electronics

Article

Enhancing Large Language Models for Specialized Domains: A Two-Stage Framework with Parameter-Sensitive LoRA Fine-Tuning and Chain-of-Thought RAG

Yao He, Xuanbing Zhu, Donghan Li and Hongyu Wang *

The School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China; 12309010@mail.dlut.edu.cn (Y.H.); zhuxuanbing@mail.dlut.edu.cn (X.Z.); lidonghan@mail.dlut.edu.cn (D.L.)
* Correspondence: whyu@dlut.edu.cn

Abstract: Large language models (LLMs) have shown impressive general-purpose language capabilities, but their application in specialized domains such as healthcare and law remains limited due to two major challenges, namely, a lack of deep domain-specific knowledge and the inability to incorporate real-time information updates. This paper focuses on addressing these challenges by introducing parameter-sensitive low-rank adaptation (LoRA) and retrieval-augmented generation (RAG), named SensiLoRA-RAG, a two-stage framework designed to enhance LLM performance in domain-specific question-answering tasks. In the first stage, we propose a parameter-sensitive LoRA fine-tuning method that efficiently adapts LLMs to specialized domains using limited high-quality professional data, enabling rapid and resource-efficient specialization. In the second stage, we develop a chain-of-thought RAG mechanism that dynamically retrieves and integrates up-to-date external knowledge, improving the model's ability to reason with current information and complex domain context. We evaluate our framework on tasks in the medical and legal fields, demonstrating that SensiLoRA-RAG significantly improves answer accuracy, domain relevance, and adaptability compared to baseline methods.

Keywords: large language models; fine-tuning; retrieval-augmented generation

check for updates

Academic Editor: Domenico Rosaci

Received: 8 April 2025
Revised: 5 May 2025
Accepted: 6 May 2025
Published: 11 May 2025

Citation: He, Y.; Zhu, X.; Li, D.; Wang, H. Enhancing Large Language Models for Specialized Domains: A Two-Stage Framework with Parameter-Sensitive LoRA Fine-Tuning and Chain-of-Thought RAG. *Electronics* **2025**, *14*, 1961. https://doi.org/10.3390/electronics14101961

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

General large language models (LLMs) are advanced artificial intelligence (AI) systems trained on vast amounts of text data to understand and generate human-like language. These models, exemplified by systems like GPT-4 [1], Claude [2], and LLaMA [3,4], represent a significant evolution beyond traditional language models [5–7] through their scale and capabilities. The key advantage of LLMs lies in their versatility and adaptability. Through prompt engineering, users can define specific roles, contexts, or constraints that guide the model's responses without requiring technical modifications to the underlying architecture [1].

Despite their impressive capabilities, LLMs face some significant challenges when applied to specialized domains such as healthcare and law. These domains are characterized by highly specialized and continuously evolving bodies of knowledge. Legal systems frequently introduce new regulations and case rulings [8], and the medical field rapidly produces new research findings and treatment guidelines [9]. Unfortunately, LLMs are typically trained on large, general-purpose datasets that may lack coverage of the most current or domain-specific information. As a result, their knowledge often lags behind real-world developments, which can lead to inaccurate or outdated answers in high-stakes scenarios.

Furthermore, both medical and legal texts are filled with complex terminologies and nuanced expressions that require deep contextual understanding. For human practitioners, years of training are required to master this language. LLMs, in contrast, frequently struggle with accurately interpreting such content, leading to misinterpretations and potential misinformation.

Another critical drawback is that LLMs cannot efficiently update knowledge dynamically. Since their knowledge is fixed at the time of training, they cannot incorporate new knowledge, such as recent legal rulings or medical advancements, without being retrained. However, retraining is costly in both computation and resources, making frequent updates impractical [10]. As a result, LLMs often provide outdated or incomplete responses, especially in fast-evolving fields, limiting their effectiveness in real-time specialized domains.

To address the limitations of general large language models in specialized domains, we propose SensiLoRA-RAG, a novel two-stage framework that enhances model performance with minimal domain-specific data and computational cost. Firstly, considering that it is difficult to construct a large number of standard training datasets in a specific domain, in the first stage, we design a novel parameter sensitivity low-rank adaptation (parameter sensitivity LoRA) to improve the domain adaptation of general LLMs. Its advantage is that it enables the general LLM to quickly absorb and understand specific domain knowledge, so as to enhance the performance of understanding the retrieval content in the second stage. Specifically, unlike standard LoRA, which assigns a fixed rank to all weight matrices, our method can dynamically adjust ranks based on parameter sensitivity. This allows the model to efficiently learn domain-specific knowledge with only a small amount of data, enabling rapid adaptation without the need for extensive retraining.

Secondly, to further address the challenge of real-time knowledge updating, inspired by the chain of thought of DeepSeek [11], we propose a novel retrieval-augmented generation (RAG) approach based on a long-thought chain. Specifically, instead of relying solely on static training data, the model can intelligently retrieve and integrate up-to-date external information before generating responses. The reasoning chain consists of three steps: (1) user intent analysis, where the model analyzes the core intent behind a query; (2) solution thinking, where it evaluates possible response strategies and thinks about whether to use external knowledge; and (3) response generation, where LLM generates the response based on external knowledge or not. By combining adaptive fine-tuning with dynamic retrieval, SensiLoRA-RAG enables LLMs to adapt efficiently in specialized domains while staying updated with the latest knowledge, making it well-suited for real-world specialized domains.

In a nutshell, we summarize our main contributions as follows:

- We propose SensiLoRA-RAG, a novel two-stage framework designed to enhance the performance of general LLMs in specialized domains. By combining adaptive fine-tuning with real-time retrieval, our approach enables efficient domain adaptation while maintaining access to the latest knowledge. This framework significantly reduces the need for extensive retraining, making it cost-effective.
- We introduce a parameter-sensitive LoRA fine-tuning algorithm, which dynamically adjusts rank allocation based on parameter sensitivity, improving fine-tuning efficiency and effectiveness with minimal domain-specific data. Additionally, we propose a chain-of-thought RAG retrieval strategy, which enhances knowledge integration by guiding retrieval through a structured reasoning process.

- We conducted comprehensive experiments on two specialized domains, i.e., law and healthcare, demonstrating that our method significantly improves the performance of general LLMs in domain-specific tasks.

2. Related Works

2.1. Large Language Models

Large language models (LLMs) have emerged as a transformative technology in artificial intelligence, significantly advancing natural language processing (NLP) and reshaping human–computer interaction paradigms. These models, typically based on Transformer architectures and trained on web-scale corpora, demonstrate remarkable capabilities in tasks such as machine translation, text generation, and information retrieval [12–15].

Early NLP models like Word2Vec and GloVe focused on learning high-quality word embeddings to capture semantic relationships in vector spaces [6,7]. However, the introduction of the Transformer architecture marked a turning point. OpenAI's GPT series, particularly GPT-3 with 175 billion parameters, showcased the emergent phenomenon of in-context learning—enabling zero-shot and few-shot generalization across diverse tasks without fine-tuning [16]. This laid the groundwork for a new interaction model with LLMs.

A major milestone was the deployment of ChatGPT (https://openai.com/chatgpt/overview/, accessed on 8 April 2025), which introduced reinforcement learning from human feedback (RLHF) to align model outputs with user intent. Its three-stage training pipeline (supervised fine-tuning, reward modeling, and policy optimization) significantly improved controllability and safety [17]. GPT-4 further extended these advances into the multimodal domain, achieving near-human performance on various professional benchmarks through improved data curation and training techniques [1]. Open-source efforts have played a crucial role in democratizing LLM research. Meta's LLaMA series (7B–65B) demonstrated how architectural optimizations could yield competitive performance with fewer parameters [3]. LLaMA-2 introduced a refined RLHF pipeline to enhance safety and alignment [4], while variants such as Vicuna [18], Guanaco [19], and Mistral [20] explored efficient fine-tuning and attention mechanisms, making LLMs more accessible and adaptable. Google's PaLM family pushed the frontier of scaling. PaLM-540B, trained using the Pathways system on thousands of TPUs, achieved state-of-the-art results on multi-step reasoning [21]. Follow-up work, Flan-PaLM, systematically validated instruction fine-tuning across 1800+ tasks, confirming the importance of task diversity and scale in generalization [22]. Domain-specific LLMs are also gaining traction. Med-PaLM demonstrated near-expert performance on medical QA benchmarks like MedQA [23], illustrating the potential of LLMs in vertical applications. PaLM-2 emphasized computational efficiency through mixed-objective training, highlighting the role of optimization techniques in scalable deployment [24].

2.2. Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) not only addresses the resource constraints encountered when customizing large models for downstream tasks but also significantly enhances their performance in these tasks. By avoiding the catastrophic forgetting inherent in full fine-tuning of large models, PEFT demonstrates broad applicability across diverse architectures, including large language models, large vision models, and multimodal models. As a technical approach, PEFT offers an efficient, highly generalizable, and scalable solution for adapting pre-trained models to specialized use cases [25].

One approach to parameter-efficient fine-tuning (PEFT) involves freezing the parameters of a pre-trained model while introducing only a minimal number of trainable parameters into the model architecture. During adaptation to downstream tasks, only the added modules or newly introduced parameters are updated, thereby significantly reducing computational resources such as storage and memory. Since this method inherently relies on introducing additional parameters, it can be categorized as additive adaptation. The serial adapter, one of the earliest adapters applied in NLP, incorporates adapter modules after both the self-attention layer and the feed-forward network (FFN) layer within each Transformer block to learn task-specific knowledge for downstream tasks [26]. AdapterFusion improves computational efficiency by introducing adapter layers exclusively after the normalization layer, following the FFN layer [27]. Prefix tuning generates prefix vectors via a multi-layer perceptron (MLP) to ensure stability during optimization, appending these learnable vectors to the key and value components of Transformer layers [28]. However, this method does not account for the varying importance of prefixes across different layers. To address this limitation, adaptive prefix tuning (APT) introduces an adaptive gating mechanism to dynamically regulate prefix importance, thereby enhancing the flexibility and effectiveness of prefix tuning [29].

Another approach to parameter-efficient fine-tuning (PEFT) involves selecting a critical subset of existing parameters. Specifically, a binary mask matrix is applied to each parameter matrix of the large model, where elements in the mask correspond one-to-one to those in the parameter matrix. A value of 1 in the binary mask indicates that the corresponding parameter requires fine-tuning, while 0 signifies that the parameter remains frozen. Diff Pruning employs a differentiable approximation of the L0-norm penalty to regularize learnable mask matrices, thereby improving parameter efficiency [30]. PaFi adopts a straightforward strategy by directly selecting parameters with the smallest absolute magnitudes as trainable [31]. FishMask first calculates parameter importance based on an approximation of the Fisher information and retains the top-k most important parameters for training [32]. LT-SFT, inspired by the lottery ticket hypothesis [33,34], identifies a subset of parameters that exhibit the largest changes in training loss during the initial fine-tuning phase to construct the mask matrix [35].

Pre-trained large models often exhibit extremely low intrinsic dimensionality [36]. In other words, the entire parameter space of a pre-trained large model can be represented by a low-rank space. This implies the existence of reparameterization methods that fine-tune only the low-rank space, with LoRA being the most representative approach. LoRA introduces a low-rank down-projection matrix and a low-rank up-projection matrix, whose product serves as an update matrix [37]. This update matrix is merged into the original parameter matrix via summation. However, LoRA raises the challenge of selecting an optimal rank. To address this limitation, DyLoRA pre-defines a range of ranks and dynamically adjusts the rank within this range during each training iteration, avoiding the use of a fixed unified rank [38]. AdaLoRA redefines the update matrix using singular value decomposition (SVD) as three learnable parameter matrices [39]. During training, it computes importance scores based on the moving average of gradient-weight product magnitudes and prunes singular values accordingly to achieve a dynamically adjusted rank. VeRA (vector-based random matrix adaptation) shares the low-rank up-projection and down-projection matrices across all layers, training only two learnable scaling vectors and a diagonal matrix to adapt these shared matrices [40].

2.3. Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) has emerged as a powerful solution to mitigate the hallucination problem in large language models, particularly in domain-specific and knowledge-intensive tasks [41–46]. By retrieving relevant document fragments from an external knowledge base through semantic similarity matching, RAG enriches LLM-generated responses with accurate and contextually relevant information. This approach significantly improves factual accuracy and enhances the overall reliability of generated content. The RAG research paradigm is evolving, and we categorize its development into three stages: naive RAG [47], advanced RAG, and modular RAG.

The naive RAG paradigm is an early approach that integrates retrieval and generation, enabling LLMs to incorporate external knowledge into their responses. It follows a "Retrieve-Read" framework, where relevant information is first retrieved from a knowledge base and then used to generate informed responses. However, it faces significant challenges at the retrieval, generation, and augmentation stages, as follows: (1) retrieval challenges: naive RAG struggles with both precision and recall in retrieval, often retrieving misaligned document chunks while failing to capture critical information [42,48]; (2) generation difficulties: despite leveraging external documents, the model remains prone to hallucinations, generating responses that lack sufficient support from retrieved content. Another key concern is that the model may over-rely on retrieved data, restating external content rather than synthesizing meaningful insights or constructing well-reasoned responses [41]; (3) augmentation hurdles: integrating retrieved information into the generation process is challenging, as merging content from multiple sources may lead to disjointed or incoherent responses. Redundancy is another issue, as overlapping information from different sources can result in repetitive outputs. Furthermore, identifying the most relevant passages remains a complex task, affecting the model's ability to generate concise, well-structured, and contextually appropriate responses [43].

Advanced RAG builds upon the naive RAG framework by refining both the retrieval and augmentation processes to improve response precision and contextual relevance. It addresses key limitations by improving how external knowledge is selected, ranked, and incorporated into the model's final output. The framework optimizes pre-retrieval [47,49–51] and post-retrieval stages to ensure that retrieved information is contextually appropriate and seamlessly integrated into the generation process. The pre-retrieval process focuses on improving document indexing and retrieval strategies to maximize relevance. Unlike traditional RAG, which retrieves information in a single pass, advanced methods enhance retrieval quality through query expansion, semantic filtering, and multi-step retrieval. Query expansion techniques refine search terms, while semantic filtering eliminates irrelevant content before retrieval. Additionally, iterative or multi-hop retrieval methods allow the system to gather deeper contextual information by dynamically refining the query based on initial results. These enhancements significantly improve retrieval precision, ensuring that the model processes only the most relevant knowledge. Once relevant content is retrieved, effectively integrating it into the model's response is essential for coherence and relevance. The post-retrieval stage utilizes re-ranking and context compression to prioritize and refine retrieved information. Re-ranking places the most critical content at the forefront of the model's input, minimizing distractions from less relevant details.

Compared to naive RAG and advanced RAG, modular RAG employs a variety of strategies to enhance the retrieval module. For example, Wang et al. [52] introduced additional search modules to refine retrieval quality and expand relevant content. RAGFusion [53] proposes a multi-query strategy that expands a single user query into multiple related queries and improves retrieval accuracy through parallel vector search and intelligent re-ranking algorithms. Chen [54] and Wang et al. [55] proposed a novel memory framework that utilizes generated content as "self-memory", continuously enriching the available memory pool through iteration. The memory selector identifies the most relevant content for the next generation round, overcoming the constraints of a fixed retrieval library and enhancing generation quality. Since some user queries may be semantically ambiguous, making it difficult to retrieve relevant content, Ma et al. [47] proposed rewrite–retrieve–read, which leverages LLMs to refine and clarify user queries before retrieval. Relying on a single retrieval algorithm often results in suboptimal retrieval coverage and accuracy. Ren et al. [10] proposed a hybrid retrieval strategy that significantly enhances retrieval accuracy by incorporating multiple approaches, including keyword-based, semantic, and vector search techniques.

3. Preliminaries: LLM

Large language models (LLMs) are deep neural networks trained on the vast corpora of textual data to generate, comprehend, and process human language. They are typically built on the Transformer architecture [56], which uses self-attention mechanisms to model long-range dependencies and contextual relationships in text. During training, the model learns general linguistic patterns and factual knowledge by optimizing the next-token prediction objective, where it estimates the probability distribution of the next token given a sequence of preceding tokens:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta), \tag{1}$$

where $x_1, x_2, \ldots, x_T$ represents a sequence of input tokens, and θ denotes the parameters of the model.

4. Proposed Method: SensiLoRA-RAG

Although the LLMs show strong performance, they still struggle in specialized domains, such as healthcare and law. This is mainly due to the lack of domain-specific knowledge in LLMs and the high cost of retraining. Then, to address these issues, we propose a novel two-stage framework, SensiLoRA-RAG, which enables LLMs to adapt efficiently in specialized domains while staying updated with the latest knowledge, making it well-suited for real-world specialized domains, as shown in Figure 1. Next, we will elaborate on these two stages.

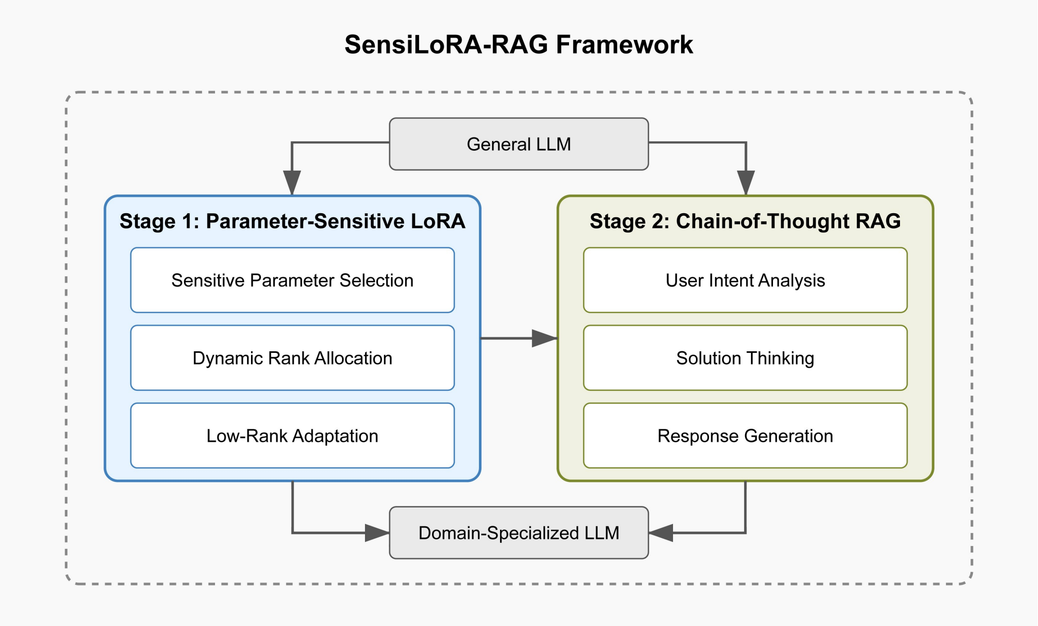


Figure 1. The illustration of our proposed SensiLoRA-RAG framework.

4.1. Parameter Sensitivity LoRA

Parameter-efficient fine-tuning (PEFT) methods, such as LoRA, address computational resource limitations when adapting pre-trained large models to downstream tasks. However, LoRA suffers from two critical limitations: (1) the need to manually select task-specific ranks for different downstream tasks; (2) the neglect of varying parameter activation patterns across tasks. To overcome these issues, we propose a dynamic rank algorithm guided by sensitive parameter ratios, as shown in Figure 2, which operates through the following three key stages: sensitive parameter selection, dynamic rank allocation, and LoRA-based fine-tuning.

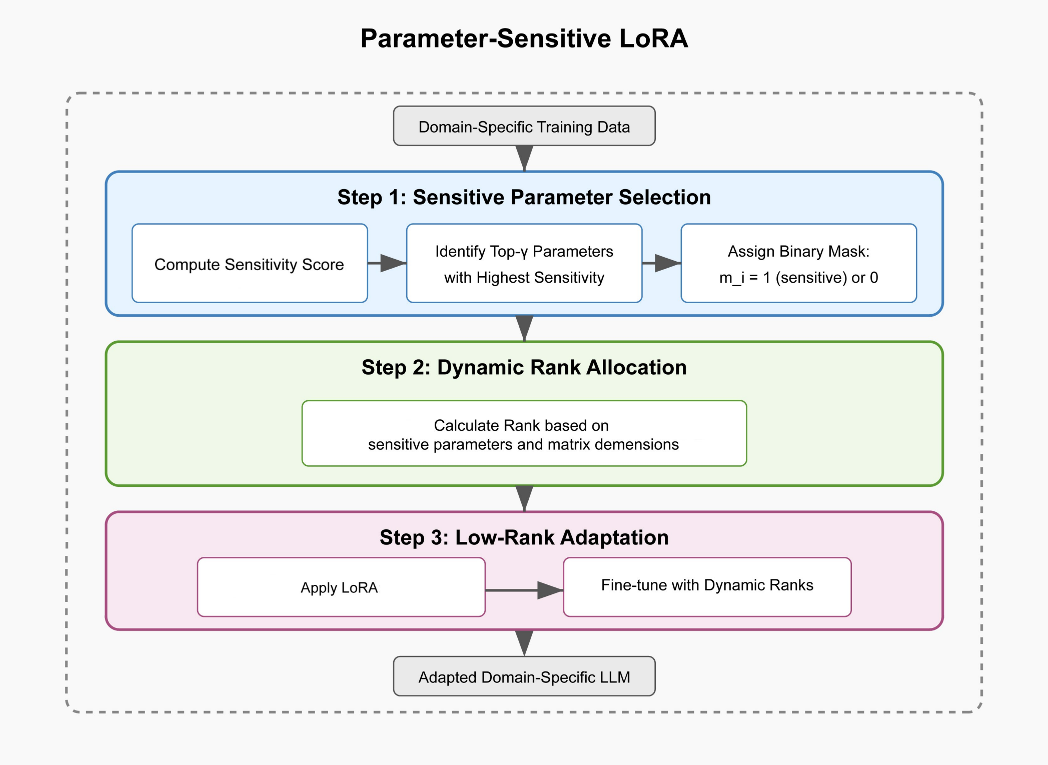

Parameter-Sensitive LoRA
Domain-Specific Training Data
Step 1: Sensitive Parameter Selection
Compute Sensitivity Score
Identify Top-γ Parameters with Highest Sensitivity
Assign Binary Mask: m_i = 1 (sensitive) or 0
Step 2: Dynamic Rank Allocation
Calculate Rank based on sensitive parameters and matrix demensions
Step 3: Low-Rank Adaptation
Apply LoRA
Fine-tune with Dynamic Ranks
Adapted Domain-Specific LLM

Figure 2. The illustration of our proposed parameter sensitivity LoRA method.

(1) Step 1—selecting sensitive parameters: To determine parameter sensitivity, inspired by [57], we compute a sensitivity score s_i for each parameter $w_i \in \{w_0, \ldots, w_i, \ldots, w_N\}$. The sensitivity is defined as the time-averaged squared gradient magnitude across all training samples in the downstream task dataset $|D^{tr}|$. Formally, this is calculated as follows:

$$s_i = \frac{1}{|D^{tr}|} \sum_{x \in D^{tr}} (\nabla \mathcal{L}_x(w_i))^2, \tag{2}$$

where $|D^{tr}|$ denotes the total number of training samples, and $\nabla \mathcal{L}_x(w_i)$ represents the gradient of parameter w_i with respect to the loss for sample x. Subsequently, the top-γ parameters with the highest sensitivity scores are identified as critical parameters. To explicitly mark these parameters, we assign a binary mask m_i to each parameter matrix w, where $m_i = 1$ flags sensitive parameters and $m_i = 0$ indicates non-sensitive ones.

(2) Step 2—allocating adaptive rank: We observed that different parameter matrices exhibit varying numbers of sensitive parameters, leading to a natural intuition, which is that matrices with higher ratios of sensitive parameters should be assigned higher ranks, while

those with lower ratios receive lower ranks. Formally, for a parameter matrix $W \in \mathbb{R}^{d \times k}$, its rank r is determined by the following:

$$r = \frac{s \times \alpha}{d \times k}, \tag{3}$$

where s denotes the number of sensitive parameters, d and k are the matrix dimensions, and α is a hyperparameter controlling the global rank budget.

(3) Step 3—LoRA-based fine-tuning: After dynamically assigning ranks to each parameter matrix, we directly apply the standard LoRA fine-tuning procedure:

$$\Delta W = W_{\text{down}} W_{\text{up}}, \tag{4}$$

$W_{\text{up}} \in \mathbb{R}^{r \times k}$ and $W_{\text{down}} \in \mathbb{R}^{d \times r}$are LoRA trainable matrix, where the rank r is allocated in step 2.

4.2. Chain-of-Thought RAG

Although the fine-tuning process can effectively enhance the performance of general LLMs in specific fields, constructing such high-quality fine-tuning datasets in real-world scenarios is often challenging and time-consuming. While these specific domains lack fine-tuning datasets, they possess a wealth of available textual resources. We can consider making full use of the LLM's context-understanding ability to improve the performance of the general LLM in specific fields. Employing the RAG algorithm is a practical approach. By constructing a knowledge base from available textual resources, we can retrieve relevant document fragments based on the user's query and provide them to the LLM to enhance response quality. However, in real-world applications, we may face the following key problems: (1) users often submit concise or complex queries. If queries are used directly, they may fail to retrieve relevant content, negatively impacting response quality; (2) the user's query questions are not always related to the knowledge base content. For example, if the user asks "Hello", the LLM can reply without the knowledge base. To address this issue, inspired by DeepSeek's thinking-before-response approach [11], we propose a novel chain-of-thought RAG. As shown in Figure 3, the algorithm contains three key steps: user intent analysis, solution thinking, and response generation. Here, we elaborate on each step in order:

(1) Step 1—user intent analysis: Different from the existing work that generates multi-perspective user query questions, we propose employing LLM to thoroughly analyze the underlying intent behind the user's current query question. This process is very similar to the DeepSeek model, which thinks long before answering. Specifically, given a user's query question q, we construct a prompt for user intent mining, as illustrated in Figure 4a. By deeply understanding user queries, we can address the challenge of overly concise inputs, enhance retrieval accuracy, and ultimately improve response quality.

(2) Step 2—solution thinking: After thoroughly analyzing the user's intentions, we need to further think about how to solve the user's problems. Here, we may face the following key problems: first, the extracted user intentions from Step 1 may vary widely, meaning that some may not accurately reflect the user's true needs; second, user questions are not always related to the knowledge base content, which means that more flexible strategies are needed to schedule knowledge base retrieval, which can improve both the speed and quality of replay. To address this, we introduce a structured problem-solving prompt, illustrated in Figure 5. In this prompt, we guide the LLM to think about the solution of the user's intention, and further guide the LLM on whether to use the retrieve tool. The parameter *retrieveTool* explicitly defines the conditions under which retrieval should be applied.

(3) Step 3—response generation: Following the previous step, we use JSON parsing to analyze the LLM's response and determine whether to incorporate retrieved content or generate a direct reply. Accordingly, we design the final response prompt, as illustrated in Figure 4b.

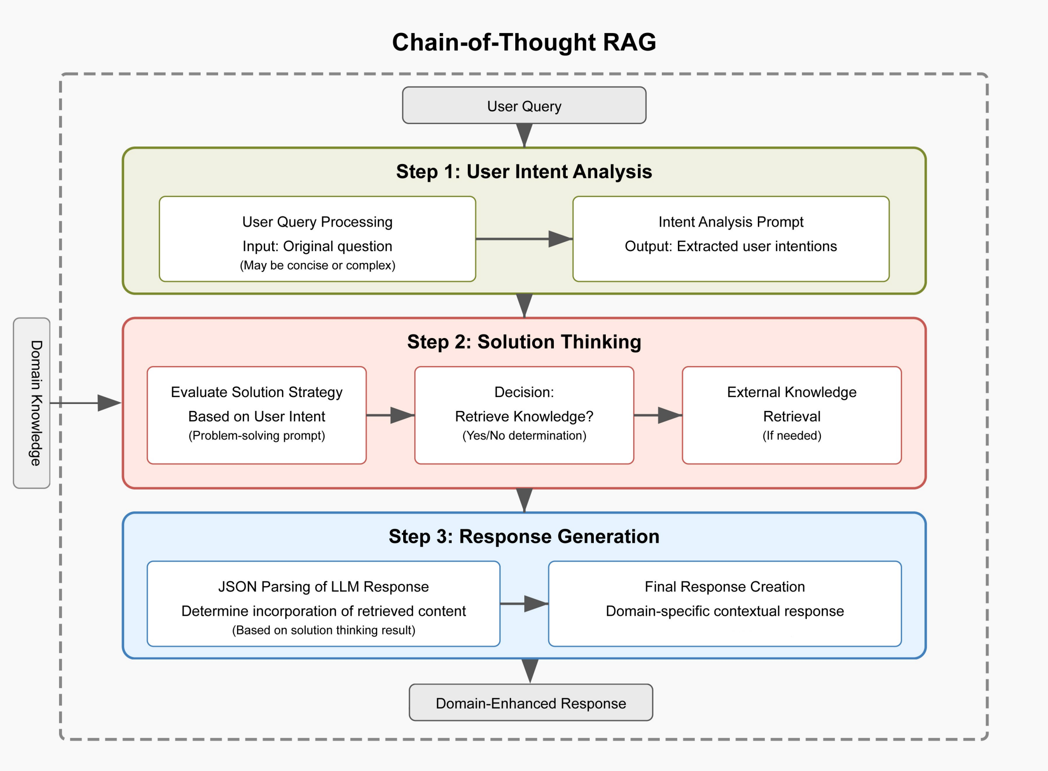
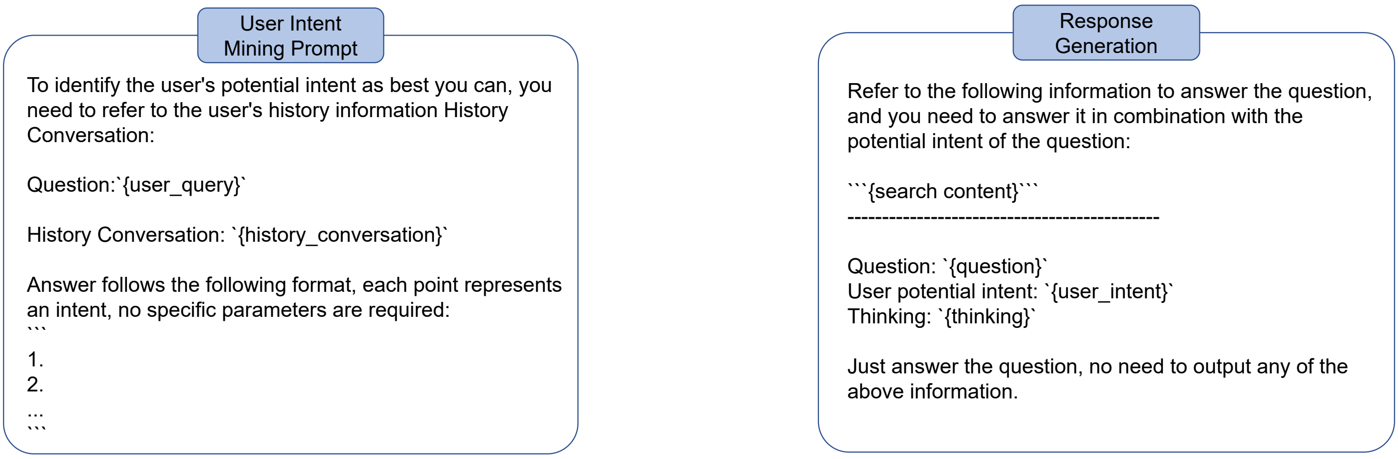

Chain-of-Thought RAG
User Query
Step 1: User Intent Analysis
User Query Processing
Input: Original question
(May be concise or complex)
Intent Analysis Prompt
Output: Extracted user intentions
Domain Knowledge
Step 2: Solution Thinking
Evaluate Solution Strategy
Based on User Intent
(Problem-solving prompt)
Decision:
Retrieve Knowledge?
(Yes/No determination)
External Knowledge
Retrieval
(If needed)
Step 3: Response Generation
JSON Parsing of LLM Response
Determine incorporation of retrieved content
(Based on solution thinking result)
Final Response Creation
Domain-specific contextual response
Domain-Enhanced Response

Figure 3. The illustration of our proposed chain-of-thought RAG method.



User Intent Mining Prompt

~~~~
To identify the user's potential intent as best you can, you
need to refer to the user's history information History
Conversation:

Question:`{user_query}`

History Conversation: `{history_conversation}`

Answer follows the following format, each point represents
an intent, no specific parameters are required:
```
1.
2.
...
```
~~~~

(**a**)

Response Generation

~~~~
Refer to the following information to answer the question,
and you need to answer it in combination with the
potential intent of the question:

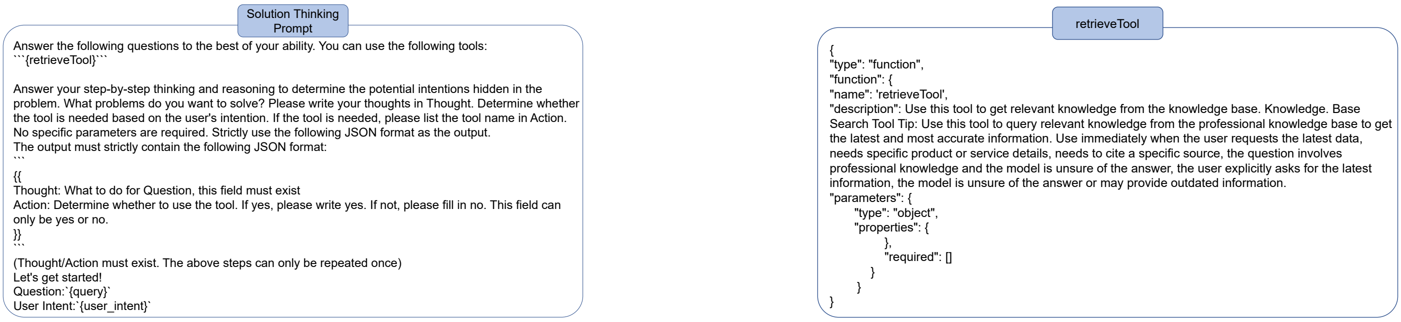
```{search content}```
----------------------------------------------

Question: `{question}`
User potential intent: `{user_intent}`
Thinking: `{thinking}`

Just answer the question, no need to output any of the
above information.
~~~~

(**b**)

Figure 4. Illustrate the user intent analysis prompt and generated prompt of chain-of-thought RAG: (**a**) user intent analysis prompt, which guides the LLM to deeply mine the user's true intention. (**b**) The generation prompt, which guides the LLM to answer user query questions based on the retrieved content and user intention.


Solution Thinking Prompt
retrieveTool

~~~
Answer the following questions to the best of your ability. You can use the following tools:
```{retrieveTool}```

Answer your step-by-step thinking and reasoning to determine the potential intentions hidden in the
problem. What problems do you want to solve? Please write your thoughts in Thought. Determine whether
the tool is needed based on the user's intention. If the tool is needed, please list the tool name in Action.
No specific parameters are required. Strictly use the following JSON format as the output.
The output must strictly contain the following JSON format:
```
{{
Thought: What to do for Question, this field must exist
Action: Determine whether to use the tool. If yes, please write yes. If not, please fill in no. This field can
only be yes or no.
}}
```
(Thought/Action must exist. The above steps can only be repeated once)
Let's get started!
Question:`{query}`
User Intent:`{user_intent}`
~~~

(**a**)

```
{
"type": "function",
"function": {
"name": 'retrieveTool',
"description": Use this tool to get relevant knowledge from the knowledge base. Knowledge. Base
Search Tool Tip: Use this tool to query relevant knowledge from the professional knowledge base to get
the latest and most accurate information. Use immediately when the user requests the latest data,
needs specific product or service details, needs to cite a specific source, the question involves
professional knowledge and the model is unsure of the answer, the user explicitly asks for the latest
information, the model is unsure of the answer or may provide outdated information.
"parameters": {
    "type": "object",
    "properties": {
        },
        "required": []
      }
    }
}
```

(**b**)

Figure 5. Illustrates the solution-thinking prompt of chain-of-thought RAG: (**a**) the solution thinking prompt, which guides the LLM to give solutions based on user intention and guides the LLM to determine whether to use the retrieve tool; (**b**) explains the definition of the parameter "retrieveTool".

5. Experiments

5.1. Experimental Settings

Datasets. Following LAW-GPT, we evaluate our method of parameter sensitivity LoRA on two datasets, including legal knowledge question-answering and medical knowledge question-answering tasks. For the legal knowledge question-answering dataset, we used **crimekgassistant-52k**; a multi-category legal consultation question-answering dataset was reconstructed using ChatGPT based on crimekgassistant's original QA pairs, featuring regenerated responses with enhanced detail and improved linguistic standardization across thirteen case types, including marriage and family, labor disputes, traffic accidents, credit and debts, criminal defense, contract disputes, real estate disputes, infringement, company law, medical disputes, relocation and resettlement, administrative litigation, and construction projects. This dataset served as pretraining data for Xiezhi, a Chinese legal large language model. **Huatuo-65k:** Huatuo-26M, currently the largest traditional Chinese medicine (TCM) question-answering dataset, comprises over 26 million high-quality medical QA pairs. These pairs encompass comprehensive topics including diseases, symptoms, treatment methods, pharmaceutical information, and TCM theory. The creation of this dataset provides invaluable resources for research in natural language processing (NLP), information extraction, and question-answering systems within the TCM domain. To align the scale with legal QA datasets, we randomly sampled 60,000 entries for the training set and 5000 entries for the test set from this collection.

Evaluation Metrics. To evaluate the performance of the proposed method, we select BLEU, ROUGE, Bert-sim, and edit distance as evaluation metrics [5,58–60]. These metrics are widely recognized standards for assessing text generation quality, particularly in tasks involving open-ended or semi-structured outputs. BLEU measures the n-gram precision between the generated and reference texts, offering a quantitative assessment of content overlap. ROUGE focuses on recall-oriented measures, capturing the degree to which the generated text covers the reference information. Bert-sim measures the vector similarity of two sentences in the semantic space, while edit distance measures the minimum number of operations required to transform one string into another, including insertion, deletion, and substitution of characters.

Implementation Details. Following LAW-GPT, we employ the Lion optimizer to fine-tune the model for 10 epochs. All experiments were conducted on an NVIDIA A6000 GPU.

5.2. Discussion of Results

Question-Answering Task. We evaluated LoRA, AdaLoRA, DyLoRA, and our SensiLoRA on the crimekgassistant-52k and Huatuo-65k datasets. The results are listed in Tables 1 and 2. As can be seen from the results, our SensiLoRA outperforms the baseline in multiple metrics, including BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, fully demonstrating that the answers generated after fine-tuning with our method have better linguistic fluency. Furthermore, our method also shows improvements in two similarity metrics, proving that the answers generated after fine-tuning with our method are closer to the reference answers. Specifically, (1) our method achieved an improvement of approximately 12∼37.5% in BLEU and ROUGE metrics in the legal question-answering task compared to LoRA. (2) Our method achieved a significant improvement in BLEU in the medical question-answering task.

Table 1. Results of question-answering on the law dataset.

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Bert-Sim	Edit Distance
LoRA	3.51×10^{-82}	1.60×10^{-4}	2.93×10^{-5}	1.60×10^{-4}	0.682	207.3
AdaLoRA	3.71×10^{-82}	1.72×10^{-4}	3.04×10^{-5}	1.72×10^{-4}	0.687	184.4
DyLoRA	3.81×10^{-82}	1.83×10^{-4}	3.06×10^{-5}	1.83×10^{-4}	0.700	180.0
SensiLoRA	4.15×10^{-82}	2.20×10^{-4}	3.28×10^{-5}	2.20×10^{-4}	0.700	172.7

Table 2. Results of question-answering on the medical dataset.

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Bert-Sim	Edit Distance
LoRA	9.52×10^{-233}	3.42×10^{-2}	1.82×10^{-5}	3.42×10^{-2}	0.706	299.6
AdaLoRA	8.67×10^{-233}	2.90×10^{-2}	7.27×10^{-5}	2.90×10^{-4}	0.710	291.1
DyLoRA	7.73×10^{-233}	2.85×10^{-2}	2.16×10^{-4}	2.85×10^{-2}	0.727	296.0
SensiLoRA	7.70×10^{-5}	3.47×10^{-2}	2.68×10^{-4}	3.47×10^{-2}	0.728	291.1

5.3. Ablation Study

How do hyperparameters impact our method's performance? To investigate this, in Table 3, we analyze the influence of hyperparameter variations on the medical QA dataset. We tested hyperparameter values ranging from 1100 to 1300 while using LoRA as the baseline. The results show that while performance fluctuates slightly with different hyperparameter values, our method consistently surpasses LoRA. This confirms that hyperparameter settings do not compromise the robustness of our approach.

Table 3. Analysis of hyperparameter α.

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Bert-Sim	Edit Distance
LoRA	9.52×10^{-233}	3.42×10^{-2}	1.82×10^{-5}	3.42×10^{-2}	0.706	299.6
SensiLoRA with $\alpha = 1100$	9.67×10^{-5}	3.50×10^{-2}	5.71×10^{-5}	3.50×10^{-2}	0.728	299.3
SensiLoRA with $\alpha = 1200$	7.70×10^{-5}	3.47×10^{-2}	2.68×10^{-4}	3.47×10^{-2}	0.728	299.4
SensiLoRA with $\alpha = 1300$	9.57×10^{-233}	3.50×10^{-2}	1.89×10^{-4}	3.50×10^{-2}	0.728	291.1

Is our dynamic rank method effective? To answer this question, in Table 4, we conducted experiments on the medical QA dataset to validate the efficacy of the dynamic rank approach. Specifically, we compared SensiLoRA against LoRA baselines with fixed ranks of 8, 16, and 32. The results demonstrate that SensiLoRA consistently outperforms LoRA across all tested ranks, proving the effectiveness of our dynamic rank strategy.

Table 4. Analysis of rank.

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Bert-Sim	Edit Distance
LoRA-8	9.52×10^{-233}	3.42×10^{-2}	1.82×10^{-5}	3.42×10^{-2}	0.706	299.6
LoRA-16	9.11×10^{-233}	3.38×10^{-2}	1.78×10^{-4}	3.38×10^{-2}	0.714	294.3
LoRA-32	9.76×10^{-233}	3.47×10^{-2}	1.05×10^{-4}	3.47×10^{-2}	0.726	292.6
SensiLoRA	7.70×10^{-5}	3.47×10^{-2}	2.68×10^{-4}	3.47×10^{-2}	0.728	291.1

Can our CoT-RAG method improve response quality? We visualized a reply case in Tables 5 and 6. Fine-tuning refers to training the model with parameter sensitivity LoRA, while RAG denotes a retrieval-augmented generation method that lacks user intent analysis and solution thinking. As shown in the table, our method accurately provides the correct answer, whereas the RAG method fails to do so. We speculate that the model may not really understand the intention of the user's question, resulting in the model still giving an incorrect reply even if the context information is given. This demonstrates that guided reasoning enables the model to better grasp the user's true intent, leading to more accurate responses and improved overall reply quality, thereby validating the effectiveness of our method.

Table 5. Comparison of responses from different models to the question: *Xiao Zhang's online pharmacy wants to apply for a change in business scope, including the addition of a new type of drug. How long does Xiao Zhang have to wait to find out whether the provincial Food and Drug Administration approves his change application?* (Green highlights indicate correct answers, while red highlights indicate incorrect responses).

Question's Answer	Fine-Tuning	RAG	CoT-RAG
Answer: According to Article 20 of the "Internet Drug Information Service Management Measures", the provincial food and drug supervision and administration department needs to make a review decision within 20 working days after accepting the change application, and notify the applicant in writing, stating the reasons. Therefore, Xiao Zhang needs to wait 20 working days to learn the review result.	Xiao Zhang needs to wait for the provincial food and drug administration to review and approve his change application, which usually takes 60 to 180 days. The specific time depends on the application approval process and the number of required materials. If the application is approved, Xiao Zhang can start producing, marketing, and selling new drugs. If the application is rejected, Xiao Zhang needs to reapply or seek other solutions.	Answer: Based on the information provided in the reference information, it is not possible to determine how long Xiao Zhang needs to wait to know whether the provincial food and drug administration approves his change application. This depends on factors such as whether the business scope and new drugs applied by Xiao Zhang are within the supervision scope of the Food and Drug Administration, as well as the time of application and the processing speed of relevant departments. Therefore, it is impossible to give a definite answer.	Answer: Xiao Zhang needs to wait for the approval result from the provincial food and drug administration. The specific approval time depends on the work efficiency of the provincial food and drug administration and the approval speed of the individual application. Generally speaking, the approval time ranges from 1 to 3 weeks, but the specific approval time will also be affected by factors such as the approval department, the number of applicants, and the quality of the application materials. Therefore, Xiao Zhang needs to communicate with the provincial food and drug administration in advance to understand the approval process and the time required to make good time arrangements.

Table 6. Comparison of responses from different models to the question: *My baby has a fever caused by a cold. He has taken cough medicine and antipyretics, but his body temperature is still high. What should I do?* (Green highlights indicate correct answers, while red highlights indicate incorrect responses).

Question's Answer	Fine-Tuning	RAG	CoT-RAG
Answer: If you still feel your body temperature rises after taking antipyretics, it is recommended that you first determine whether the frequency and dosage of the medication are correct. If they are correct, you can consider changing to antipyretics with different ingredients or strengthening the body's cooling measures, such as ice compresses. As for the fever caused by the baby's cold, it is recommended that you consider symptomatic treatment with Chinese medicine or medications with appropriate temperatures after determining whether the baby's body temperature has reached the fever standard. At the same time, you should also pay attention to the baby's diet and rest, and keep the indoor air circulation and temperature appropriate. As for taking cough medicine, you need to understand the specific symptoms and drug ingredients in order to make targeted suggestions. It is recommended that you go to a regular medical institution for further examination and treatment.	Answer: If the baby has a fever caused by cold and the body temperature is still high after taking cough medicine, it may be due to the excitement of the cough center caused by coughing, which leads to an increase in body temperature. It is recommended to seek medical attention in time and seek help from professional doctors. While waiting for medical treatment, you can use a wet towel or wet cotton swab to wipe the baby's forehead and armpits to help lower the body temperature. At the same time, pay attention to giving the baby plenty of water, keep the indoor air circulating, and avoid excessive activity of the baby. If other symptoms occur, such as worsening cough, shortness of breath, vomiting, etc., you should seek medical attention immediately. I wish the baby a speedy recovery!	Answer: According to your description, your baby still feels a fever after taking antipyretics. This may be caused by incorrect drug dosage or frequency of use. It is recommended that you adjust the frequency and dosage of the drug in time, and pay attention to the baby's warmth and rest, and avoid excessive activity and fatigue. For fever caused by cold in babies, it is recommended that you first determine whether the baby's body temperature meets the fever standard, and consider symptomatic treatment with Chinese medicine or drugs with appropriate temperature. At the same time, pay attention to the baby's diet and rest, and keep the indoor air circulation and temperature appropriate. If the symptoms worsen or last for a long time, it is recommended to seek medical attention in time for timely diagnosis and treatment. In short, it is recommended that you seek medical attention in time and conduct the necessary diagnosis and treatment in order to control the disease in time and improve the baby's quality of life and physical health.	Answer: Hello! According to the information you provided, your baby's body temperature is still high after taking antipyretics, and symptoms such as fever and cough caused by wind and cold appear, which require further treatment. First of all, you need to determine the correct dosage and usage of antipyretics. If the frequency and dosage of the medicine are correct, you can consider changing to antipyretics with different ingredients or strengthening the body's cooling measures, such as ice compresses. As for the fever caused by the baby's wind and cold, it is recommended that you consider symptomatic treatment with Chinese medicine or medicines with appropriate temperatures after determining whether the baby's body temperature has reached the fever standard. At the same time, you should also pay attention to the baby's diet and rest, and keep the indoor air circulation and temperature appropriate. As for the baby taking cough medicine, you need to understand the specific symptoms and drug ingredients in order to make targeted suggestions. It is recommended that you go to a regular medical institution for further examination and treatment.

6. Conclusions

In this work, we introduced SensiLoRA-RAG, a novel two-stage framework designed to address the challenges faced by general large language models (LLMs) in specialized domains. By integrating parameter-sensitive LoRA fine-tuning with a structured chain-of-thought-based retrieval-augmented generation (RAG) approach, our framework significantly enhances both domain adaptation and real-time knowledge integration. Extensive experiments conducted on domain-specific tasks in legal and healthcare settings validate the effectiveness of our approach. The results demonstrate that SensiLoRA-RAG significantly outperforms baseline models in both knowledge retention and real-time adaptability, reinforcing its potential for practical applications in specialized domains. In future work, we will (1) explore the scalability of SensiLoRA-RAG to a wider range of specialized domains beyond healthcare and law, such as finance, engineering, and education; (2) develop mechanisms to ensure the trustworthiness and explainability of retrieved information, particularly for high-stakes domains where transparency is critical and (3) examine the potential for continuous learning mechanisms that would allow the model to efficiently update its domain knowledge without extensive retraining. These directions represent promising opportunities to further enhance the capabilities of LLMs in specialized domains while addressing the challenge of domain adaptation.

Author Contributions: Conceptualization, Y.H. and H.W.; methodology, Y.H., H.W., X.Z. and D.L.; validation, Y.H.; writing—original draft preparation, Y.H.; writing—review and editing, Y.H. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Dalian Science and Technology Innovation Fund Project 2022JJ11CG002.

Data Availability Statement: The data presented in this study are available on request from the GitHub repositories: https://github.com/shibing624/MedicalGPT (accessed on 25 March 2025) and https://github.com/CSHaitao/LexiLaw (accessed on 25 March 2025).

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

Abbreviations

The following abbreviations are used in this manuscript:

LLM	large language model
RAG	retrieval-augmented generation
LoRA	low-rank adaptation

References

1. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
2. Anthropic. Claude 3 Haiku: Our Fastest Model Yet. 2024. Available online: https://www.anthropic.com/news/claude-3-haiku (accessed on 20 March 2025).
3. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
4. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv* **2023**, arXiv:2307.09288.
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
6. Goldberg, Y.; Levy, O. word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv* **2014**, arXiv:1402.3722.
7. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
8. Sánchez, D.B.; García, N.A.; Jiménez, Á.B.; Nieto, M.G.; Morales, P.M.; Salas, N.S.; Hernán, C.G.; Coll, P.H.; Ponsoda, E.M.; Ibá nez, P.C. MEL: Legal Spanish Language Model. *arXiv* **2025**, arXiv:2501.16011.
9. Wu, C.; Qiu, P.; Liu, J.; Gu, H.; Li, N.; Zhang, Y.; Wang, Y.; Xie, W. Towards evaluating and building versatile large language models for medicine. *npj Digit. Med.* **2025**, *8*, 58. [CrossRef]
10. Ren, Y.; Cao, Y.; Guo, P.; Fang, F.; Ma, W.; Lin, Z. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 293–306.
11. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv* **2025**, arXiv:2501.12948.
12. Fernandes, D.; Matos-Carvalho, J.P.; Fernandes, C.M.; Fachada, N. DeepSeek-V3, GPT-4, Phi-4, and LLaMA-3.3 Generate Correct Code for LoRaWAN-Related Engineering Tasks. *Electronics* **2025**, *14*, 1428. [CrossRef]

13. Jiao, J.; Jiang, L.; Zhou, Q.; Wen, R. Evaluating Large Language Model Application Impacts on Evasive Spectre Attack Detection. *Electronics* **2025**, *14*, 1384. [CrossRef]
14. Jiao, J.; Yang, H.; Wen, R. LLM-WFIN: A Fine-Grained Large Language Model (LLM)-Oriented Website Fingerprinting Attack via Fusing Interrupt Trace and Network Traffic. *Electronics* **2025**, *14*, 1263. [CrossRef]
15. Fragakis, N.; Trichopoulos, G.; Caridakis, G. Empowering Education with Intelligent Systems: Exploring Large Language Models and the NAO Robot for Information Retrieval. *Electronics* **2025**, *14*, 1210. [CrossRef]
16. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
17. Casper, S.; Davies, X.; Shi, C.; Gilbert, T.K.; Scheurer, J.; Rando, J.; Freedman, R.; Korbak, T.; Lindner, D.; Freire, P.; et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv* **2023**, arXiv:2307.15217.
18. Chiang, W.L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J.E.; et al. Vicuna: An Open-Source Chatbot Impressing gpt-4 with 90%* Chatgpt Quality. 2023. Available online: https://vicuna.lmsys.org (accessed on 14 April 2023).
19. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 10088–10115.
20. Jiang, A.Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D.S.; Casas, D.d.l.; Hanna, E.B.; Bressand, F.; et al. Mixtral of experts. *arXiv* **2024**, arXiv:2401.04088.
21. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 11324–11436.
22. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.* **2024**, *25*, 1–53.
23. Singhal, K.; Tu, T.; Gottweis, J.; Sayres, R.; Wulczyn, E.; Amin, M.; Hou, L.; Clark, K.; Pfohl, S.R.; Cole-Lewis, H.; et al. Toward expert-level medical question answering with large language models. *Nat. Med.* **2025**, *31*, 943–950. [CrossRef]
24. Anil, R.; Dai, A.M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. PaLM 2 technical report. *arXiv* **2023**, arXiv:2305.10403.
25. Bao, Q.; Zhao, J.; Liu, Z.; Liang, N. AI-Assisted Inheritance of Qinghua Porcelain Cultural Genes and Sustainable Design Using Low-Rank Adaptation and Stable Diffusion. *Electronics* **2025**, *14*, 725. [CrossRef]
26. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 2790–2799.
27. Pfeiffer, J.; Kamath, A.; Rücklé, A.; Cho, K.; Gurevych, I. AdapterFusion: Non-destructive task composition for transfer learning. *arXiv* **2020**, arXiv:2005.00247.
28. Li, X.L.; Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv* **2021**, arXiv:2101.00190.
29. Zhang, Z.R.; Tan, C.; Xu, H.; Wang, C.; Huang, J.; Huang, S. Towards Adaptive Prefix Tuning for Parameter-Efficient Language Model Fine-tuning. *arXiv* **2023**, arXiv:2305.15212.
30. Guo, D.; Rush, A.M.; Kim, Y. Parameter-efficient transfer learning with diff pruning. *arXiv* **2020**, arXiv:2012.07463.
31. Liao, B.; Meng, Y.; Monz, C. Parameter-Efficient Fine-Tuning without Introducing New Latency. *arXiv* **2023**, arXiv:2305.16742.
32. Sung, Y.L.; Nair, V.; Raffel, C.A. Training neural networks with fixed sparse masks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24193–24205.
33. Frankle, J.; Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv* **2018**, arXiv:1803.03635.
34. Malach, E.; Yehudai, G.; Shalev-Schwartz, S.; Shamir, O. Proving the lottery ticket hypothesis: Pruning is all you need. In Proceedings of the International Conference on Machine Learning, PMLR, Online, 13–18 July 2020; pp. 6682–6691.
35. Ansell, A.; Ponti, E.M.; Korhonen, A.; Vulić, I. Composable sparse fine-tuning for cross-lingual transfer. *arXiv* **2021**, arXiv:2110.07560.
36. Aghajanyan, A.; Zettlemoyer, L.; Gupta, S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv* **2020**, arXiv:2012.13255.
37. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
38. Valipour, M.; Rezagholizadeh, M.; Kobyzev, I.; Ghodsi, A. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv* **2022**, arXiv:2210.07558.
39. Zhang, Q.; Chen, M.; Bukharin, A.; He, P.; Cheng, Y.; Chen, W.; Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv* **2023**, arXiv:2303.10512.
40. Kopiczko, D.J.; Blankevoort, T.; Asano, Y.M. Vera: Vector-based random matrix adaptation. *arXiv* **2023**, arXiv:2310.11454.
41. 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:2312.10997.

42. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–55. [CrossRef]
43. Lyu, Y.; Li, Z.; Niu, S.; Xiong, F.; Tang, B.; Wang, W.; Wu, H.; Liu, H.; Xu, T.; Chen, E. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–32. [CrossRef]
44. Naganawa, H.; Hirata, E. Enhancing Policy Generation with GraphRAG and YouTube Data: A Logistics Case Study. *Electronics* **2025**, *14*, 1241. [CrossRef]
45. Bao, X.; Lv, Z.; Wu, B. Enhancing Large Language Models with RAG for Visual Language Navigation in Continuous Environments. *Electronics* **2025**, *14*, 909. [CrossRef]
46. Kwon, M.; Bang, J.; Hwang, S.; Jang, J.; Lee, W. A Dynamic-Selection-Based, Retrieval-Augmented Generation Framework: Enhancing Multi-Document Question-Answering for Commercial Applications. *Electronics* **2025**, *14*, 659. [CrossRef]
47. Ma, X.; Gong, Y.; He, P.; Zhao, H.; Duan, N. Query rewriting in retrieval-augmented large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 5303–5315.
48. Wang, X.; Wang, Z.; Gao, X.; Zhang, F.; Wu, Y.; Xu, Z.; Shi, T.; Wang, Z.; Li, S.; Qian, Q.; et al. Searching for best practices in retrieval-augmented generation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; pp. 17716–17736.
49. Peng, W.; Li, G.; Jiang, Y.; Wang, Z.; Ou, D.; Zeng, X.; Xu, D.; Xu, T.; Chen, E. Large language model based long-tail query rewriting in taobao search. In Proceedings of the Companion Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 20–28.
50. Zheng, H.S.; Mishra, S.; Chen, X.; Cheng, H.T.; Chi, E.H.; Le, Q.V.; Zhou, D. Take a step back: Evoking reasoning via abstraction in large language models. *arXiv* **2023**, arXiv:2310.06117.
51. Gao, L.; Ma, X.; Lin, J.; Callan, J. Precise zero-shot dense retrieval without relevance labels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 1762–1777.
52. Wang, X.; Yang, Q.; Qiu, Y.; Liang, J.; He, Q.; Gu, Z.; Xiao, Y.; Wang, W. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv* **2023**, arXiv:2308.11761.
53. Brown, J.D.; Raudaschl, A.; Johnson, D. Anchoring global security: Autonomous shipping with mind reading AI, GPT-core and mamba-core agents, rag-fusion, ai communities, hive-ai, and the human psyche. *Res. Oper. Indaiatuba* **2023**, *10* . [CrossRef]
54. Cheng, X.; Luo, D.; Chen, X.; Liu, L.; Zhao, D.; Yan, R. Lift yourself up: Retrieval-augmented text generation with self-memory. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 43780–43799.
55. Wang, S.; Xu, Y.; Fang, Y.; Liu, Y.; Sun, S.; Xu, R.; Zhu, C.; Zeng, M. Training data is more valuable than you think: A simple and effective method by retrieving from training data. *arXiv* **2022**, arXiv:2203.08773.
56. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
57. He, H.; Cai, J.; Zhang, J.; Tao, D.; Zhuang, B. Sensitivity-aware visual parameter-efficient fine-tuning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 11825–11835.
58. Mokady, R.; Hertz, A.; Bermano, A.H. Clipcap: Clip prefix for image captioning. *arXiv* **2021**, arXiv:2111.09734.
59. Liang, G.; Zhang, B.; Wang, Y.; Li, X.; Ye, Y.; Wang, H.; Luo, C.; Ye, K. Lg-vq: Language-guided codebook learning. *arXiv* **2024**, arXiv:2405.14206.
60. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.

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.