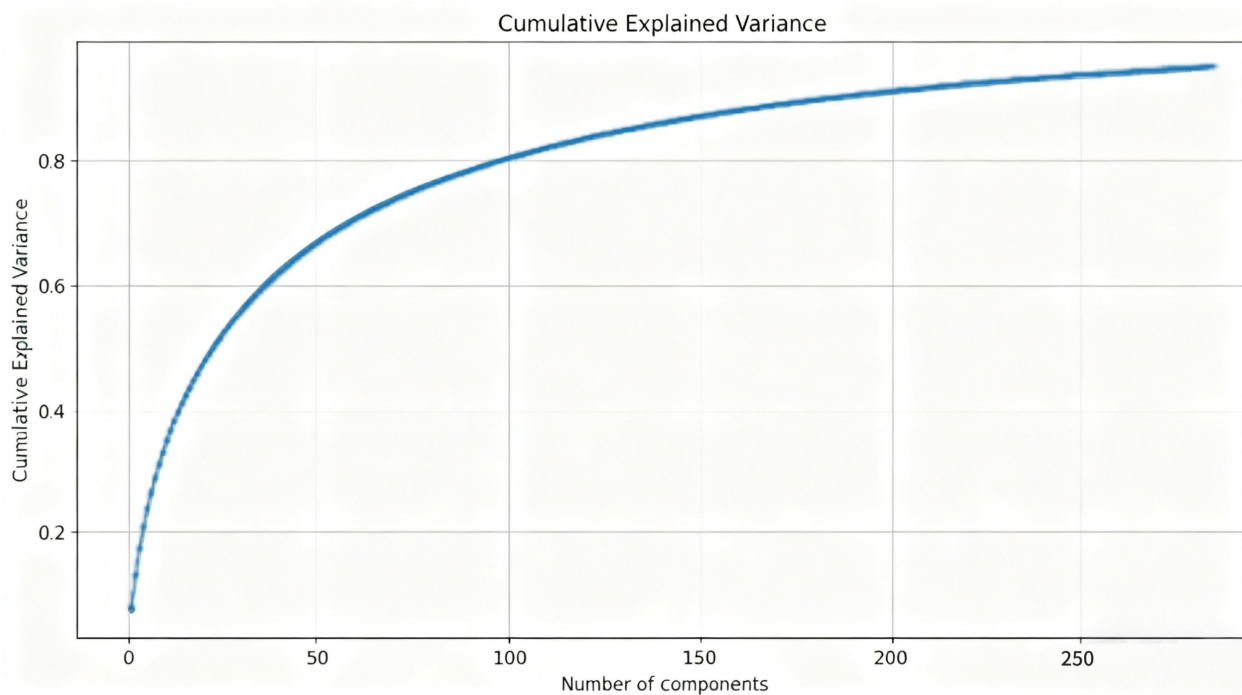


Figure 8. Cumulative explained variance for the first 280 dimensions.

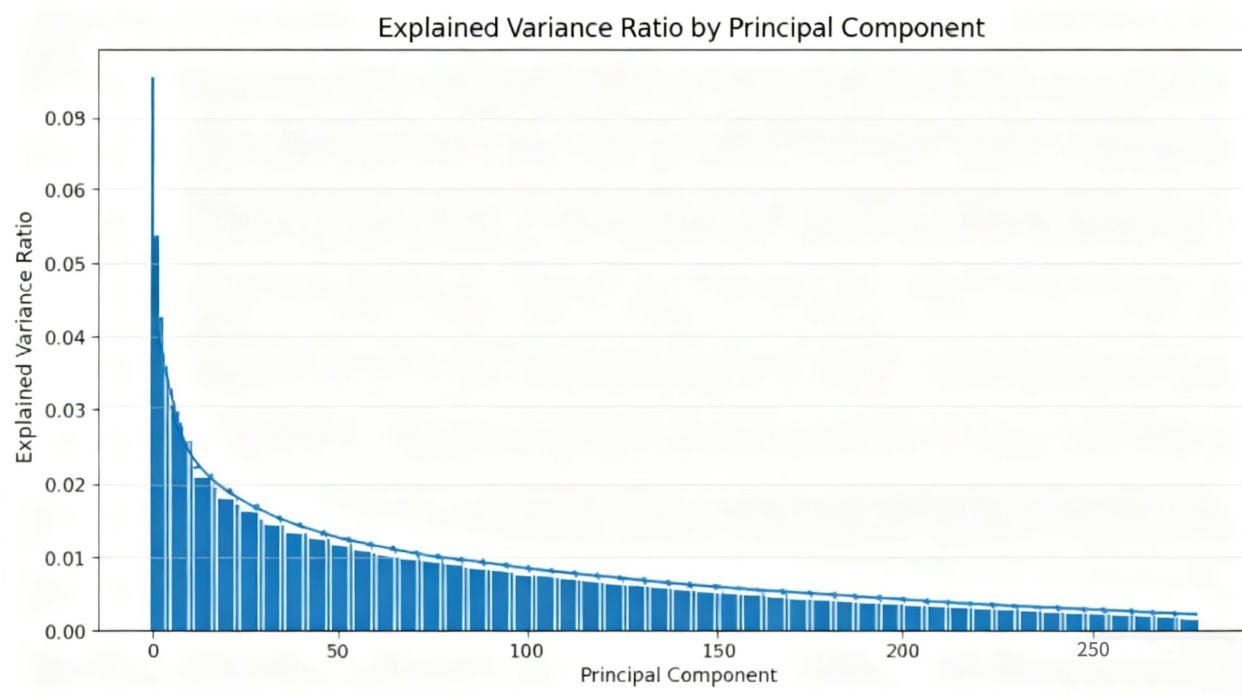


Figure 9. Principal component explained variance ratio.

To evaluate the response time and accuracy of retrieval before and after dimensionality reduction, this paper primarily examines metrics such as retrieval latency and Mean Absolute Error (MAE). The corresponding MAE calculation formula is expressed as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

The results are shown in Table 3. After dimensionality reduction, the retrieval response time was significantly reduced, with the average retrieval latency shortened by 31% on the system's mixed image–text dataset. Although dimensionality reduction caused some loss of information features, leading to a slight decrease in the fine-grained discrimination capability of vector representations and a corresponding increase in MAE, the increase was less than 0.01, indicating a negligible impact. Therefore, this method is considered effective and acceptable in balancing retrieval efficiency and accuracy.

Table 3. Performance Comparison Before and After Dimension Reduction.

Metric	Time Performance (ms)	Accuracy Performance (MAE)
Cosine Similarity	0.5365	0.3799
Cosine Similarity-PCA	0.1692	0.3814

3.4. Hybrid Information Retrieval and Response Generation

3.4.1. Knowledge Graph Retrieval

This study applies an unsupervised entity–relation alignment method to match query keywords with Neo4j graph nodes and relation types. Let the entities in the keywords be denoted as e_i^q . The set of all m entities forms $Q_E = \{e_1^q, e_2^q, \dots, e_m^q\}$. Let the entities in the graph be denoted as e_j^s and the n entities set in the graph be $G_E = \{e_1^s, e_2^s, \dots, e_n^s\}$.

A dimension-reduced text embedding model converts each entity name into semantic representations, followed by calculation of the semantic similarity between them:

$$s_{i,j}^e = \text{CosSim}(v(e_i^q), v(e_j^s)) = \frac{v(e_i^q) \cdot v(e_j^s)}{\|v(e_i^q)\| \|v(e_j^s)\|} \quad (7)$$

Here, v indicates the corresponding semantic vector. The $\text{CosSim}(\cdot)$ symbol denotes the semantic similarity function. Its range is $[-1, 1]$, where values closer to 1 indicate greater semantic similarity.

Subsequently, all candidate entities are sorted in descending order of similarity; the top k_e entities are selected to form the candidate set ($ind_e = \{s_{i,j}^e, j = 1, 2, \dots, n\}$), and those with similarity greater than or equal to the threshold $\tau_E = 0.7$ are selected as final candidates:

$$S_E(e_i^q) = \{e_j^s \in ind_e \mid s_{i,j}^e \geq \tau_E\} \quad (8)$$

Similarly, the relationships extracted from keywords are denoted as r_i^q . The m relationships make up the set of $Q_R = \{r_1^q, r_2^q, \dots, r_m^q\}$. The set of $G_R = \{r_1^s, r_2^s, \dots, r_n^s\}$ is created by matching this set to the n relationships specified in the graph. The following is the definition of the similarity function:

$$s_{i,j}^r = \text{CosSim}(v(r_i^q), v(r_j^s)) = \frac{v(r_i^q) \cdot v(r_j^s)}{\|v(r_i^q)\| \|v(r_j^s)\|} \quad (9)$$

Candidate relationships are sorted by similarity from highest to lowest, and the top k_r relationships are selected to form the candidate set ($ind_r = \{s_{i,j}^r, j = 1, 2, \dots, n\}$). Those with scores exceeding the threshold of $\tau_R = 0.7$ are filtered out as final candidate relation words:

$$S_R(r_i^q) = \{r_j^s \in ind_r \mid s_{i,j}^r \geq \tau_E\} \quad (10)$$

Based on the entity set ($S_E(e_i^q)$) and the relation candidate set ($S_R(r_i^q)$), all possible structural query sets are constructed as follows:

$$S_C = S_E \times S_R = \text{Comb}(S_E, S_R), \quad (11)$$

where $\text{Comb}(\cdot)$ denotes the ordered pairs formed by pairwise combinations of elements from sets S_E and S_R , i.e., the Cartesian product.

For each candidate combination (e, r) , a corresponding Cypher query statement $(p(e, r))$ is constructed and executed in both the Neo4j graph database and the vector database:

$$\text{answer}_p = \text{Execute}(p) \quad (12)$$

If the Cpyher query (p) returns matching entities and relations, the result is accepted as a valid retrieval component.

3.4.2. Vector Database Retrieval

After intent recognition, the system converts key information into a query vector for semantic retrieval. This procedure ensures that query vectors and document vectors in the database stay inside the same vector space by using identical PCA dimensionality reduction and using the bge-zh model for text embedding. For an effective similarity search, the query vector is then entered into the Faiss vector database. This method efficiently assesses the semantic significance between the query and document fragments by using cosine similarity matching for vector retrieval:

$$S_i^{Vec} = \text{CosSim}(v^q, v_i^d) = \frac{v^q \cdot v_i^d}{\|v^q\| \|v_i^d\|} \quad (13)$$

Here, v^q represents the user query vector after model embedding and PCA dimensionality reduction, while v_i^d denotes the fragment vector for document i . The Faiss engine returns the top-k most relevant text fragments based on cosine similarity. The next answer generation module receives these extracted text fragments, as well as the knowledge graph's structured query results. These fragments are provided to the LLM as supplementary context, improving answer completeness and accuracy.

3.4.3. Hybrid-RAG Fusion Ranking Mechanism

The system applies a symmetric “Generate–Retrieve–Generate” framework, combining the knowledge graph and vector retrieval channels for unified reasoning. The relevant pseudocode is outlined in Algorithm 2 as follows.

After both retrieval steps, results are standardized into a unified format and ranked using confidence-weighted fusion:

$$\text{score}_i = \alpha \times KG_i^{\text{score}} + \beta \times S_i^{Vec} + \gamma \times W_i \quad (14)$$

where $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 0.1$ are fusion weights determined through cross-validation on a small-scale validation set, aiming to maximize answer accuracy and information completeness. KG^{score} denotes structural confidence, comprising path length, entity matching, etc., while W represents priority weights assigned based on source relevance. Finally, the bge-reranker-m3 model dynamically adjusts the optimal top-k ranking based on question intent. The selected results are then organized into a structured prompt and passed to the LLM, which integrates them to generate coherent natural language responses. (Detailed calculations of weight values (α , β , and γ) can be found in Appendix B).

Algorithm 2: Dual-Channel Fusion Sorting Pseudocode

Input: Knowledge graph retrieval results: $\mathcal{R}_{KG}$ (entities, relations, attributes, etc.)
Vector database retrieval results: $\mathcal{R}_D$ (relevant document fragments)
Original User Query: Q
Output: The sorted results after integration $\mathcal{R}_{Final}$

```

1 Weighted Merge Sort:
2    $C_i \leftarrow \mathcal{R}_{KG} \cup \mathcal{R}_D$ ;
3   Calculate  $score_i$  ;
4   Generate preliminary rankings  $\mathcal{P}_{pre-rank} \leftarrow s\_d(C_i, score_i)[:K]$ ;

5 Rerank:
6    $reRankScore_i \leftarrow ReRank-Model.predict(Q, item)$ ;

7 Generate the final Ranking:
8    $\mathcal{R}_{Final} \leftarrow s\_d(\mathcal{P}_{pre-rank}, reRankScore)[:N]$ ;
9 return  $\mathcal{R}_{Final}$ ;

```

4. Experimental Design and Evaluation

4.1. Experimental Equipment and Parameters

In this study, BGE-zh-v1.5 was selected due to its strong performance in Chinese dense retrieval tasks and its robustness in handling technical terminology typical of welding specifications. BGE-Reranker-m3 was adopted as a lightweight cross-encoder to refine top-k results while maintaining low computational overhead, making the system deployable in industrial settings. Detailed configurations are provided in Table 4.

Table 4. Equipment configuration.

Equipment	Configuration
GPU	RTX 2080Ti 12G
CUDA	11.4
Python	3.9
Neo4j	5.18.1
LLM	DeepSeek-V3 BGE-Reranker-m3 BGE-zh-v1.5

4.2. Dataset

The dataset was built from the 2024 CCS Materials and Welding Specifications and was cleaned, deduplicated, and restructured for Q&A tasks. To create a high-quality Q&A dataset, the most representative and high-frequency industrial questions were selected. The final dataset contains 47 welding requirement questions and 39 process operation questions.

4.3. Evaluation Metrics

Performance is evaluated using precision, recall, F1 score, and MAP@k, following standard practice in information retrieval and NLP tasks.

1. Precision

Conceptual Q&A performance is evaluated by checking whether the generated answer aligns with the key content of the reference answer. The corresponding precision rate calculation formula is expressed as follows:

$$Precision = \frac{C^w}{T^w} \quad (15)$$

where C^w denotes correct words generated by the model and T^w denotes total words generated by the model.

2. Recall

The model's recall indicates its capacity to recognize and incorporate all pertinent data. The following is the precise formula used to determine recall:

$$Recall = \frac{N^{cw}}{N^{tw}} \quad (16)$$

where N^{cw} denotes the number of correct words generated by the model and N^{tw} as total number of words in the standard answer.

3. F1 Scores

F1-scores combine precision and recall, making them more suitable for scenarios where answers contain partial matches. The following is the precise formula used to determine F1 scores:

$$MAP@k = \frac{1}{M} \sum_{i=1}^M AP@k_i \quad (17)$$

4. MAP@6

Mean average precision at k ($MAP@k$) is a metric used to evaluate the quality of a model's predictions in information retrieval tasks. It computes the average across all samples and assesses the significance of the top k outcomes anticipated by the model.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (18)$$

where M represents the number of welding process issues, and $AP@k_i$ denotes the Average Precision at k for the i -th issue, calculated as follows:

$$AP@k_i = \sum_{i=1}^n P_i \cdot rel_i \quad (19)$$

P_i denotes the accuracy at the i -th position in the prediction list, calculated as follows:

$$P_i = \frac{\text{Number of relevant predictions in top } i}{i} \quad (20)$$

where rel_i is an indicator function: if the i -th prediction is in the actual answer list, $rel_i = 1$; otherwise, $rel_i = 0$. Because graph retrieval may produce multiple keywords or text segments, MAP@6 is used for evaluation.

5. Experimental Results and Discussion

5.1. Ablation Experiment

As clearly shown in Table 5, the experimental results demonstrate that removing the knowledge graph (*w/o* KG) leads to a marked drop in the F1 score from 79.35% to 43.63%, underscoring that the semantic–logical symmetry established by the KG + vector dual-engine architecture is fundamental to system effectiveness.

The knowledge graph not only delivers accurate structured representations of process entities but also provides a logical constraint mechanism that helps resolve semantic ambiguity and supports complex reasoning chains.

When the LoRA module is removed (*w/o* LoRA), MAP@6 declines from 81.48% to 71.44%, and the F1 score drops to 69.42%. These results confirm that LoRA fine tuning considerably enhances logical form generation and improves the model’s grasp of welding-specific terminology and semantics.

While the Hybrid-RAG architecture combined with LoRA provides significant performance improvements, the overall metrics remain slightly below 90% due to inherent variability in the test data. Minor batch differences and qualitative expressions in reference answers may cause small score fluctuations, though these have a negligible impact on the practical accuracy of the system.

Table 5. Ablation experiment.

Model	Precision (%)	Recall (%)	F1 Score (%)	MAP@6 (%)
SWP-Chat	74.74	84.57	79.35	81.48
SWP-Chat <i>w/o</i> KG	35.77	55.92	43.63	-
SWP-Chat <i>w/o</i> LoRA	66.20	53.24	59.02	63.30
SWP-Chat <i>w/o</i> KG, LoRA	34.43	43.56	38.46	-

Note: Bold = best result.

5.2. Comparative Experiments

To assess the system’s domain-specific performance, SWP-Chat was evaluated on the same dataset against several mainstream RAG-enhanced large language models, including Qwen, ChatGLM, and DeepSeek.

Since these baseline models do not contain domain knowledge of manufacturing or welding, a shared vector database was adopted to ensure comparability. On this basis, the proposed model integrates a structured knowledge graph and a dual-channel fusion strategy to verify the contribution of structural reasoning to answer precision and domain relevance.

As summarized in Table 6, SWP-Chat achieved the highest performance across all key metrics. It outperformed existing RAG-based LLM systems in precision, recall, and F1 score, achieving an F1 score of 81.23%. This clearly demonstrates the effectiveness of the proposed Hybrid-PCA-RAG architecture and the symmetric “Generate–Retrieve–Generate” design in tackling complex process-oriented Q&A tasks.

The main advantage of SWP-Chat lies in its ability to represent and reason over multi-step procedural logic rather than relying solely on fragment-level retrieval as in traditional RAG systems. By generating structured logical forms that guide Cypher-based reasoning, the system can answer multi-stage process questions with greater coherence and accuracy.

Although its average response time is approximately 43% longer than conventional RAG systems, this trade-off is acceptable in industrial contexts where traceability and correctness take precedence over speed.

Table 6. Effectiveness evaluation of process operation Q&A.

Metric Model	Precision (%)	Recall (%)	F1 Score (%)	Processing Time (s)
DeepSeek-R1	28.13	32.68	30.23	4.20
ERNIE4.5-8k	29.17	33.42	31.15	3.87
Llama3-70b	28.93	33.07	30.86	3.69
ChatGLM3-32k	28.71	33.51	30.92	3.67
Qwen3-32B	29.40	33.33	31.24	3.88
Ours	78.08	84.65	81.23	5.50

Note: Bold = best result.

As shown in Table 7, the proposed system achieves an F1 score of 77.73% in welding specification question answering, maintaining a leading position among all compared models. This demonstrates the effectiveness of the proposed approach for handling domain-specific queries with complex contextual dependencies.

Welding specifications usually contain multiple parameters, standards, and regulatory clauses stored in structured or semi-structured formats. In such contexts, the knowledge graph plays a central role—its explicit entity–relationship–entity symmetry compensates for the limitations of pure vector retrieval by accurately representing constraints and relationships within the requirements.

Comprehensive analysis reveals a consistent pattern across all models, including ours, where precision is lower than recall. This is not a model defect but, rather, an intentional optimization aligned with user needs in practical engineering scenarios. Unlike process operation queries, which rely on numerical or parameterized data (e.g., current, voltage, or welding speed), specification-related questions often involve unstructured text derived from standards or manuals. Such content is rich in complex semantics, synonyms, and context-dependent expressions, which naturally favor higher recall during retrieval.

Table 7. Effectiveness evaluation of welding specification Q&A.

Metric Model	Precision (%)	Recall (%)	F1-Scores (%)	Processing Time (s)
DeepSeek-R1	38.88	53.07	44.88	4.36
ERNIE4.5-8k	41.69	48.81	44.97	3.03
Llama3-70b	38.64	51.39	44.11	3.22
ChatGLM3-32k	38.36	51.24	43.87	3.16
Qwen3.5-144b	38.90	51.98	44.50	3.35
Ours	71.97	84.50	77.73	5.42

Note: Bold = best result.

5.3. System Usability Analysis

To evaluate the value of this system in practical industrial applications, we further conducted a comparative analysis between SWP-Chat and traditional knowledge graph-based query methods in terms of usability and interaction efficiency.

As shown in Table 8, traditional KG-based queries require users to write Cypher statements, creating a high technical barrier and returning unordered results that are difficult to interpret. Pure vector retrieval (*w/o* KG) often yields redundant or inconsistent answers, while models without LoRA (*w/o* LoRA) fail to generate correct logical chains, leading to incomplete or ambiguous outputs.

SWP-Chat overcomes these limitations through its symmetric Hybrid-RAG design. By automatically generating logical forms and executing Cypher queries, it enables natural language interaction without requiring database expertise. The system’s dual-engine retrieval—semantic vector matching and symbolic reasoning—ensures precise and context-aware responses, even for complex welding queries.

Ultimately, SWP-Chat delivers concise, interpretable, and highly accurate answers in fluent natural language. This balanced input–output symmetry, combined with dual-channel retrieval, markedly enhances industrial reliability, reasoning accuracy, and user experience in ship welding process applications.

Table 8. Model performance and remarks.

Model	Results	Remark
Only-KG	Query statement: MATCH (n:Other{name:"Hydrogen Content Grade H15"})-[r:Measurement Method]-(m); MATCH (n:Other{name:"Hydrogen Content Grade H15"})-[r:Diffusion Hydrogen Content]-(q) RETURN m,q Query Results: [15 cm ³]; [10 cm ³]; [Mercury Method]; [Thermal Conductivity Method]; [Glycerin Method]	This requires knowledge of Cypher, increasing the operational threshold and limiting usability for non-technical users; query results are returned as unordered nodes (content and method), preventing employees from directly associating specific methods with content.
SWP-Chat <i>w/o</i> KG, LoRA	Hydrogen content grade H15 may be determined using the mercury method, thermal conductivity method specified in ISO 3690 [31], or the glycerol method specified in 2.3.6.2. The corresponding diffusion hydrogen measurement procedure and specimen preparation quantity shall be performed according to Table 2.3.6.1 (the thermal conductivity method must meet the minimum collection time requirements at different test temperatures specified in Table 2.3.6.1). The hydrogen content grade is H15.	Without a knowledge graph structure, the system cannot accurately extract key information, leading to redundant and sometimes ambiguous responses. In addition, The system merely returns isolated strings such as "Table 2.3.6.1", stripped of any explanatory context.
SWP-Chat <i>w/o</i> LoRA	The definition of hydrogen content grade H15 cannot be found in the provided information. Applicable measurement methods include the glycerol method. Diffusion hydrogen content refers to the hydrogen content collected through hydrogen measurement tests (such as thermal conductivity, gas chromatography, or carrier gas thermal extraction) under specific	Without LoRA fine tuning, the model exhibits insufficient intent recognition and logical generation capabilities, resulting in recall failures and an inability to generate correct Cypher statements. This leads to poor system reliability and incomplete information retrieval.
SWP-Chat <i>w/o</i> KG	Hydrogen content grade H15 can be measured using the mercury method, thermal conductivity method, or glycerol method. The corresponding diffusion hydrogen content cannot be determined from the information provided.	Vector-only retrieval introduces high semantic noise and lacks precise inference, making it difficult to determine exact hydrogen diffusion values.
SWP-Chat	The measurement methods corresponding to hydrogen content grade H15 include the mercury method, thermal conductivity method, and glycerin method. When using the glycerin method, the diffusion hydrogen content is 10 cm ³ ; for other methods, the diffusion hydrogen content is 15 cm ³ .	The combination of LLM-based logic generation and automatic Cypher execution enables zero-threshold natural interaction, providing concise and accurate answers, regardless of user expertise.

5.4. Significance Testing

To objectively validate the performance enhancement of the proposed Hybrid-PCARAG fusion strategy for question-answering systems, this paper compares the constructed system with a system based on traditional RAG technology. Wilcoxon signed-rank tests are employed to conduct significance analysis on the query-answering results.

5.4.1. Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test is used to compare two paired samples and determine whether the performance differences are statistically significant. For samples $X_1, X_2, \dots, X_i$ obtained by this system and samples $Y_1, Y_2, \dots, Y_i$ obtained by the RAG-only system, the difference (d_i) between each pair of data is calculated as follows:

$$d_i = X_i - Y_i \quad (21)$$

Subsequently, $|d_i|$ are sorted in ascending order, assign each to a rank (R_i), and the rank sums are computed for corresponding positive and negative differences:

$$\text{Rank sum for positive differences: } W^+ = \sum R_i^+ (d_i > 0)$$

$$\text{Rank sum for negative differences: } W^- = \sum R_i^- (d_i < 0)$$

Finally, the relevant test statistic is calculated by taking $W = \min(W^+, W^-)$. For large samples in this system ($n > 30$), the normal approximation is employed in this paper:

$$Z = \frac{W - \mu^W}{\sigma^W}, \quad (22)$$

where p is the effective sample size, corresponding to μ^W, σ^W , and the calculation formula is expressed as follows:

$$\mu^W = \frac{p(p+1)}{4} \quad (23)$$

$$\sigma^W = \sqrt{\frac{n(n+1)(2n+1)}{24}} \quad (24)$$

5.4.2. Results Analysis

Given the high-precision requirements for question-answering performance in practical applications, this study sets the significance level (α) at 5%. The F1 score was calculated for each query–answer pair using both the proposed method and the baseline model, yielding 86 paired samples. The test results were $W = 375, Z = -6.44, p < 0.001$, indicating a statistically significant improvement. The effect size was calculated as $r = Z / \sqrt{N}$, resulting in $r = 0.694$, indicating a large effect.

To control for multiple comparisons, Bonferroni correction was applied to F1, precision, and recall, yielding an adjusted threshold of $\alpha' = \alpha/3 = 0.0167$. All three p-values remained below this level ($p < 0.0167$), confirming that the observed improvement is statistically robust rather than random.

After 1000 bootstrap resamples, the 95% confidence interval for the F1-score gain over the “w/o KG, w/o LoRA” variant was [33.55, 50.35] and [13.68, 28.93] compared with “w/o LoRA”, further supporting the significant benefit of the symmetric generate–retrieve–generate framework integrated with Hybrid-RAG.

As illustrated in Figures 10–12, SWP-Chat achieves the highest values for F1 score, precision, and recall among all systems. Its median lines lie well above those of the ablation variants, and both upper and lower quartiles shift rightward, indicating consistently stronger performance rather than isolated high scores.

Although a few lower outliers are observed, the higher median and narrower interquartile range demonstrate stable and reliable improvements across most queries. Together, the statistical tests and distribution visualizations confirm that the proposed Hybrid-RAG strategy with the symmetric “generate–retrieve–generate” framework delivers significant and consistent gains in welding process question answering.

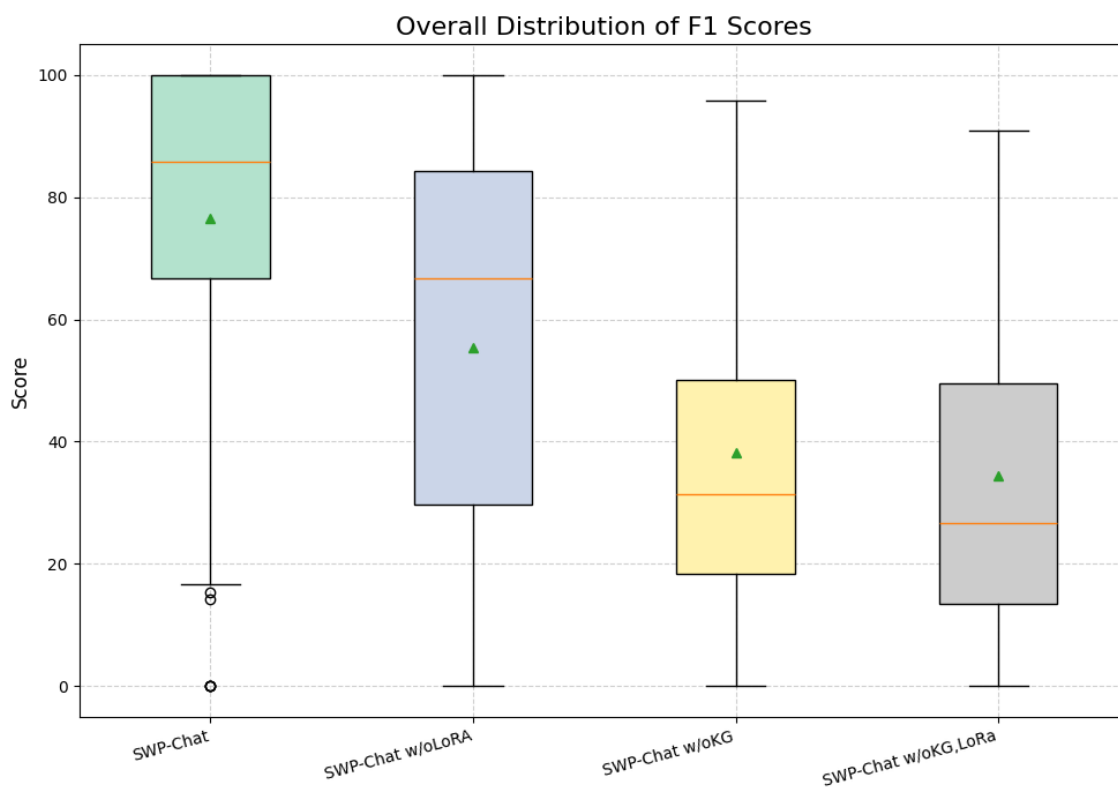


Figure 10. Distribution of F1 scores for the system before and after optimization. The green triangle indicates the mean value of each system, the solid orange line marks the median, and open circles denote individual outliers.

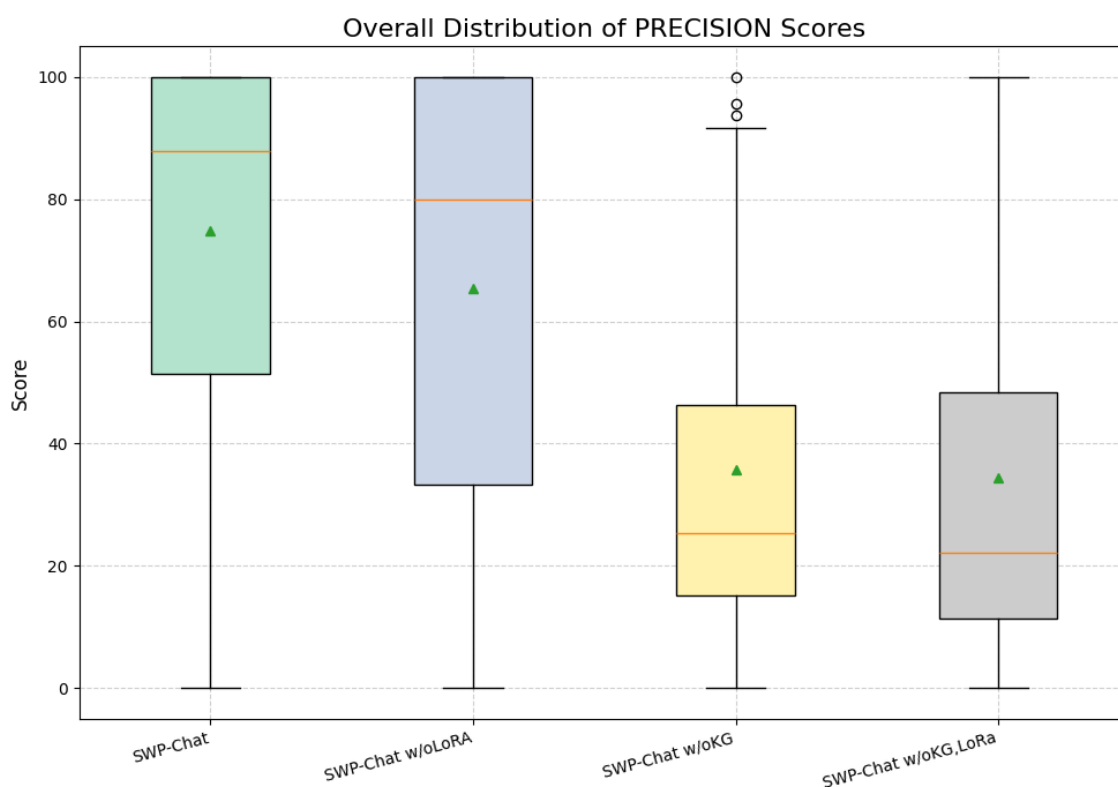


Figure 11. Distribution of precision for the system before and after optimization. The green triangle indicates the mean value of each system, the solid orange line marks the median, and open circles denote individual outliers.

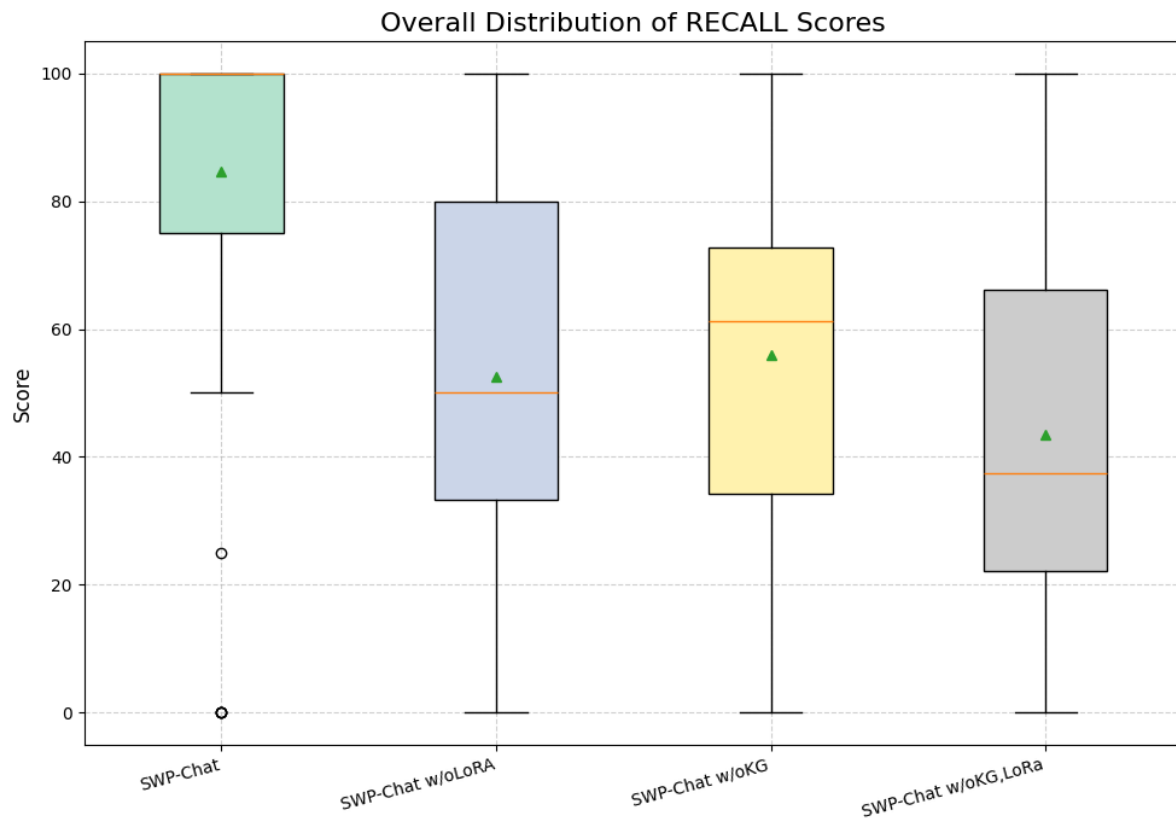


Figure 12. Distribution of recall for the system before and after optimization. The green triangle indicates the mean value of each system, the solid orange line marks the median, and open circles denote individual outliers.

6. Conclusions

This paper proposes SWP-Chat, a symmetric Hybrid-RAG question-answering system designed to support intelligent decision making in ship welding processes. The system significantly outperforms conventional RAG baselines across multiple evaluation metrics.

Unlike typical generate-then-retrieve LLM workflows, SWP-Chat first converts natural language queries into structured logical forms that guide precise Cypher queries over a Neo4j knowledge graph. This generation-first design reduces retrieval noise and prevents loss of structured information when answering complex procedural questions.

The approach fuses the logical rigor of the knowledge graph with the semantic breadth of a vector database, creating a practical semantic–logical balance. Vector retrieval provides high semantic recall, while Neo4j enforces entity–relation constraints and deterministic reasoning. Together, these parallel channels deliver richer, more accurate context for diverse question types.

To improve efficiency, PCA is applied to compress 1024-dimensional embeddings to a lower-dimensional space; this reduces average retrieval latency by 31% while retaining over 95% of variance. Ablation studies show that PCA preserves retrieval accuracy while substantially improving response speed, supporting real-world deployment.

Although the system performs well overall, several limitations remain:

- (1) Knowledge graph maintenance and coverage: Real-time updates rely on manual review, and incomplete KG coverage limits structured reasoning, even when relevant documents exist in the vector database. Expanding multi-source data integration and automated update mechanisms will improve consistency.

- (2) Segmentation granularity of the vector database: Paragraph-based segmentation can compress key procedural details into short references, causing fragments to lose critical context during retrieval. Adaptive merging strategies based on syntax or topic coherence will improve semantic integrity.
- (3) Ambiguity in user queries: Without explicit disambiguation, identical terms may have different meanings in different process contexts, producing partial or incorrect answers. Future work will add LLM-based entity disambiguation and clarification questions to reduce ambiguity.

In summary, the symmetric Hybrid-RAG framework provides interpretable, audit-ready answers suitable for industrial use. Future work will address automatic KG expansion, dynamic chunking strategies, and incremental learning to improve scalability and robustness.

Author Contributions: Conceptualization, S.Y. and L.C.; investigation, B.J.; methodology, L.C.; software, Y.Z.; validation, L.Q.; data curation, X.X.; writing—original draft, L.C. All authors have read and agreed to the published version of the manuscript.

Funding: Zhenjiang Science and Technology Program (grant number: JC2024021).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Shuxia Ye, Yongwei Zhang and Liang Qi were employed by the Jiangsu Shipbuilding and Ocean Engineering Design and Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
RAG	Retrieval-Augmented Generation
KG	Knowledge Graph

Appendix A. LoRA Fine-Tuning Hyperparameter Configuration

To ensure reproducibility and transparency, Table A1 lists the full configuration used for LoRA fine tuning on DeepSeek-V3. Consistent with widely adopted parameter-efficient adaptation practice for Transformer-based LLMs, a small rank and scaling factor were used to balance adaptation capability and computational efficiency. The optimizer, learning rate, and training epochs were selected to ensure stable convergence on the development dataset without introducing overfitting or catastrophic forgetting.

Hyperparameter Settings for LoRA Fine Tuning

Moreover, small-rank LoRA configurations ($r \leq 16$) have shown strong performance in instruction-tuned LLMs of comparable scale while significantly reducing trainable parameters [32,33]. Our empirical validation indicated that lower ranks (e.g., $r = 4$) led to unstable logical form generation, while larger ranks increased GPU memory usage with marginal improvement. Therefore, $r = 8$ was selected as the optimal trade-off between accuracy and efficiency.

Table A1. Hyperparameter settings.

Parameter	Value
Base Model	DeepSeek-V3
LoRA Rank (r)	8
LoRA Scaling Factor (α)	16
LoRA Dropout Rate	0.1
Optimizer	AdamW
Learning Rate	2×10^{-4}
Batch Size	4
Target Modules	q_proj, k_proj, v_proj, o_proj
Training Epochs	3
Max Sequence Length	512

Appendix B. Sensitivity Analysis of Weighting Coefficients

Appendix B.1. Sensitivity Analysis of Weighting Coefficients (α , β , and γ)

Equation (14) assigns three weighting coefficients to balance:

- (1) Logical form confidence derived from knowledge graph retrieval (α);
- (2) Semantic relevance from vector database recall (β);
- (3) LLM-based answer plausibility score (γ).

To determine suitable values, a localized grid search combined with five-fold cross-validation was performed on the development dataset. The search space was constrained by $\alpha + \beta + \gamma = 1$, $\alpha, \beta, \gamma \in 0.1, 0.2, \dots, 0.8$.

For each valid triplet (α, β, γ), the system generated responses and was evaluated using precision, F1 score, and MAP@6. After comparing all feasible combinations, the triplet of (α, β, γ) = (0.6, 0.3, 0.1) achieved the highest averaged performance in all metrics. Therefore, these coefficients were adopted in the final model.

Appendix B.2. Sensitivity Test Design

To examine whether model performance is sensitive to different coefficient settings, five representative weighting configurations were selected:

Table A2. Parameter settings.

Setting ID	α	β	γ
S1	0.8	0.1	0.1
S2	0.7	0.2	0.1
S3	0.6	0.3	0.1
S4	0.5	0.4	0.1
S5	0.4	0.5	0.1

These settings allow for observation of performance changes when shifting emphasis from graph-based reasoning (α) to semantic recall (β) while keeping the plausibility weight (γ) constant. Each setting was evaluated under identical conditions. The following table summarizes the results:

Table A3. Sensitivity test results under different (α, β, γ) settings.

Setting	Precision	Recall	F1-Score	MAP@6
S1 (0.8, 0.1, 0.1)	68.32%	77.12%	72.45%	76.59%
S2 (0.7, 0.2, 0.1)	72.58%	81.02%	76.57%	80.01%
S3 (0.6, 0.3, 0.1)	74.74%	84.57%	79.35%	81.48%
S4 (0.5, 0.4, 0.1)	69.25%	80.36%	74.39%	78.66%
S5 (0.4, 0.5, 0.1)	66.51%	77.62%	71.42%	75.38%

Note: Bold = best result.

Appendix B.3. Sensitivity Result Analysis

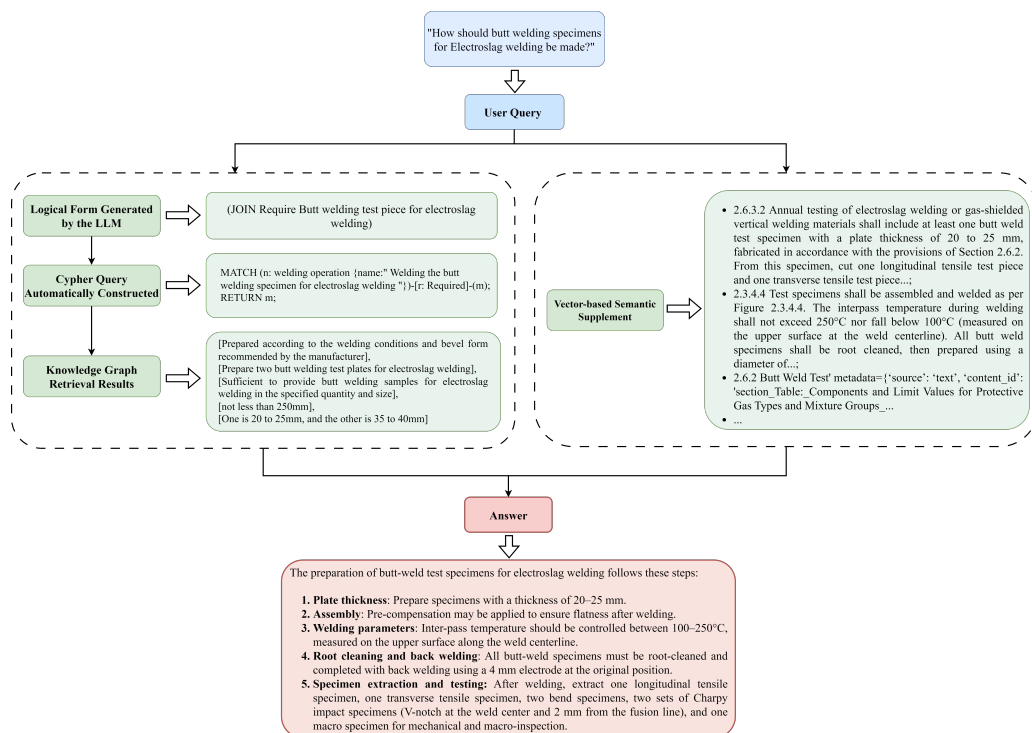
The results indicate:

1. Performance changes significantly as α and β vary, demonstrating that the fusion coefficients directly affect system behavior.
2. Increasing β excessively (favoring semantic recall) introduces retrieval noise, reducing precision and F1 (S4 $\rightarrow$ S5).
3. Overweighting α (favoring strict logical paths) reduces semantic coverage, causing missing context (S1 $\rightarrow$ S2).
4. The (0.6, 0.3, 0.1) configuration provides the best trade-off between (1) structural reasoning from the knowledge graph, (2) semantic richness from vector recall, and (3) output plausibility from the LLM.

Therefore, the final adopted coefficients are not arbitrarily selected; they represent the empirically optimal balance discovered through grid search and cross-validation.

Appendix C. End-to-End Auditability Example

To demonstrate the traceability and auditability of the proposed SWP-Chat system, this appendix provides a complete example diagram covering the entire reasoning chain: user query $\rightarrow$ logical form $\rightarrow$ Cypher translation $\rightarrow$ retrieval of evidence $\rightarrow$ final answer.

**Figure A1.** End-to-End system framework.

References

1. The State Council of the People's Republic of China. China Remains World's Top Manufacturer for 15 Consecutive Years: Official. *English News Release—Official Statistics*, 9 July 2025. Available online: https://english.www.gov.cn/archive/statistics/202507/09/content_WS686e064dc6d0868f4e8f3fe1.html (accessed on 12 November 2025).
2. Iatrou, C.P.; Ketzel, L.; Graube, M.; Häfner, M.; Urbas, L. Design classification of aggregating systems in intelligent information system architectures. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; Volume 1, pp. 745–752.
3. Wang, B.; Wang, G.; Huang, J.; You, J.; Leskovec, J.; Kuo, C.-C. Inductive learning on commonsense knowledge graph completion. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
4. Zeng, Z.; Cheng, Q.; Si, Y. Logical rule-based knowledge graph reasoning: A comprehensive survey. *Mathematics* **2023**, *11*, 4486. [\[CrossRef\]](#)
5. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, H.; Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv* **2023**, arXiv:2312.10997v2.
6. Knollmeyer, S.; Caymazer, O.; Grossmann, D. Document GraphRAG: Knowledge graph enhanced retrieval augmented generation for document question answering within the manufacturing domain. *Electronics* **2025**, *14*, 2102. [\[CrossRef\]](#)
7. Shaftee, S. Integration of product and manufacturing design: A systematic literature review. *Procedia CIRP* **2023**, *121*, 19–24. [\[CrossRef\]](#)
8. Yahya, M.; Ali, A.; Mehmood, Q.; Yang, L.; Breslin, J.G.; Ali, M.I. A benchmark dataset with Knowledge Graph generation for Industry 4.0 production lines. *Semant. Web* **2024**, *15*, 461–479. [\[CrossRef\]](#)
9. Meckler, S. Procedure model for building knowledge graphs for industry applications. *arXiv* **2024**, arXiv:2409.13425.
10. Guo, L.; Li, X.; Yan, F.; Lu, Y.; Shen, W. A method for constructing a machining knowledge graph using an improved transformer. *Expert Syst. Appl.* **2024**, *237*, 121448. [\[CrossRef\]](#)
11. Liu, X.; Wang, H. Knowledge Graph Construction and Decision Support Towards Transformer Fault Maintenance. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; IEEE: Dalian, China, 2021; pp. 661–666.
12. Zhang, Y.; Zhang, S.; Huang, R.; Huang, B.; Liang, J.; Zhang, H.; Wang, Z. Combining deep learning with knowledge graph for macro process planning. *Comput. Ind.* **2022**, *140*, 103668. [\[CrossRef\]](#)
13. Dong, J.; Jing, X.; Lu, X.; Liu, J.; Li, H.; Cao, X.; Du, C.; Li, J.; Li, L. Process knowledge graph modeling techniques and application methods for ship heterogeneous models. *Sci. Rep.* **2022**, *12*, 2911. [\[CrossRef\]](#)
14. Zhou, B.; Bao, J.; Chen, Z.; Liu, Y. KGAssembly: Knowledge graph-driven assembly process generation and evaluation for complex components. *Int. J. Comput. Integr. Manuf.* **2022**, *35*, 1151–1171. [\[CrossRef\]](#)
15. Shi, X.; Tian, X.; Gu, J.; Yang, F.; Ma, L.; Chen, Y.; Su, T. Knowledge graph-based assembly resource knowledge reuse towards complex product assembly process. *Sustainability* **2022**, *14*, 15541. [\[CrossRef\]](#)
16. Lin, L.; Zhang, S.; Fu, S.; Liu, Y. FD-LLM: Large language model for fault diagnosis of complex equipment. *Adv. Eng. Inform.* **2025**, *65*, 103208. [\[CrossRef\]](#)
17. Meng, S.; Wang, Y.; Yang, C.-F.; Peng, N.; Chang, K.-W. LLM-A*: Large language model enhanced incremental heuristic search on path planning. *arXiv* **2024**, arXiv:2407.02511.
18. Liu, Y.; Zhou, Y.; Liu, Y.; Xu, Z.; He, Y. Intelligent fault diagnosis for CNC through the integration of large language models and domain knowledge graphs. *Engineering* **2025**, *53*, 311–322. [\[CrossRef\]](#)
19. Werheid, J.; Melnychuk, O.; Zhou, H.; Huber, M.; Rippe, C.; Joosten, D.; Keskin, Z.; Wittstamm, M.; Subramani, S.; Drescher, B.; et al. Designing an LLM-based copilot for manufacturing equipment selection. *arXiv* **2024**, arXiv:2412.13774. [\[CrossRef\]](#)
20. Jiang, X.; Qiu, R.; Xu, Y.; Zhu, Y.; Zhang, R.; Fang, Y.; Xu, C.; Zhao, J.; Wang, Y. RAGraph: A general retrieval-augmented graph learning framework. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 29948–29985.
21. Wang, J.; Fu, J.; Wang, R.; Song, L.; Bian, J. PIKE-RAG: SPecialized Knowledge and Rationale augmented generation. *arXiv* **2025**, arXiv:2501.11551.
22. Barron, R.C.; Grantcharov, V.; Wanna, S.; Eren, M.E.; Bhattarai, M.; Solovyev, N.; Tompkins, G.; Nicholas, C.; Rasmussen, K.Ø.; Matuszek, C.; et al. Domain-specific retrieval-augmented generation using vector stores, knowledge graphs, and tensor factorization. In Proceedings of the 2024 International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 18–20 December 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1669–1676.
23. Opoku, D.O.; Sheng, M.; Zhang, Y. DO-RAG: A domain-specific QA framework using knowledge graph-enhanced retrieval-augmented generation. *arXiv* **2025**, arXiv:2505.17058.
24. Russell-Gilbert, A. RAAD-LLM: Adaptive Anomaly Detection Using LLMs and RAG Integration. Ph.D. Dissertation, Mississippi State University, Starkville, MS, USA, 2025.

25. Sarmah, B.; Mehta, D.; Hall, B.; Rao, R.; Patel, S.; Pasquali, S. HybridRAG: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In Proceedings of the ICAIF'24: 5th ACM International Conference on AI in Finance Brooklyn NY USA, 14–17 November 2024; ACM: New York, NY, USA, 2024; pp. 608–616.
26. Tram, M.H.B. Efficacy of GraphRAG (Knowledge Graph-enhanced RAG) Beyond Keyword Matching for Information Retrieval. Ph.D. Dissertation, University of Jyväskylä, Jyväskylä, Finland, 2025.
27. Zhai, S.; Ji, H.; Zhang, K.; Wu, Y.; Ma, Z. Approximate Query for Industrial Fault Knowledge Graph Based on Vector Index. *Int. J. Softw. Eng. Knowl. Eng.* **2025**, *35*, 525–545. [[CrossRef](#)]
28. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic parsing on Freebase from question-answer pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, DC, USA, 18–21 October 2013; Association for Computational Linguistics: Seattle, WA, USA, 2013; pp. 1533–1544.
29. Maswadi, K.; Ghani, N.A.; Hamid, S.; Rasheed, M.B. Human activity classification using decision tree and naïve Bayes classifiers. *Multimed. Tools Appl.* **2021**, *80*, 21709–21726. [[CrossRef](#)]
30. Guan, K.; Du, L.; Yang, X. Relationship extraction and processing for knowledge graph of welding manufacturing. *IEEE Access* **2022**, *10*, 103089–103098. [[CrossRef](#)]
31. China Classification Society. Rules for Materials and Welding (2024) released by China Classification Society (CCS). *Ship Stand. Eng.* **2024**, *57*, 2. (In Chinese with English abstract)
32. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, L.; Chen, W.; Chen, Y. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685.
33. Ben Zaken, E.; Goldberg, Y.; Ravfogel, S. BitFit: Simple Parameter-Efficient Fine-Tuning for Transformer-Based Masked Language-Models. *arXiv* **2022**, arXiv:2106.10199.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.