A Comparative Study of Fine-Tuning, Retrieval-Augmented Generation and Hybrid Approaches for Large Language Models

Aditya Dinesh K[1], Sanjay D K[2], Dr. Rakesh Kumar[3]

[1]*Student, Dept. of MSc DS, AIMIT, St Aloysius (Deemed to be University), Mangaluru*
[2]*Student, Dept. of MSc DS, AIMIT, St Aloysius (Deemed to be University), Mangaluru*
[3]*HoD, Dept. of ITs, AIMIT, St Aloysius (Deemed to be University), Mangaluru*

October 30, 2025

Abstract

Large Language Models (LLMs) are used through either by fine-tuning or Retrieval Augmented Generation (RAG) to specific area tasks, but empirically comparing these two approaches within a unified framework is limited. The three adaptation strategies, which are fine-tuning alone, RAG alone, and hybrid approach varying between the three, are performed in this study with the same base model, data, and evaluation configuration. We evaluate performance by cosine similarity scoring with true knowledge generated by a higher-capacity evaluator model, and a qualitative error analysis and practical implementation issues. The findings indicate that domain alignment is greatly enhanced by fine-tuning, factual grounding is enhanced by RAG, and hybrid method is stable and more accurate. The paper concludes by providing the advice of choosing the methods of adaptation depending on computational cost, domain specificity and real world application requirements.

1 Introduction

Large language models (LLMs) are the backbone of understanding and generating natural-language content. They are doing well at such tasks as answering questions, summarizing text, maintaining a dialogue, and even generating domain specific content. But, training them to perform specialized work remains challenging, full fine-tuning of large pre-trained models would consume a ton of GPU memory, and retrieval-based systems also have drawbacks such as increased system complexity and higher latency.

Researchers have developed effective fine-tuning tricks, such as Low-Rank Adaptation (LoRA), in order to address the resource crunch, where the model weights are frozen but a few small adapter parameters are modified. More recently, that was quantized with quantization by Tim Dettmers et al with Quantized Low-Rank Adaptation (QLoRA) and now you can train huge models on even weaker GPUs. Reverse-generation On the reverse side, not only can retrieval-augmented generation (RAG) pipelines be used to reduce full re-training, but they introduce complexity and retrieval costs.

This paper compares three approaches to adapting a specified downstream task: (1) fine-tuning using QLoRA, (2) executing a pure RAG pipeline using the base model, and (3) a hybrid approach that fine-tunes using QLoRA and executes the RAG. In terms of the fine-tuning stage, we utilized a rank-8 adaptation on a consumer-grade NVIDIA GeForce

RTX 3050 6GB card, and it required approximately 10 hours with 10 epochs since the card was resource-bounded.

The goal of running these three setups on the same task (using the same dataset) with similar evaluation metrics is to provide answers to the following questions: Which strategy provides the best trade-off between performance, resource efficiency, inference cost, and real-world deployment practicality? We will make (a) QLoRA fine-tuning on small hardware and (b) a RAG pipeline comparison, (c) empirically testing the hybrid method in a controlled environment. The results must inform practitioners and researchers in particular those dealing with constrained compute to effectively tailor LLMs to domain-specific purposes.

2 Methodology

2.1 Model and Fine-Tuning Setup

In this research, we chose a starting point of the Qwen2.5 0.5 billion-parameter model. Quantized Low-Rank Adaptation (QLoRA) is a reasonably effective method for fine-tuning large models. The main idea is that the weights are compressed and only a small number of adapters are trained. QLoRA allows large language models to be trained using moderate hardware when quantization and low-rank adapters are combined.

Our fine-tuning configuration is as follows:

- **Model:** Qwen2.5 (0.5B parameters)
- **Adapter rank:** 8 (low-rank adapters)
- **Epochs:** 20
- **Hardware:** NVIDIA GeForce RTX 3050 (single-card GPU, 95 Watts)
- **Training time:** Approximately 10 hours (limited by GPU constraints)
- **Batch size:** 6 per device
- **Gradient accumulation steps:** 6
- **Learning rate:** 5e-4

This architecture enabled us to train a domain-adapted model even with minimal hardware through quantization and efficient adapter training. The data was collected and preprocessed accordingly, as detailed in the next section.

2.2 Dataset Preparation

Our dataset preparation involved the following major steps:

1. Source Document Extraction

We used the Docling library to extract text from PDF files relevant to our topic. The documents were transformed into chunks of raw text for subsequent processing.

2. Question-Answer Pair Generation

We constructed question-answer pairs from the extracted text of a PDF documentation. We created a dictionary containing context ID, context text, questions, and answers to ensure traceability. We then generated a JSON file containing question-answer pairs with context IDs. Additionally, we created a simplified version without context IDs to test training approaches, resulting in two datasets for comparative training.

3. Data Cleaning and Quality Improvement

We refined the question-answer JSON to improve overall quality through the following steps: removing irrelevant or off-topic pairs, eliminating redundant punctuation and excessive whitespace, and ensuring each pair was understandable and domain-relevant. We used Ollama models, specifically Gemma 3:4B, to score and filter unnecessary content.

The final cleaned dataset comprised 6,445 question-answer pairs, which were used for fine-tuning.

2.3 Training Pipeline

We preprocessed the information to support causal language modelling (chat-style) in the following way: each sample was placed in a conversation template. The tokenizer of the base model was loaded, and the pad token was configured to the EOS token when necessary and the dataset was converted to this format.

We then loaded the model in 8-bit or quantised mode (`load_in_8bit=True`) to make it memory-efficient, and configured it to undergo gradient checkpointing, and configured it to train with k-bits with the help of `prepare_model_for_kbit_training`.

The training continued until convergence, with an estimated training time of 10 hours on the RTX 3050

2.4 Retrieval and Hybrid Pipeline Configuration

The hybrid and retrieval pipeline systems are typically applied in cases where the source documents are geographically or contextually distant from the end user.

During the fine-tuning process, I constructed a Retrieval-Augmented Generation (RAG) pipeline and a hybrid pipeline (fine-tuned model + retrieval). To enable retrieval, I used an out-of-domain external document corpus, prepared embeddings using sentence transformers or LLM outputs, and stored them in a vector database (ChromaDB), retrieving the top-k documents per query. In these pipelines, the retrieved context was concatenated with the user query and input into the model (base or fine-tuned).

2.5 Evaluation Metrics and Setup

We compared the three pipelines (fine-tuning alone, RAG alone, and hybrid) on the same held-out test set derived from the Q&A data. The evaluation encompassed the following

metrics: accuracy and appropriate generation metrics, analysis of error types (hallucinations, domain errors, generalisation failures), and operational overhead (inference latency, retrieval cost). We also performed qualitative analysis of sample outputs to identify failure modes and understand model behaviour.

How we did it, we added a superior RAG with a bigger model Gemma3:4B, we iterated over 1200 QnA and compared its cosine distance with each of the three Qwen models that we had using cross encoders. The cosine similarity was computed using the standard formula:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|}$$

Then we captured all the scores that were obtained for each model and we did our final evaluation on how it is closer to the tru knowledge.

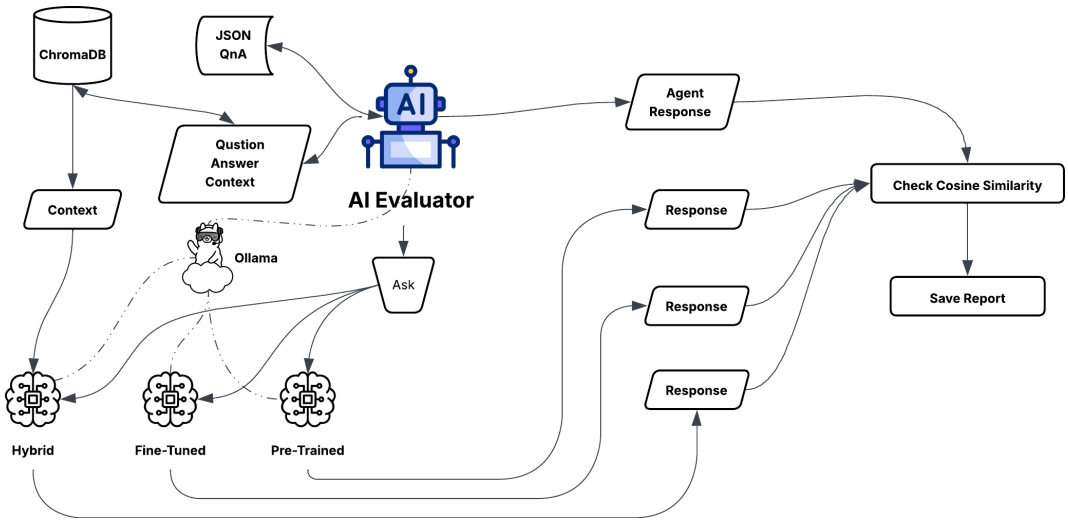

Figure 1: Overview of the evaluation workflow used to compare pre-trained, fine-tuned, and hybrid models using a retrieval-enabled evaluator model.

3 Results

An empirical assessment of three adaptation strategies: fine-tuning and RAG alone and a combination of the two, demonstrated the presence of evident and quantitative differences in their performance. In both models, we computed the score of the cross-encoder encoder of cosine similarity against a higher capacity evaluator (Gemma3:4B).

3.1 1. Overall Performance Comparison

The pre-trained model showed the lowest and worst-performing scores and a cosine similarity mean of only 0.339 as seen in the box plot (Figure 2). In fine-tuning it with QLoRA the distribution shot up, reaching a mean of 0.461 - a 39.26% increase over the baseline.

The best numbers were obtained with addition of RAG to fine-tuning: mean is 0.737, corresponding to 55.89% relative improvement with simple fine-tuning and a greater than 117% improvement with the unmodified pre-trained baseline. It is evident that the two methods carry with them complementary benefits.

Figure 2: Model Accuracy Comparison & Percentage Improvement.

3.2 2. Question-wise Performance Trends

A detailed question-level analysis further confirms the benefit of fine-tuning. Figures 3 and 4 plot cosine similarity scores for approximately 1,200 samples.

- The fine-tuned model slightly outperforms the pre-trained model, showing slightly higher peaks and a more stable results.
- The RAG-only model produces better results than both pre-trained and fine-tuned models but remains inferior to the hybrid version.
- The hybrid model demonstrates the highest similarity to true knowledge and the highest minimum similarity scores, indicating stronger robustness across the variety of questions.

These trends reinforce that fine-tuning enhances domain-alignment, whereas RAG contributes contextual grounding and factual accuracy.

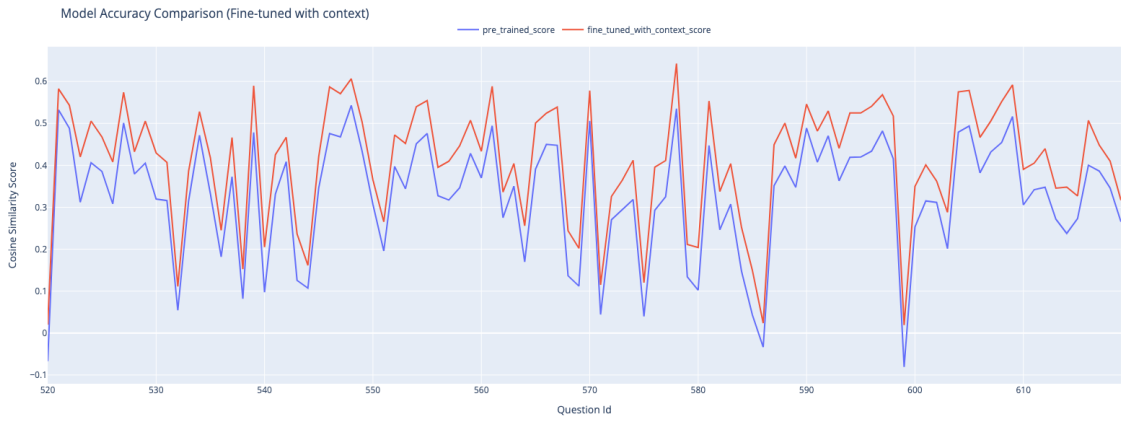

Figure 3: Model Trained with low-quality data, QnA pairs with context.

Figure 4: Model Trained with high-quality data, QnA pairs with without context.

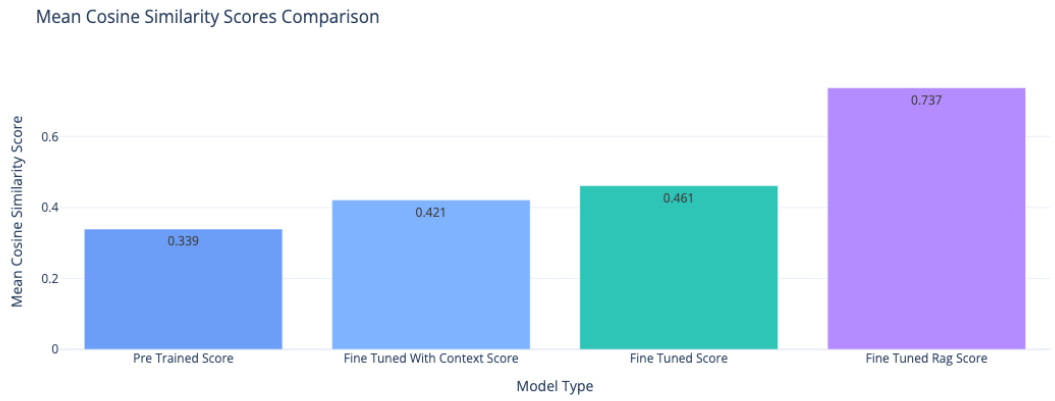

Figure 5: Enter Caption

3.3 3. Score Distribution Analysis

On histogram and KDE density visualisation Figure 6, it can be seen that there is a distinct separation among the three approaches. The distribution of the pre-trained model is

very broad and with lower similarity values centered, whereas the fine-tuning changes the distribution to a mid-range with reduced variation. The hybrid model is a high-density cluster of about 0.75 - 0.85, which means that the hybrid model is much closer to the evaluator model. This is consistent with the previous results: the hybrid adaptation offers both the highest alignment and the best output quality. 4 Discussion and Conclusions

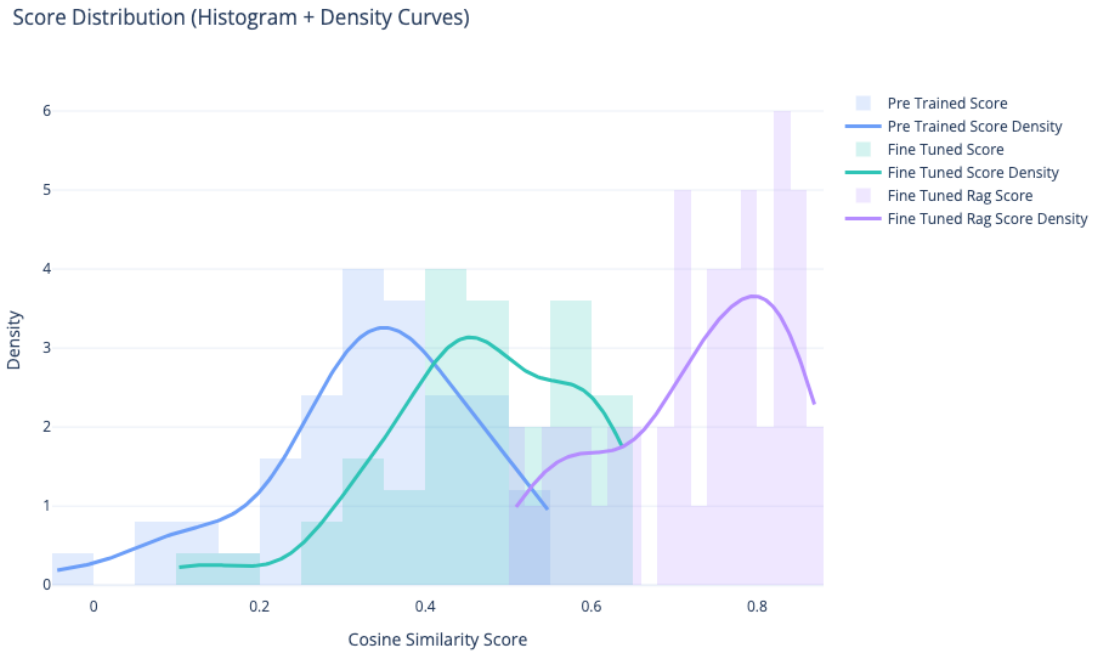

Figure 6: Score distribution histogram and density curve for the three model types.

4 Discussion and Conclusions

This paper presents a comparative assessment of three practical methods of large language model adaptation fine tuning, retrieval augmented generation (RAG), and a combination of the two methodologies. The experimental findings indicate that all the three strategies are more effective in comparison with the base model, but the magnitude and character of the improvements vary significantly.

Fine-tuning provides significant advances in the case of the model adaptation to a specific field. Lightweight QLoRA-based tuning on small hardware is already able to move the model knowledge even closer to domain-aligned knowledge. Nonetheless, there is an apparent computational cost to this. It is costly, time-consuming, and difficult to maintain especially when fine-tuning is carried out on several domains. Furthermore, a fined tuned model is likely to be specialised and thus it will lack flexibility when tasked to do things that it has not been trained with.

RAG, however, provides another way out, by providing external knowledge on a dynamic basis during inference. This prevents retraining, allows multi-domain use and allows the model to make use of the latest information. Individually, RAG already outperforms the pre-trained baseline, and particularly in factual or document related tasks.

The main conclusion is that the hybrid model of a fine-tuning with RAG is always more effective compared to other strategies. The hybrid arrangement is based on the parameter-level adaptation and at the retrieval level, grounding. Fine-tuning limits the errors, which occur due to misunderstanding of the domain and retrieval also maintains the responses in the accurate content related to context. This two-mechanism mechanism has the best accuracy, most constant distribution and the best overall evaluation score.

However, the hybrid approach has the shortcomings of the two approaches: it is costly to compute, more difficult to run, and could be locked into domain lock thus can yield domain lock-in when fine-tuning data is limited. A pure RAG pipeline can still be desirable in multi-domain environments. However, in the cases where the domain specialization is required, such as medical, legal, educational, or technical, the hybrid approach provides the greatest performance returns.

In short, fine-tuning has great power and is expensive; RAG is versatile and effective but is low in profound domain correspondence; and the hybrid model is the quickest and strongest. Fine-tuning with RAG can offer the best results when it is meticulously performed with the help of a high-quality retrieval system and must become the default method of domain-specific LLM usage.

References

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems, 36*, 10088–10115.

Wu, X. K., Chen, M., Li, W., Wang, R., Lu, L., Liu, J., ... & Wang, F. Y. (2025). LLM fine-tuning: Concepts, opportunities, and challenges. *Big Data and Cognitive Computing, 9*(4), 87.

Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E. P., Bing, L., ... & Lee, R. (2023). LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. *Proceedings of EMNLP 2023*, 5254–5276.

Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., & Zhao, T. (2023). LOFTQ: LoRA-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*.

Wang, X., Aitchison, L., & Rudolph, M. (2023). LoRA ensembles for large language model fine-tuning. *arXiv preprint arXiv:2310.00035*.

Patil, S. S., Rathore, A. S., & Ramteke, M. (2024). QLoRA-based fine-tuning of LLMs on multiple medical reasoning tasks. *ISCMI 2024*, 177–181. IEEE.

Chan, C. M., Xu, C., Yuan, R., Luo, H., Xue, W., Guo, Y., & Fu, J. (2024). RQ-RAG: Learning to refine queries for retrieval-augmented generation. *arXiv preprint arXiv:2404.00610*.

Bruckhaus, T. (2024). RAG does not work for enterprises. *arXiv preprint arXiv:2406.04369*.