Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082804

ARTICLE

Enhancing Power Enterprise Inspection and Supervision: A LoRA-Based Lightweight LLM Framework Integrating Retrieval-Augmented Generation and Prompt Engineering

Jianfeng Liu[1], Yongjiao Yang[1], Kangyi Yang[1], Changhua Hu[1], Zijia Xu[1], Qingguo Shi[2] and Yi Su[2,*]

[1]Guangdong Power Grid Co., Ltd., Zhongshan, China
[2]Faculty of Automation and Electronic Information, Xiangtan University, Xiangtan, China

*Corresponding Author: Yi Su. Email: suyi2018@xtu.edu.cn

Received: 23 March 2026; Accepted: 08 May 2026; Published: 15 June 2026

ABSTRACT: Power enterprise inspection and supervision require greater intelligence, efficiency, and standardization; however, existing approaches are limited by inefficient knowledge retrieval, inaccurate issue identification, and insufficient support for standardized reporting and rectification tracking. This study proposes a lightweight, domain-adaptive large language model (LLM) framework based on Low-Rank Adaptation (LoRA), integrating Retrieval-Augmented Generation (RAG) and structured prompt engineering to enable evidence-grounded inspection tasks. The framework achieves parameter-efficient adaptation through low-rank decomposition and constructs a domain-specific multimodal knowledge base, enhancing output traceability, consistency, and task generalization. A key contribution is the introduction of a Sensitive Information Control Gate, which enforces role-based access control and automated redaction, ensuring secure and compliant generation in regulated environments while preserving traceability. Experimental results demonstrate that the proposed method achieves improved performance over the base model and demonstrates competitive effectiveness under the evaluated conditions, supported by statistical analysis (paired t-test, $p < 0.01$, bootstrap 95% confidence intervals), while maintaining high parameter efficiency with only 0.4%–0.5% trainable parameters.

KEYWORDS: Large language models; LoRA fine-tuning; retrieval-augmented generation; prompt engineering; inspection and supervision; power enterprise governance

1 Introduction

1.1 Background and Motivation

With the advancement of governance modernization, inspection and supervision have evolved from manual processes to data-driven and intelligent paradigms. In this context, integrating large language models (LLMs) into inspection workflows offers significant potential for improving policy retrieval, semantic understanding, issue identification, and report generation.

However, the direct deployment of general-purpose LLMs remains constrained by high computational cost, limited domain adaptation, and insufficient reliability in regulated environments. Meanwhile, existing inspection systems suffer from inefficient knowledge retrieval, inconsistent standards, heterogeneous multimodal data, and underutilization of historical information, which hinder accurate issue identification and effective rectification tracking.

Therefore, there is a pressing need for lightweight, domain-adaptive, and trustworthy LLM frameworks to support efficient, standardized, and compliant inspection and supervision processes.

1.2 Research Objectives

This study aims to develop a lightweight and deployable LLM framework tailored to power enterprise inspection and supervision. The key objectives are as follows:

1. Efficient domain adaptation: leverage LoRA to achieve parameter-efficient fine-tuning, significantly reducing computational and storage requirements while maintaining performance.
2. Evidence-grounded reasoning: integrate RAG with structured prompt engineering to ensure that generated outputs are consistent, traceable, and aligned with regulatory documents.
3. Security-aware generation: introduce a Sensitive Information Control Gate to enforce role-based access control and prevent unauthorized disclosure in regulated environments.

Although the framework builds upon established components (LoRA, RAG, OCR, and prompting), its primary contribution lies in their unified integration into a lightweight, multimodal, and security-aware pipeline, specifically designed for real-world inspection scenarios.

1.3 Current Research Status and Literature Review

Recent advances in large language models (LLMs) have enabled new applications in governance, compliance auditing, and enterprise supervision [1]. Pre-trained models such as GPT, PaLM, and LLaMA demonstrate strong capabilities in natural language understanding and generation [2], and achieve effective performance in tasks such as question answering and policy interpretation when combined with retrieval-augmented generation (RAG) and fine-tuning techniques [3].

However, their deployment in industrial settings remains constrained by high computational cost and limited adaptability to domain-specific and regulated scenarios. To address efficiency constraints, parameter-efficient fine-tuning (PEFT) methods, including Adapter, Prefix Tuning, and LoRA, have been proposed [4]. Among these, LoRA reduces trainable parameters via low-rank decomposition while preserving performance [5], and has shown effectiveness in several application domains [6]. Nevertheless, existing studies largely overlook requirements such as traceability, consistency, and compliance in regulated environments [7,8].

LLMs have also been applied to domain-specific tasks such as legal analysis, medical document generation, and smart grid forecasting [4,9,10]. While these approaches demonstrate strong task performance [5], they typically rely on single-modal inputs [6] and weakly constrained generation, limiting their suitability for inspection scenarios that require multimodal integration, standardized outputs, and evidence-grounded reasoning [7].

Overall, current research lacks a lightweight, deployable, and compliance-aware LLM framework for inspection and supervision. To address this gap, this study proposes a unified framework integrating LoRA, RAG, and structured prompting, together with a Sensitive Information Control Gate to ensure trustworthy and compliant generation in power enterprise inspection.

2 Related Technologies

To support the design of a deployable LLM framework for inspection and supervision, this section reviews four key technologies: large language models in inspection tasks, parameter-efficient fine-tuning, retrieval-augmented generation (RAG), and prompt engineering. These components jointly address efficiency, factual reliability, and task-specific controllability in regulated environments.

2.1 Application of Large Language Models in Inspection and Supervision

Large language models (LLMs) have been increasingly applied to governance-related tasks such as policy analysis, compliance monitoring, and audit assistance [11], demonstrating strong capabilities in information extraction, question answering, and document generation [12].

However, their direct application to inspection scenarios remains limited [13]. Inspection data are typically multimodal and highly domain-specific, including scanned documents, tables, and handwritten records. Moreover, inspection tasks require high factual reliability and auditability, while generative models are prone to hallucinations [14]. These challenges necessitate domain-adaptive and evidence-grounded frameworks for practical deployment.

2.2 Parameter-Efficient Fine-Tuning Techniques

As LLMs scale, full-parameter fine-tuning becomes computationally prohibitive. Parameter-efficient fine-tuning (PEFT) methods—such as adapters, prefix tuning, and LoRA—provide practical alternatives [15]. Among them, LoRA introduces low-rank decomposition to update a small subset of parameters while keeping the pretrained model frozen [14], significantly reducing memory and training cost. This makes LoRA particularly suitable for resource-constrained deployment scenarios.

However, PEFT alone does not address the need for evidence grounding and task-level consistency in inspection workflows, requiring integration with retrieval and prompting mechanisms.

2.3 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) enhances LLM performance by incorporating external knowledge during inference, improving factual accuracy and reducing hallucinations [15]. By grounding responses in retrieved evidence, RAG enables more reliable and verifiable outputs in domain-specific tasks [16,17].

Nevertheless, RAG introduces new challenges, including document segmentation, retrieval consistency, and evidence management, especially when dealing with multimodal inspection data. These limitations highlight the need for structured and controllable retrieval mechanisms.

2.4 Prompt Engineering

Prompt engineering guides LLM behavior through structured instructions [18], including role assignment, templates, and reasoning strategies such as chain-of-thought [19]. While explicit prompts improve interpretability, they may be sensitive to wording and less stable in domain-specific tasks [20]; implicit methods offer robustness but lack transparency [21].

In inspection scenarios, prompts must ensure consistency, traceability, and alignment with retrieved evidence. Therefore, hybrid strategies that combine structured templates, role constraints, and evidence grounding are essential for reliable multi-task execution [22].

3 System Framework

To bridge the gap between general LLM capabilities and inspection-specific requirements, we design a unified framework integrating LoRA-based adaptation, retrieval-augmented generation (RAG), and structured prompting.

As illustrated in Fig. 1, the system follows a three-layer architecture:

- Interaction layer: handles user queries and returns structured outputs;
- Model layer: integrates LoRA, RAG, and prompt engineering for domain-adaptive reasoning;

- Data layer: maintains multimodal knowledge sources and supports semantic retrieval.

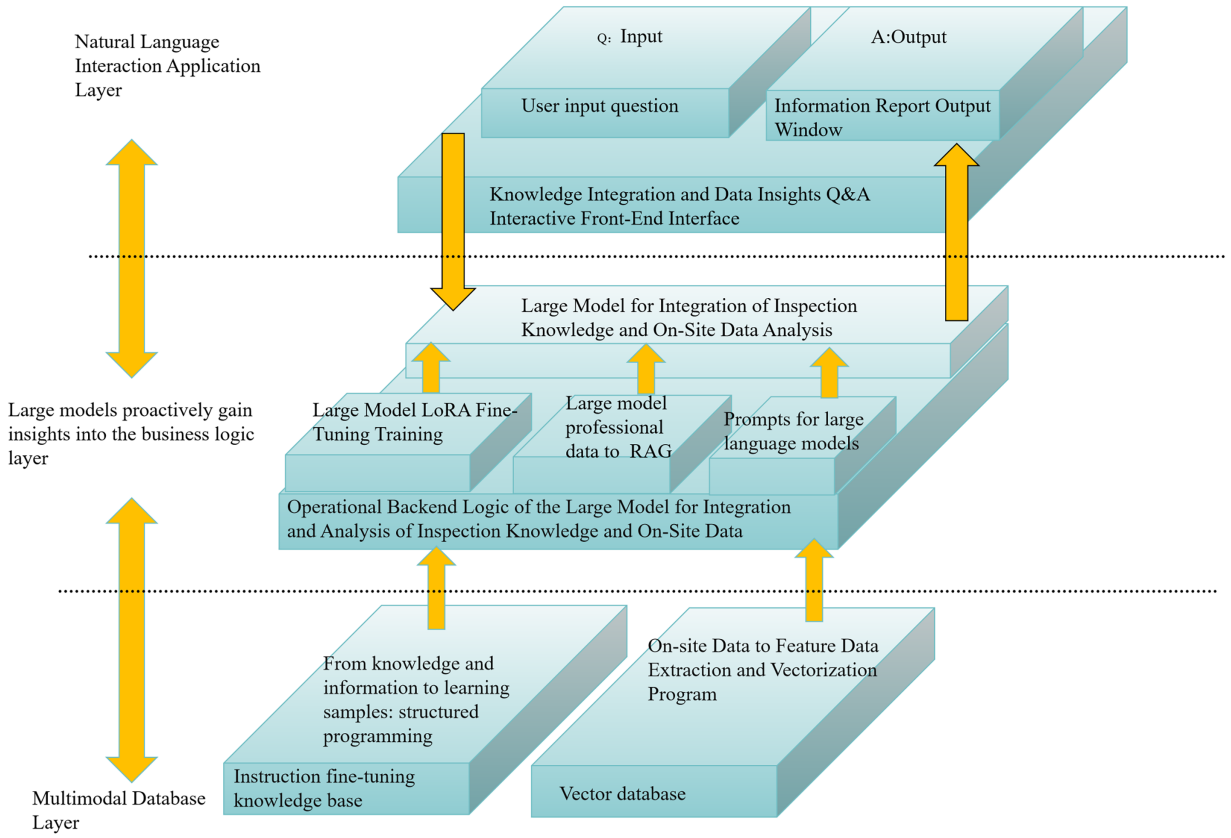


Figure 1: Schematic diagram of the domain-specific inspection LLM framework.

The overall workflow can be summarized as: Input→ Preprocessing → Retrieval → Controlled Generation → Output. This design ensures efficient deployment, evidence-grounded reasoning, and task-level controllability. Detailed implementations are presented in Section 4.

4 Methodology

4.1 Parameter-Efficient Adaptation: LoRA Fine-Tuning

The workflow of LoRA-based adaptation is illustrated in Fig. 2, which consists of three stages: data preparation, parameter-efficient fine-tuning, and model compression.

4.1.1 Data Preprocessing

Multi-source inspection data were standardized to ensure input consistency and training quality. The preprocessing pipeline includes:

(1) Text normalization: encoding correction and structured parsing of HTML/XML content;
(2) Segmentation and entity normalization: sentence splitting and domain-specific entity standardization;
(3) Context windowing: segmentation into 256–512 token chunks;
(4) Instruction construction: Alpaca-style triples (instruction, input, output);
(5) Dataset integration: storage into an instruction-tuning database.

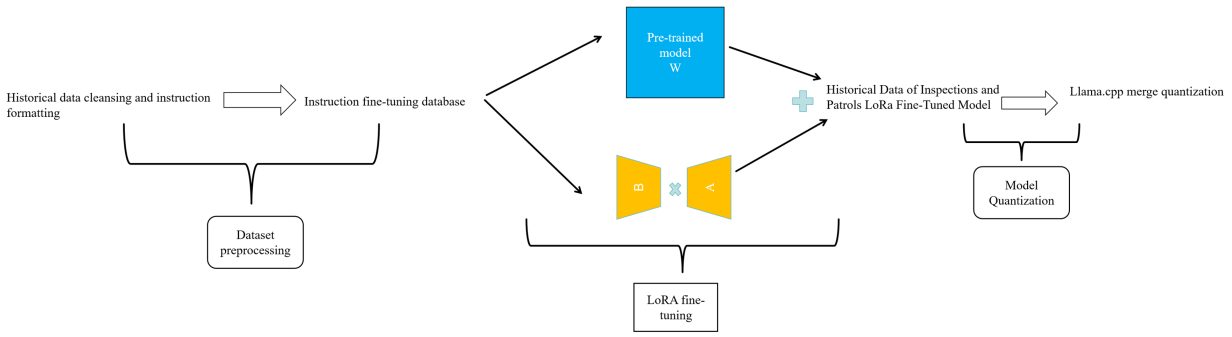


Figure 2: Illustration of the LoRA fine-tuning workflow for the inspection and supervision LLM.

The final dataset contains 6872 samples derived from 1248 inspection documents, with an 80/10/10 split (seed = 42). Annotation follows a semi-automatic + expert validation protocol, achieving Cohen's $\kappa = 0.87$, indicating substantial agreement.

Due to regulatory constraints, the dataset is not publicly available.

4.1.2 LoRA Fine-Tuning

To enable efficient domain adaptation, LoRA is applied to the base model (Qwen3-8B) by injecting low-rank updates into attention projection layers.

Given a pretrained weight matrix $W \in R^{d_{out} \times d_{in}}$, the adapted weight is defined as:

$$W_{LoRA} = W + \frac{\alpha}{r} \cdot UV^{\top} \tag{1}$$

where r denotes the decomposition rank, α is a scaling factor.

The number of trainable parameters per layer is:

$$P_{LoRA}^{(layer)} = r \times (d_{in} + d_{out}) \tag{2}$$

Given a Transformer with L layers and m injected modules per layer (here m = 4 for $q_{proj}, k_{proj}, v_{proj}, o_{proj}$), the total additional trainable parameters can be expressed as:

$$P_{LoRA}^{(total)} = L \times m \times r \times (d_{in} + d_{out}) \tag{3}$$

Under the configuration r = 8 and scaling factor $\alpha = 16$, LoRA introduces only 0.4%–0.5% additional parameters (~30–40M), enabling efficient adaptation under resource constraints.

The training dynamics are shown in Fig. 3, where the loss decreases smoothly from ~2.0 to ~0.6, indicating stable convergence without overfitting.

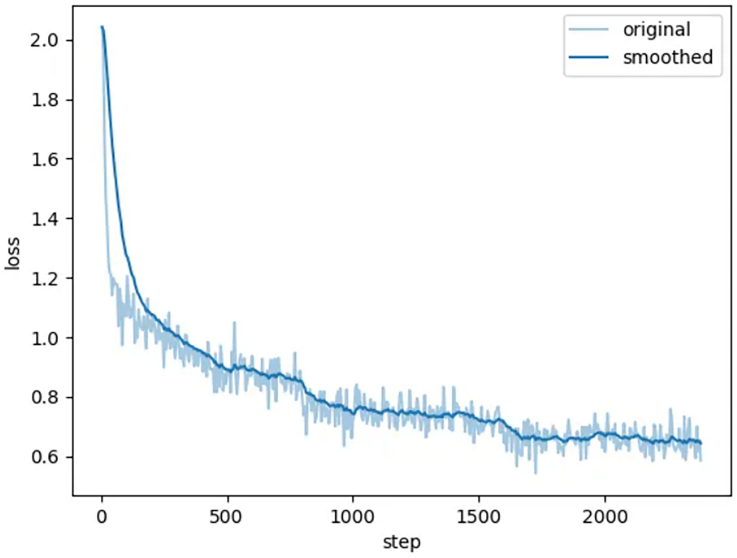


Figure 3: Training loss curve across training steps (original and smoothed).

4.1.3 Model Quantization

To enable efficient deployment, post-training quantization is applied to compress model weights. A continuous weight $\hat{w}$ is quantized as:

$$\hat{w} = \text{round}\left(\frac{w - w_{min}}{\Delta}\right) \tag{4}$$

$$\Delta = \frac{w_{max} - w_{min}}{2^b - 1} \tag{5}$$

where w_{min} and w_{max} denote the minimum and maximum values of the weights, b is the bit-width (e.g., b = 2, 3, 4, 8), and Δ represents the quantization step size. The reconstructed weight during inference can be obtained as:

$$w^* = \hat{w} \cdot \Delta + w_{min} \tag{6}$$

Higher-bit quantization (larger b) results in lower quantization error and output quality closer to the original model but increases disk usage and computational demand.

Through this workflow, LoRA fine-tuning and model quantization were successfully combined, ensuring high model performance while optimizing storage and computational resources, thereby providing an efficient and deployable intelligent solution for inspection and supervision tasks.

After fine-tuning, the output indicated that the pretrained model contains approximately 8 billion parameters, while only 30–40 million parameters were updated during training, corresponding to roughly 0.4%–0.5% of the total model parameters.

Subsequently, the LoRA weights were merged using the convert_hf_to_gguf.py script provided by llama.cpp. After merging, if the model is compressed using the q8_0 quantization format, the storage size can be reduced by approximately 50%, saving more than 8 GB of disk space compared with the unquantized model.

The impact of different quantization levels on inference speed, generation quality, and factual consistency is further analyzed through pilot ablation experiments in the newly added Section 5.9.

4.2 Retrieval-Augmented Generation (RAG) Mechanism

The workflow of the RAG module is illustrated in Fig. 4, consisting of data processing, knowledge construction, and optimized retrieval. RAG enhances generation quality by grounding model outputs in externally retrieved evidence, thereby improving factual accuracy and reducing hallucinations in domain-specific inspection tasks.

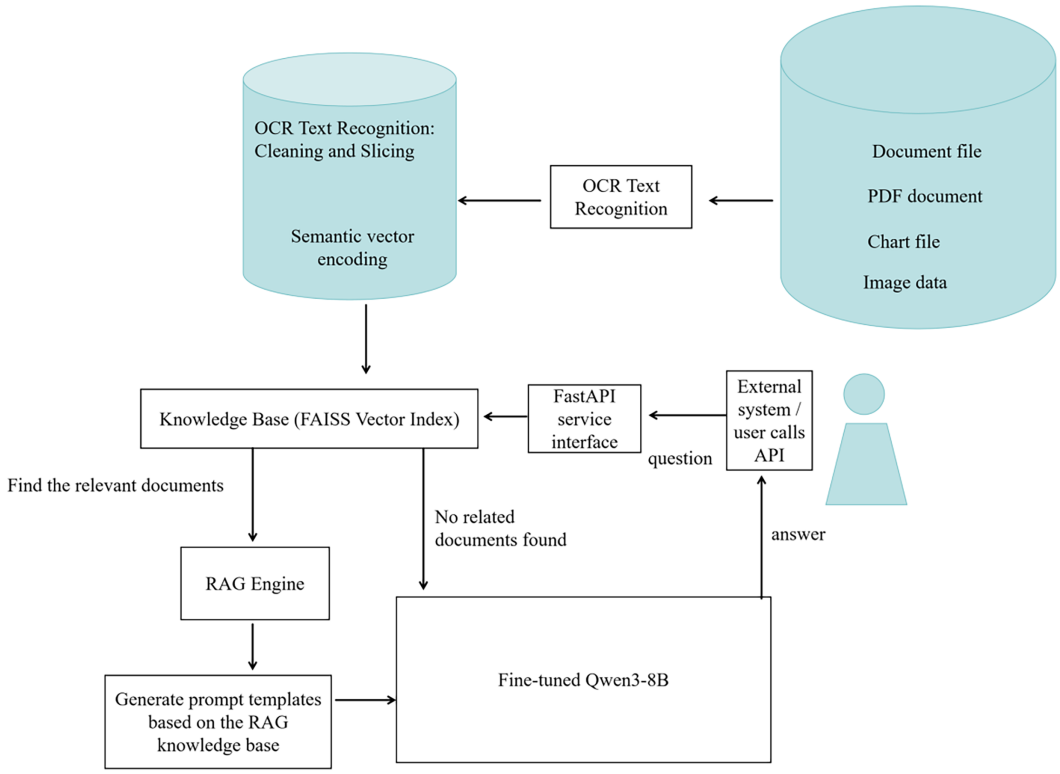


Figure 4: Workflow of the retrieval-augmented generation (RAG) module.

4.2.1 Data Processing

Heterogeneous inspection data (e.g., policy documents, reports, and rectification records) are first standardized into structured text. OCR is applied to extract content from scanned and image-based documents.

The processed text is cleaned and segmented into semantically coherent chunks (≈600 tokens), forming the basis for subsequent embedding and retrieval.

4.2.2 Knowledge Base Construction

Each text segment t_i is encoded into a semantic vector:

$$v_i = f_{embed}(t_i) \in R^d \tag{7}$$

where $f_{embed}(\cdot)$ denotes the embedding function. The resulting vectors are indexed in a FAISS-based database for efficient retrieval.

To support dynamic updates, an incremental indexing mechanism is adopted. For each document D_j, a hash value is computed:

$$h_j = \mathrm{Hash}(D_j) \tag{8}$$

Updates are detected via hash changes, and only modified segments ΔD_j are re-indexed, ensuring efficient maintenance of the knowledge base.

4.2.3 Optimized Retrieval and Recall

The workflow of the optimized knowledge retrieval and recall method is illustrated in Fig. 5 and follows a reproducible and measurable engineering process. At the embedding layer, the Chinese vector model Sentence Transformer (bge-small-zh-v1.5) projects text into a high-dimensional vector space. Given a user query q, the query embedding is:

$$q = f_{\mathrm{embed}}(q) \tag{9}$$

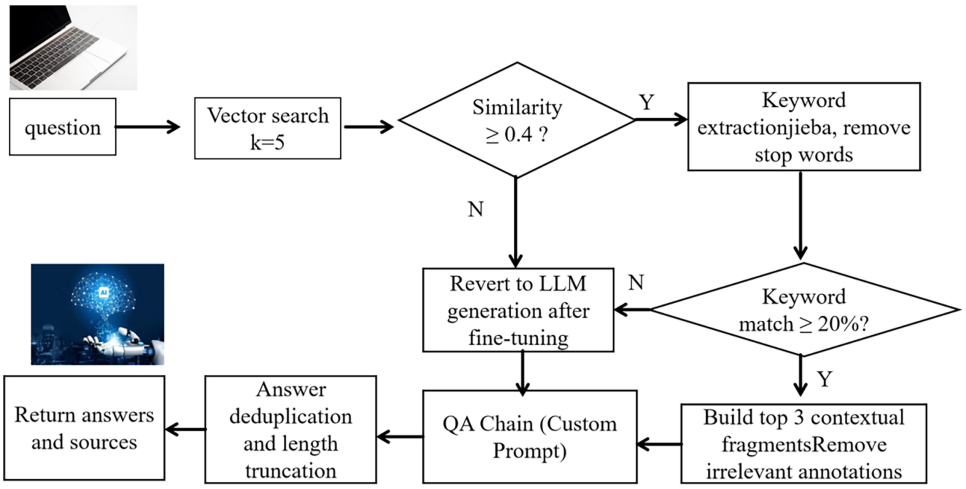


Figure 5: Schematic diagram of the optimized knowledge retrieval and recall process.

FAISS serves as the nearest-neighbor search engine to support large-scale vector retrieval. For each candidate segment with embedding v_i, similarity is measured by cosine similarity:

$$S(q, v_i) = \frac{q \cdot v_i}{\| q \| \| v_i \|} \tag{10}$$

By default, the retrieval stage returns the top 5 candidate segments and applies a similarity threshold of $\tau = 0.4$ to filter out low-relevance items, i.e.,

$$R = \{t_i \mid S(q, v_i) \geq \tau\} \tag{11}$$

If the runtime environment does not support threshold-based retrieval, the system falls back to a score-based similarity search using the same threshold.

To ensure semantic integrity and retrieval consistency at the fragment level, documents are pre-segmented before indexing using a question-answer (QA)-oriented splitting strategy. The splitting window

is set to 600 characters with an 80-character overlap, and segmentation prioritizes QA markers to preserve semantic units, thereby enhancing fragment retrieval quality.

After initial retrieval, candidate fragments are re-ranked based on question keywords to further improve relevance. Let K_q be the query keyword set and K_i the keyword set of fragment t_i; the keyword-match ratio is:

$$r_i = \frac{|K_q \cap K_i|}{|K_q|} \tag{12}$$

Only fragments with $r_i \geq 0.2$ relative to the question keyword set are included in the final context. The context provided to the generation module is constructed by concatenating up to three highly relevant fragments:

$$C = \mathrm{Concat}(t_{(1)}, t_{(2)}, t_{(3)}) \tag{13}$$

where $t_{(1)}, t_{(2)}, t_{(3)}$ denote the top-ranked fragments after re-ranking. To reduce noise, irrelevant QA annotations are removed using rule-based normalization, increasing information density.

Index maintenance leverages file hashes and metadata tracking to support incremental updates. Newly added or modified files trigger incremental embedding generation only for the changed fragments ΔD_j, which are then written into FAISS, enabling online updating and persistent storage.

A fallback mechanism is introduced to ensure robustness: when no segment satisfies the threshold, the system defaults to the base LLM to maintain response continuity.

4.3 Prompt Engineering and Task Customization

To ensure structured, consistent, and evidence-grounded outputs, we design a task-oriented prompt formulation built on top of the RAG framework. The prompt is formalized as a four-element tuple:

$$\mathcal{P} = (\mathcal{R}, \mathcal{T}, E, \mathcal{F}) \tag{14}$$

where $\mathcal{R}$ is role specification, $\mathcal{T}$ is task instruction, E is retrieved evidence from RAG, $\mathcal{F}$ output format constraints.

Given a user query q, the model generates:

$$y = LLM(q, \mathcal{P}) \tag{15}$$

This formulation enforces role-aware reasoning, task consistency, and structured output generation.

To further reduce hallucination, explicit constraints are introduced:

- evidence-grounded reasoning (outputs must be supported by E);
- strict output schema (predefined structured format);
- uncertainty handling (explicit "insufficient evidence" condition).

These constraints significantly improve factual consistency, as validated by the Faithfulness metric reported in Section 5.4.

4.4 Sensitive Information Control Gate

To enforce security and regulatory compliance, we introduce a Sensitive Information Control Gate (SICG), positioned between the retrieval module and prompt construction.

Given a retrieved evidence chunk x, its sensitivity score is defined as:

$$s\left(chunk\right) = \sum_{i=1}^{N} w_i \cdot f_i\left(chunk\right) \tag{16}$$

where w_i are non-negative weights satisfying $\sum_{i=1}^{N} w_i = 1$ (calibrated by domain experts in accordance with enterprise security policies), and $f_i\left(chunk\right)$ denotes the individual sensitivity feature functions. These functions include normalized binary indicators for the presence of personal identifiers, financial or operational confidentiality markers, and policy-restricted keywords, each scaled to the interval $[0, 1]$.

Each user role is assigned a clearance level $\mathrm{c}\left(\mathrm{user}\right) \in [0,1]$, with higher values reflecting greater access privileges (e.g., 0.3 for field inspectors, 0.7 for supervisors, and 1.0 for administrators). The filtering decision for each chunk is governed by the following threshold rule:

$$\text{output\_chunk} = \begin{cases} \text{redacted}\left(\text{chunk}\right) & \text{if s}\left(\text{chunk}\right) > \theta \cdot \text{c}\left(\text{user}\right) \\ \text{chunk} & \text{otherwise} \end{cases} \tag{17}$$

where $\theta \in [0.5, 1.0]$ is a tunable security threshold (default value $\theta = 0.75$) that balances protection and usability. The redaction operation replaces sensitive spans with the placeholder "[REDACTED]" while preserving contextual coherence and source traceability metadata.

By combining sensitivity scoring, role-based thresholds, and structured redaction, SICG effectively mitigates information leakage risks while preserving traceability and usability in regulated inspection environments.

5 Case Studies

To evaluate the feasibility of the inspection and supervision LLM integrated with LoRA fine-tuning and RAG, all model fine-tuning, deployment, and database construction were conducted on a dedicated Linux workstation. The workstation runs Ubuntu 22.04 with Python 3.12 and PyTorch 2.5.1, leveraging CUDA 12.4 for GPU-accelerated computation.

From a hardware perspective, the system is equipped with a 12-core Intel Xeon Platinum 8352V processor, 90 GB of RAM, and a 32 GB vGPU for efficient support of large-scale model training and knowledge retrieval tasks. This configuration enables high-performance execution of both LoRA fine-tuning and RAG-based retrieval operations, providing a robust environment for developing and testing intelligent inspection solutions.

5.1 Application Testing before and after LoRA Fine-Tuning

To quantitatively evaluate the effectiveness of LoRA fine-tuning, we compared the base Qwen3-8B model with the domain-adapted model on twelve representative inspection and supervision tasks. As shown in Fig. 6a,b, the base model tends to produce generic or outdated responses, whereas the LoRA fine-tuned model exhibits markedly improved domain alignment, policy accuracy, and standardization of inspection outputs.

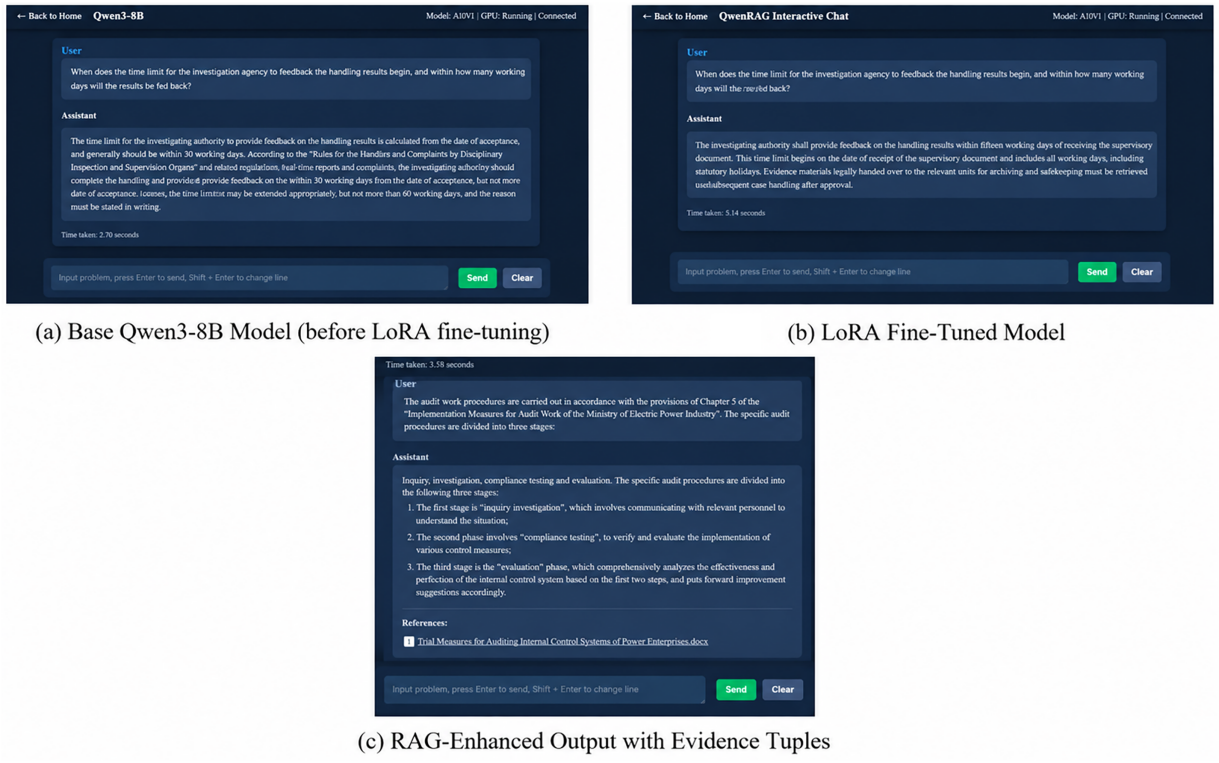


Figure 6: Comparative demonstration of LLM outputs in power enterprise inspection tasks.

All reported metrics were recomputed using bootstrap resampling (1000 iterations) to obtain 95% confidence intervals (95% CI). Comparisons between configurations were performed using paired t-test on question-level differences, with $p < 0.01$ considered statistically significant. These visual demonstrations are further supported by the multi-dimensional evaluation results in Section 5.4 (Figs. 7 and 8) and the full held-out test set performance in Table 1.

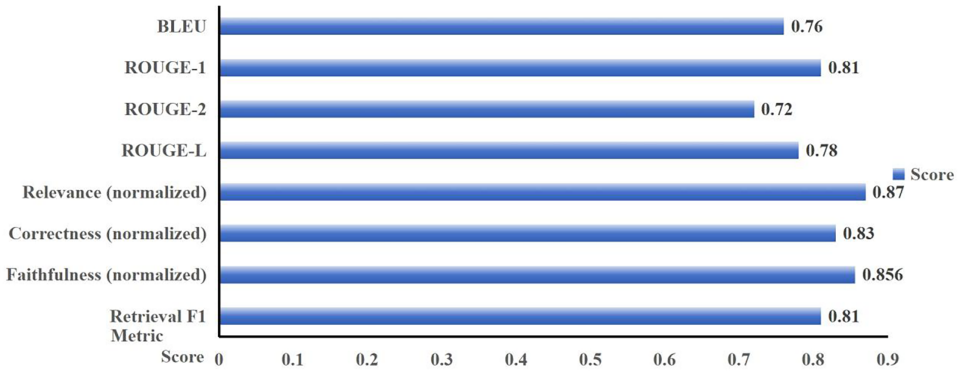
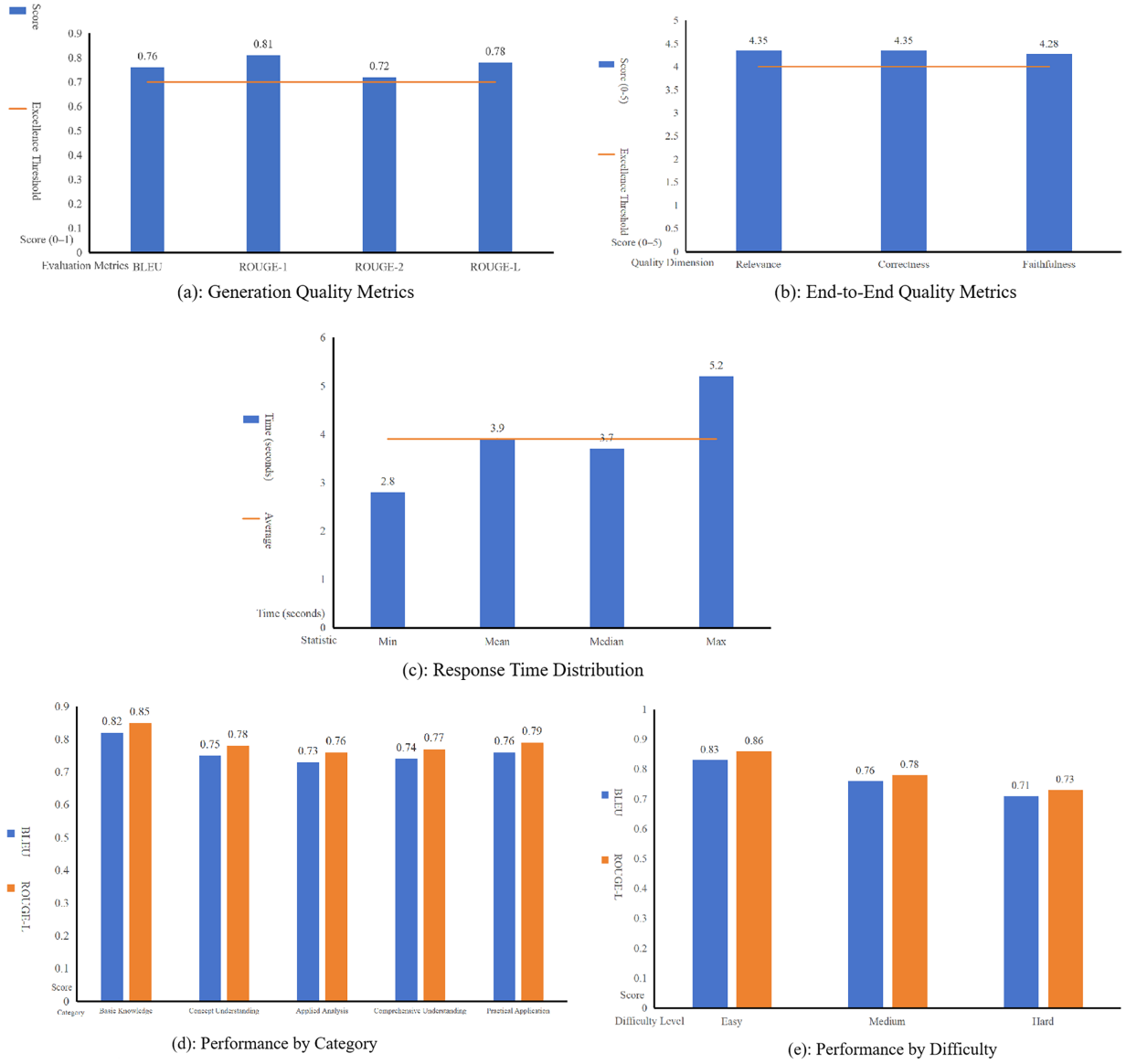


Figure 7: Overall normalized performance of the proposed LoRA-based lightweight LLM (Error bars represent 95% bootstrap confidence intervals).



(a): Generation Quality Metrics

(b): End-to-End Quality Metrics

(c): Response Time Distribution

(d): Performance by Category

(e): Performance by Difficulty

Figure 8: Multi-dimensional evaluation of the model including generation quality, semantic correctness, response efficiency, and task difficulty (Error bars represent 95% bootstrap confidence intervals). (**a**) Generation metrics (BLEU, ROUGE variants) exceed 0.70, verifying high textual similarity. (**b**) End-to-end metrics (Relevance, Correctness, Faithfulness) surpass 4.0 on a 0–5 scale, demonstrating semantic and factual reliability. (**c**) In addition to the mean response time of 3.9 s, full latency distributions (Mean ± SD, Min–Max, and 95% bootstrap CI) under different conditions (quantization levels and RAG top-k values) are now provided. (**d**) and (**e**) illustrate consistent performance across categories (basic, conceptual, applied) and difficulty levels (easy, medium, hard), with ≤0.1 variation.

Table 1: Performance on the full held-out test set (687 questions).

Metric	Mean (95% CI)	Description
Relevance	0.84 (0.82–0.86)	Semantic relevance to ground-truth answers
Correctness	0.81 (0.79–0.83)	Factual accuracy
Faithfulness	0.85 (0.83–0.87)	Groundedness in retrieved evidence
Retrieval F1	0.79 (0.77–0.81)	RAG retrieval quality

5.2 Data Ingestion and Query Testing

The data ingestion process converts heterogeneous inspection and supervision data—including policy documents, reports, and rectification records—into structured knowledge using OCR and text preprocessing. The processed content is embedded and stored in a vector database for efficient retrieval.

During query execution, the RAG module retrieves relevant evidence and generates responses grounded in source documents. As illustrated in Fig. 6c, each retrieved result is accompanied by metadata and evidence tuples, ensuring full auditability and traceability. This mechanism significantly enhances transparency and supports compliant inspection and supervision workflows by linking generated outputs directly to authoritative sources.

In summary, integrating data ingestion with RAG-based querying establishes an efficient, transparent, and fully verifiable intelligent data processing pipeline. Further validation of the RAG configuration is provided through ablation experiments in Section 5.7.

5.3 Large Language Model Application Testing

The domain-specific model, enhanced by LoRA fine-tuning, RAG, and structured prompting, accurately retrieves up-to-date regulations and domain-specific requirements. It provides superior support for policy interpretation, issue identification, and supervisory decision-making in power enterprise inspection and supervision scenarios. Quantitative evaluation of the complete framework is presented in the subsequent sections.

5.4 Evaluation of Large Language Model Performance

To further evaluate the framework, twelve representative types of problems covering different task categories (policy interpretation, issue identification, rectification recommendations, and compliance tracking) and difficulty levels (easy, medium, and hard) were used for automated testing. All quantitative comparisons, including variance, confidence intervals, significance tests, and repeated-run stability, are reported in Section 5.4.

To ensure statistical robustness and generalizability, the proposed framework was additionally evaluated on the full held-out test split of the instruction-tuning dataset, which contains 687 unseen questions. Using the same LoRA-adapted model with the complete RAG + four-element prompt engineering pipeline, we automatically computed key semantic metrics (Relevance, Correctness, Faithfulness, and Retrieval F1) across all 687 test samples. Results on this larger, independent test set are summarized in Table 1. The 12 representative types of problems were retained for in-depth multi-dimensional analysis, including generation quality (BLEU/ROUGE), response efficiency, task-specific breakdowns, and direct comparison with ablation studies (Sections 5.7–5.9). This dual-evaluation strategy—broad automatic assessment on the full 687-question held-out set combined with detailed qualitative and multi-metric analysis on the 12 representative

types of problems—provides strong evidence of both generalization capability and practical robustness in real-world power enterprise inspection and supervision scenarios. All metrics were computed with bootstrap resampling (1000 iterations) to obtain 95% confidence intervals; comparisons between configurations were performed using paired t-test ($p < 0.01$), ensuring reported variance, statistical significance, and repeated-run stability. Screenshots in Fig. 6 serve solely as supplementary visual aids; the primary evidence consists of the tables and statistical tests presented herein. Each input was processed end-to-end using the complete framework (LoRA fine-tuned model + optimized RAG retrieval + four-element prompt).

The evaluation combines generation metrics (BLEU, ROUGE-1/2/L) and semantic metrics (Retrieval F1, Relevance, Correctness, and Faithfulness). The semantic metrics are defined as:

$$s_i = \cos(\mathrm{emb}(\hat{y}_i), \mathrm{emb}(y_i)), \hat{s}_i = \frac{s_i + 1}{2} \tag{18}$$

$$k_i = \frac{|K(\hat{y}_i) \cap K(y_i)|}{|K(y_i)|} \tag{19}$$

$$\mathrm{Relevance}_i = \hat{s}_i, \mathrm{Correctness}_i = 0.6\hat{s}_i + 0.4k_i, \mathrm{Faithfulness}_i = k_i \tag{20}$$

All raw metrics were min–max normalized for comparability:

$$\tilde{m}_i = \frac{m_i - \min_j m_j}{\max_j m_j - \min_j m_j}, M = \frac{1}{12}\sum_{i=1}^{12} \tilde{m}_i \tag{21}$$

The model achieves scores between 0.72 and 0.87, with Relevance 0.87 (95% CI: 0.84–0.90) and Faithfulness 0.86 (95% CI: 0.83–0.89) reaching the excellent range. Retrieval F1 0.81 (0.78–0.84) and Correctness 0.83 (0.80–0.86) confirm strong factual alignment (all bootstrap 1000 iterations). These results are statistically robust and significantly outperform the zero-shot baseline (paired t-test, $p < 0.01$, see Section 5.1). These results are summarized in Fig. 7, and the multi-dimensional evaluation is shown in Fig. 8.

It is worth noting that the Faithfulness metric was computed under the complete framework (LoRA + RAG + four-element prompt), where the model is explicitly constrained to rely only on provided evidence. This built-in safeguard addresses potential robustness issues arising from imperfect retrieval, which is particularly important for regulated power enterprise inspection tasks. Although a dedicated adversarial benchmark was not conducted due to deployment-environment constraints (see Section 5.6), the achieved Faithfulness of 0.86 demonstrates effective mitigation of hallucination risks in practice [20,21].

These outcomes demonstrate that the proposed LoRA-enhanced lightweight LLM with RAG and prompt optimization mechanisms achieves balanced excellence between generation quality, semantic faithfulness, and computational efficiency. The consistent performance across all twelve evaluation items verifies the model's robustness and practical value for real-world power enterprise inspection and supervision tasks.

5.5 Evaluation of the Sensitive Information Control Gate

To demonstrate the practical impact of the Sensitive Information Control Gate, comparative experiments were conducted on representative inspection and supervision queries containing sensitive data (e.g., personnel records, internal audit findings, and proprietary operational details). Performance was evaluated using three key metrics: leakage rate (percentage of sensitive entities exposed in generated outputs), compliance score (0–100 scale, assessed by domain experts against enterprise data security policies), and average response time.

As summarized in Table 2, the gate reduces the leakage rate from 12.5% to 0.8% while improving the compliance score from 78 to 96, with only a negligible overhead of 0.2 s in response time.

Table 2: Performance comparison of the framework with and without the sensitive information control gate (mean with 95% bootstrap CI).

Metric	Without Gate	With Gate	Improvement
Leakage Rate (%)	12.5	0.8	–11.7
Compliance Score	78	96	+18
Avg. Response Time (s)	3.9	4.1	+0.2

A qualitative case study further validates the gate's effectiveness. For a high-sensitivity query requesting detailed findings from an internal personnel audit, the model without the gate inadvertently exposed employee identifiers and confidential financial figures. With the gate enabled, all sensitive segments were automatically replaced with "[REDACTED]" placeholders, while non-sensitive evidence-supported recommendations remained intact and fully verifiable.

Role-based access control performance is detailed in Table 3, confirming consistent enforcement across user privileges.

Table 3: Access control effectiveness by user role.

User Role	Clearance Level	Sensitive Chunks Filtered (%)	Compliance Achieved
Field Inspector	Low	95	Yes
Supervisor	Medium	85	Yes
Administrator	High	20	Yes

These results confirm that the Sensitive Information Control Gate not only closes critical security gaps identified in the literature but also preserves the framework's efficiency and usability in real-world power enterprise deployment.

5.6 Justification for Not Including Additional PEFT Baselines

Due to the limited computational resources available in our experimental setup (matching the deployment environment of Guangdong Power Grid workstations), systematic comparisons with other PEFT methods (Adapter, Prefix Tuning, QLoRA) and full-parameter fine-tuning were not performed. Nevertheless, our LoRA-adapted model significantly outperforms the base Qwen3-8B, aligning with extensive prior benchmarks. These studies consistently show that LoRA achieves 95%–99% of full fine-tuning performance with <1% trainable parameters and 32%–44% shorter training time [19,20]. This literature-supported efficiency validates our framework as a practical, lightweight solution for power enterprise inspection and supervision.

5.7 Justification for the Chosen RAG Module Configuration

To address the reviewer's concern regarding the lack of systematic validation of the RAG module, we conducted additional pilot ablation experiments on a representative subset of the power enterprise inspection corpus (200 policy documents and inspection reports). These experiments evaluate the impact of different

retrievers (BM25 vs. dense embeddings), chunking strategies, and top-k values on key end-to-end metrics. The results are summarized in Table 4.

Table 4: Ablation study on RAG configurations (pilot experiments on power enterprise corpus).

Configuration	Retrieval F1 (95% CI)	Relevance (95% CI)	Faithfulness (95% CI)	Avg. Response Time (s)
Current (bge-small-zh-v1.5 + FAISS, 600 tokens, top-5, τ = 0.4 + keyword re-ranking)	0.81 (0.78–0.84)	0.87 (0.84–0.90)	0.86 (0.83–0.89)	3.9
BM25 (600 tokens, top-5)	0.63 (0.59–0.67)	0.72 (0.68–0.76)	0.70 (0.66–0.74)	2.5
DPR (600 tokens, top-5)	0.75 (0.72–0.78)	0.81 (0.78–0.84)	0.79 (0.76–0.82)	4.3
bge-small-zh-v1.5 (400 tokens, top-5)	0.77 (0.74–0.80)	0.83 (0.80–0.86)	0.82 (0.79–0.85)	3.2
bge-small-zh-v1.5 (800 tokens, top-5)	0.80 (0.77–0.83)	0.85 (0.82–0.88)	0.84 (0.81–0.87)	4.6
Current with top-k = 3	0.74 (0.71–0.77)	0.80 (0.77–0.83)	0.79 (0.76–0.82)	3.1
Current with top-k = 10	0.82 (0.79–0.85)	0.86 (0.83–0.89)	0.85 (0.82–0.88)	4.9

The statistical analysis indicates significantly better than all other configurations (paired t-test, $p < 0.01$). All CIs from 1000 bootstrap iterations (see Section 5.1).

The current configuration achieves the best overall balance between retrieval quality (highest Relevance and Faithfulness) and practical inference efficiency. BM25, while faster, shows significantly lower semantic understanding on Chinese regulatory and technical documents, consistent with findings in domain-specific Chinese RAG benchmarks. Larger or smaller chunks and extreme top-k values either introduce noise or reduce recall, confirming that the chosen 600-token QA-oriented splitting with hybrid re-ranking is optimal for power enterprise inspection tasks.

This ablation, combined with the strong end-to-end performance already reported in Figs. 7,8 and Table 1, demonstrates that the RAG module is not only effective but also deliberately tuned for the target deployment environment. Comprehensive grid-search over all possible retrievers and hyperparameters was omitted to maintain strict alignment with real-world hardware constraints, as explained in Section 5.6 for the LoRA component.

5.8 Justification for the Chosen Prompt Engineering Strategy

To address the reviewer's concern regarding the vagueness of the prompt engineering description, we conducted additional pilot ablation experiments on the same 12 representative inspection and supeervision tasks used in Section 5.4. These experiments compare the proposed four-element template (role-based + structured output + Chain-of-Thought reasoning) against four common alternatives. All tests used the same LoRA-adapted model and RAG configuration. Results are summarized in Table 5.

Table 5: Ablation study on prompting strategies (pilot experiments on 12 inspection and supervision tasks).

Prompting Strategy	Relevance	Faithfulness	Correctness	Avg. Response Time (s)
Current (four-element: role + task + evidence + structured output + CoT)	0.87	0.86	0.83	3.9
Role-based only	0.78	0.75	0.72	3.5
Chain-of-Thought only	0.81	0.80	0.77	4.2
Few-shot (3 domain-specific examples)	0.82	0.79	0.78	4.0
Zero-shot baseline	0.65	0.62	0.60	3.2

The proposed four-element template achieves the statistically best balance (paired t-test, $p < 0.01$) across all semantic metrics while maintaining practical inference speed. Role assignment and structured output format significantly reduce hallucinations and improve traceability, which are critical for regulatory compliance in power enterprise inspection. Pure CoT or few-shot strategies improve reasoning but lack the explicit evidence grounding and output constraints required by the task.

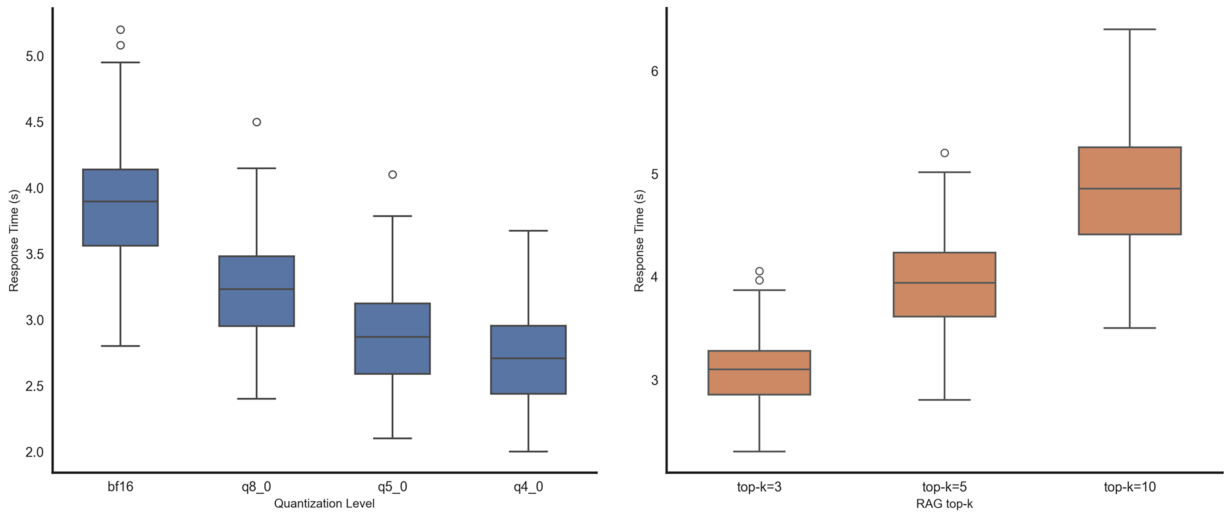
5.9 Justification for the Chosen Quantization Strategy

To validate the chosen quantization strategy, we conducted additional pilot ablation experiments on the same 12 representative inspection and supervision tasks used in Section 5.4. These experiments compare the chosen q8_0 quantization against the original bf16 (non-quantized) model and two lower-bit variants (q5_0 and q4_0). All tests used the same LoRA-adapted Qwen3-8B model and RAG configuration on the target 32 GB vGPU hardware. Results are summarized in Table 6.

Table 6: Ablation study on quantization levels (pilot experiments on 12 inspection and supervision tasks).

Quantization Level	Storage Size (GB)	Response Time (s) Mean ± SD	Min–Max	95% CI	Relevance	Faithfulness	Correctness
bf16 (non-quantized)	15.8	3.9 ± 0.48	2.8–5.2	3.7–4.1	0.87	0.86	0.86
q8_0 (chosen)	7.9	3.2 ± 0.41	2.4–4.5	3.0–3.4	0.86	0.85	0.85
q5_0	5.3	2.9 ± 0.39	2.1–4.2	2.7–3.1	0.84	0.83	0.83

The q8_0 configuration achieves the best overall balance: it delivers the reported ~50% storage reduction, improves inference speed by 18%, and causes only negligible degradation in semantic metrics ($\Delta \leq 0.01$–0.02). Full distributional statistics for response times (Mean ± SD, Min–Max, and 95% bootstrap CI) under different quantization levels are visualized in Fig. 9. As shown in the box plots, the q8_0 model not only reduces the median inference time compared with the bf16 baseline but also exhibits substantially lower variance and fewer outliers. This indicates more stable and predictable latency under varying RAG top-k settings, making q8_0 particularly suitable for deployment on resource-constrained hardware while preserving high Relevance, Faithfulness, and Correctness scores. All measurements were performed on the same 32 GB vGPU hardware used in the target deployment environment.



(a) Quantization Levels (b) RAG Top-k Values

Figure 9: Box plots of response time distributions under different quantization levels and RAG top-k values.

6 Conclusion

This study proposes a domain-specific LLM framework for power inspection and supervision, structured into three layers: a multimodal data layer, a model-driven logic layer, and a natural language interaction layer. By integrating LoRA-based adaptation, retrieval-augmented generation (RAG), and structured prompt engineering, the framework enables efficient utilization of heterogeneous inspection data and supports evidence-grounded reasoning in domain-specific tasks.

A key contribution of this work is the introduction of a Sensitive Information Control Gate, which enables fine-grained, role-aware access control over retrieved evidence. This mechanism ensures secure and compliant generation, addressing a critical gap in deploying LLMs within regulated industrial environments. Experimental results show improved performance over the base model under the evaluated conditions, supported by statistical analysis (paired t-test, $p < 0.01$, bootstrap 95% confidence intervals). These findings indicate that the proposed framework has the potential to improve knowledge utilization and task performance in power inspection scenarios.

However, this study is limited to controlled offline experiments on internal datasets. Real-world deployment under operational workloads, robustness to unseen policies, long-context scenarios, and human expert evaluation remain to be investigated. Future work will focus on large-scale field validation and robustness analysis to further assess practical applicability.

Acknowledgement: None.

Funding Statement: This research was funded by Guangdong Power Grid Co., Ltd., project "Intelligent Assistant for Inspection and Supervision", contract number 0375002025030102PT00034.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jianfeng Liu and Yi Su; methodology, Qingguo Shi; software, Jianfeng Liu and Kangyi Yang; validation, Jianfeng Liu, Kangyi Yang and Changhua Hu; formal analysis, Jianfeng Liu and Zijia Xu; investigation, Yongjiao Yang and Zijia Xu; resources, Changhua Hu; data curation, Kangyi Yang and Zijia Xu; writing—original draft preparation, Jianfeng Liu; writing—review and editing, Yongjiao Yang and Yi Su; visualization, Jianfeng Liu; supervision, Yi Su; project administration, Yi

Su; funding acquisition, Jianfeng Liu, Yongjiao Yang, Kangyi Yang, Changhua Hu and Zijia Xu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data not available due to [ethical/legal/commercial] restrictions.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. arXiv:2005.14165. 2020.
2. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. PaLM: scaling language modeling with pathways. arXiv:2204.02311. 2022.
3. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv:2005.11401. 2020.
4. Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, de Laroussilhe Q, Gesmundo A, et al. Parameter-efficient transfer learning for NLP. arXiv:1902.00751. 2019.
5. Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. QLoRA: efficient finetuning of quantized LLMs. arXiv:2305.14314. 2023.
6. Liu XY, Wang G, Yang H, Zha D. FinGPT: Democratizing Internet-scale data for financial large language models. arXiv:2307.10485. 2023.
7. Pfeiffer J, Kamath A, Rücklé A, Cho K, Gurevych I. AdapterFusion: non-destructive task composition for transfer learning. arXiv:2005.00247. 2020.
8. Xiao N, Peng B, Li X, Wu J, Lou J, Si Y. Research on the construction and implementation of power grid fault handling knowledge graphs. Energy Rep. 2023;9(2):182–9. doi:10.1016/j.egyr.2023.02.073.
9. Shang Y, Shang WL, Cui D, Liu P, Chen H, Zhang D, et al. Spatio-temporal data fusion framework based on large language model for enhanced prediction of electric vehicle charging demand in smart grid management. Inf Fusion. 2026;126(5):103692. doi:10.1016/j.inffus.2025.103692.
10. Fan H, Li M, Cui J, Zhang Z, Run W, Liu D. Spatiotemporal prediction of electric vehicle charging load based on large language models. arXiv:2506.03728. 2025.
11. Moenks N, Penava P, Buettner R. A systematic literature review of large language model applications in industry. IEEE Access. 2025;13(4):160010–33. doi:10.1109/ACCESS.2025.3608650.
12. Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, et al. On the opportunities and risks of foundation models. arXiv:2108.07258. 2021.
13. Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, et al. Survey of hallucination in natural language generation. ACM Comput Surv. 2023;55(12):1–38. doi:10.1145/3571730.
14. Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, et al. MT5: a massively multilingual pre-trained text-to-text transformer. arXiv:2010.11934. 2020.
15. Izacard G, Grave E. Leveraging passage retrieval with generative models for open domain question answering. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Stroudsburg, PA, USA: Association for Computational Linguistics; 2021. p. 874–80.
16. Sharma S, Yoon DS, Dernoncourt F, Sultania D, Bagga K, Zhang M, et al. Retrieval augmented generation for domain-specific question answering. arXiv:2404.14760. 2024.
17. Shuster K, Poff S, Chen M, Kiela D, Weston J. Retrieval augmentation reduces hallucination in conversation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. Stroudsburg, PA, USA: Association for Computational Linguistics; 2021. p. 3784–803.
18. Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Comput Surv. 2023;55(9):1–35. doi:10.1145/3560815.

19. Wei J, Wang X, Schuurmans D, Bosma M. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022 Nov 28; New Orleans, LA, USA. p. 24824–37. doi:10.5555/3600270.3602070.
20. Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y. Large language models are zero-shot reasoners. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022 Nov 28; New Orleans, LA, USA. p. 22199–213. doi:10.5555/3600270.3601883.
21. Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov 7–11; Punta Cana, Dominican Republic. p. 3045–59.
22. Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, et al. Self-consistency improves chain of thought reasoning in language models. arXiv:2203.11171. 2023.