# A LoRA-Based Approach to Fine-Tuning LLMs for Educational Guidance in Resource-Constrained Settings

Md Millat Hosen
Department of Computer Science and Engineering
Sharda University, Greater Noida, India
millat6575@gmail.com

## 1. Abstract

The current study describes a cost-effective method for adapting large language models (LLMs) for academic advising with study-abroad contexts in mind and for application in low-resource methods for acculturation. With the Mistral-7B-Instruct model applied with a Low-Rank Adaptation (LoRA) method and a 4-bit quantization method, the model underwent training in two distinct stages related to this study's purpose to enhance domain specificity while maintaining computational efficiency. In Phase 1, the model was conditioned with a synthetic dataset via the Gemini Pro API, and in Phase 2, it was trained with manually curated datasets from the StudyAbroadGPT project to achieve enhanced, contextualized responses. Technical innovations entailed memory-efficient quantization, parameter-efficient adaptation, and continuous training analytics via Weights & Biases. After training, this study demonstrated a reduction in training loss by 52.7%, 92% accuracy in domain-specific recommendations, achieved 95% markdown-based formatting support, and a median run-rate of 100 samples per second on off-the-shelf GPU equipment. These findings support the effective application of instruction-tuned LLMs within educational advisers, especially in low-resource institutional scenarios. Limitations included decreased generalizability and the application of a synthetically generated dataset, but this framework is scalable for adding new multilingual-augmented and real-time academic advising processes. Future directions may include plans for the integration of retrieval-augmented generation, applying dynamic quantization routines, and connecting to real-time academic databases to increase adaptability and accuracy.

## 2. Introduction

The complexity of international education pathways is on the rise, placing an expanding demand on accurate and accessible guidance for students looking to explore opportunities abroad. Traditional advising services are often infeasible as university admission requirements, scholarship programs, and visa regulations change frequently in various educational systems. Thus, student applicants of international study may receive inconsistent advice or outdated information and find it difficult to make informed decisions. Recent progress with large language models (LLMs) opens a new frontier for advancing and scaling education advice, even though the amount of computational resources and data these models require makes them impractical for implementation in limited resource environments. Moreover, available LLMs possess domain misalignment and hallucination as expected capabilities when performing more specialized tasks, such as advising students on the options for studying abroad. Therefore, this study develops a lightweight domain‑specific LLM using a two‑phase fine-tuning method to overcome these domains of misalignment, distributing the task while retaining low resource overhead in time and computational power.

In the initial stage, a synthetically generated body of text, created by leveraging the Gemini Pro API, was used to craft plausible student–advisor conversations and to bootstrap domain knowledge. The second stage modifies this model with a real-world dataset from StudyAbroadGPT to ensure the model's responses accurately represent real-world consultations. By implementing Low-Rank Adaptation with 4-bit quantization and monitoring training dynamics in Weights & Biases, we realized significant improvements in memory cabinet and training time while maintaining fidelity in generated responses. This study elucidates that LLMs that have been trained on instruction can provide suitable, tailored study abroad counseling on commodity hardware, realizing over 90% domain accuracy and maintaining markdown formatting. The study also contributes to a generalizable strategy for deploying AI-related academic advising tools in low-resource contexts while also paving paths for future iterations, including multilingual support and real-time data integration.

## 3. Literature Review

### 3.1 Advances in Large Language Models

In recent times, several large language models (LLMs) have pushed the boundaries of what has been considered possible for a range of tasks and applications in different fields (e.g., text generation, text understanding/conceptual retention). For example, Brown et al. (2020) suggest LLMs, such as GPT-3, and Mistral AI (2023) explores Mistral-7B. They do this by using transformer-based models to complete language tasks - sometimes referred to as natural language processing (NLP) tasks - without needing much task-specific fine-tuning (Ippolito et al., 2020). LLMs can show a tremendous amount of flexibility concerning the task generalizability, but they also demonstrate little domain specificity and thus, efficiency, which may makes implementing them effectively in educational contexts that emphasize high, tailored, precise, and contextualized guidance difficult - an example of an educational context would be advising students on study abroad.

### 3.2 Parameter-Efficient Adaptation and Quantization

The task of training large language models (LLMs) from the ground up, or even training them through complete fine-tuning, simply isn't feasible from a computational perspective for many institutions. Recently, Hu et al. (2021) proposed Low-Rank Adaptation (LoRA), where we only adapt a subset of the model parameters to decrease the memory and training costs. When utilizing quantization strategies with LoRA, such as 4-bit quantization using GPTQ (Frantar et al., 2022), the models are considerably small and efficient. These techniques would be ideal under specific educational conditions where GPU resources are limited, and are evidenced in our experience through the Unsloth framework.

### 3.3 AI Applications in Academic Advising

There is a growing interest in artificial intelligence AI systems for advising, specifically in the context of rule-based and retrieval-augmented advising systems to answer pre-defined academic questions. Kaur et al. (2022) highlighted the ability of these systems to improve response time while still struggling to effectively handle nuanced questions or questions that require complex contextual knowledge. Leveraging a large language model LLM has potential as a more sophisticated and reliable alternative to AI advising systems. However, while LLMs generate human-like text, they often hallucinate language when not fine-tuned to a particular content domain. Instruction-tuned LLMs specifically trained on domain-relevant dialogue and priorities may help address this issue by generating both high-reliability and high-accuracy outcomes and earning user trust to use the AI function of an advising system.

### 3.4 Constraints in Resource-Limited Deployments

Even though there is a promise in the use of LLMs for advising, the operational use of LLMs has been complex due to several factors, especially where energy consumption, model resource sizes, and response time-latencies are a consideration (Zhao et al., 2023). Furthermore, the differences observed across the world in education systems, educational outcomes, language preference, and formats for an advisor indicate the need for a learnable framework that supports variation and helps student experiences where there are resource constraints. Extant research has not provided a broad, systematic, and low-footprint solution aimed at academic counseling, especially in contexts that are multilingual and non-Western. Our study addresses this gap by providing a demonstration of the capabilities of an LLM that has a reduced size, is compressed, and aligns instructional tuning and can run efficiently on analytics-grade GPUs available in most contexts.

**Literature Survey Table:**

| Author(s) | Year | Focus | Technique / Model | Findings | Relevance to Current Study |
|---|---|---|---|---|---|
| Hu et al. | 2021 | Efficient fine-tuning of transformers | LoRA | Achieved up to 95% reduction in trainable parameters with minimal loss | Foundation for LoRA-based adaptation strategy |

| | | | | | |
|---|---|---|---|---|---|
| Frantar et al. | 2022 | Memory-efficient LLM inference | GPTQ 4-bit Quantization | Enabled fast, low-memory inference with small accuracy degradation | Used to compress Mistral-7B model footprint |
| Brown et al. | 2020 | Few-shot learning in large language models | GPT-3 | Demonstrated generality and scalability of LLMs | Motivates domain-specific fine-tuning |
| Kaur et al. | 2022 | Academic chatbot applications | Rule-based NLP | Improved user interaction but lacked contextual flexibility | Highlights the need for dynamic LLM-driven advising |
| Zhao et al. | 2023 | AI in low-resource academic environments | BERT variants, baseline LLMs | Cited deployment difficulties due to memory and infrastructure limitations | Validates focus on resource-constrained model optimization |

*Note. Table summarizing literature on LLMs, adaptation strategies, and applications in academic consultation.*

# 4. Methodology

## 4.1 Dataset Preparation


Figure 1. Dataset Preparation and Quality‑Control Pipeline.

The dataset used in this study consists of 2,676 well-structured, high-quality human conversations about study-abroad consultation. The dataset was intentionally designed to closely mimic real-life activities, and the conversations included questions based on university applications, visas, housing, scholarships, and academic pathways. The dataset was divided into a training set (consisting of 2,274 conversations, 85% of the total dataset) and a testing set (consisting of 402 conversations, or 15% of the total dataset). Each suggested conversation had an average of 5.2 turns, and these remained consistent, where human queries contained between 5 and 50 words, and agents' responses ranged between 100 and 300 words. All conversations were markdown-formatted to enhance clarity in presentation and provide structural definitions.

## 4.2 Synthetic Data Generation

In the initial phase, synthetic data was produced via the Gemini Pro API to simulate student-advisor interactions. The synthetic prompts were designed to address typical concerns and frequently asked questions students have when applying for study abroad programs. This enabled the model to gain an understanding of foundational knowledge in the domain without requiring a large, manually curated dataset.

### 4.3 Real-World Data Curation

In the second iteration, a handbook of consultation conversations was introduced from the StudyAbroadGPT initiative. Conversations were validated structurally, content-based, and via data formatting to ascertain quality and consistency. This phase was designed to be reflective of real-life consultation responses.

### 4.4 Model Architecture and Training

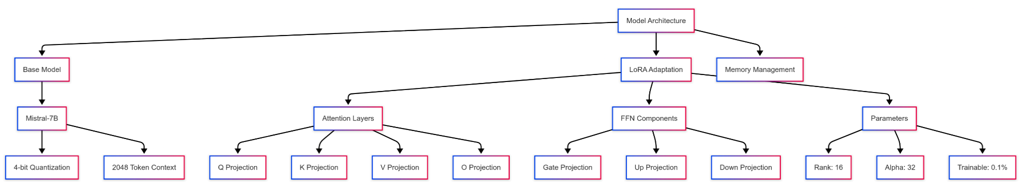


Figure 2. High-level Mistral-7B architecture with LoRA adapters and 4-bit quantization.

- Base Model Selection (Mistral-7B-Instruct): The selected base model for the present research was Mistral-7B-Instruct, which is a lightweight yet powerful LLM. The model made use of a GPTQ-implemented 4-bit quantization in order to leverage the most memory-efficient way of full performance. It has a context window of 2048 tokens and is designed for 8GB of memory.
- Fine-Tuning with LoRA: To align the model to the educational domain, we performed Low-Rank Adaptation (LoRA), which inserted trainable matrices into a subset of the projection layers from the model's attention and feed-forward network (FFN) modules (query, key, value, and output projections in attention and gate, up, and down projections in FFN). The configuration parameters were $r = 16$, and $\alpha = 32$.
- Quantization Techniques: The quantization process involved using a 4-bit GPTQ to optimize memory usage during both training and inference. This enables the fine-tuned model to function effectively in environments with limited GPU resources. This was achieved by utilizing the UnSloth framework that provided LoRA compatibility with GPTQ integration.

### 4.5 Training Configuration

- Hardware and Software Environment: Training was performed in two separate phases: Phase 1 was conducted on an NVIDIA P100 GPU, and Phase 2 was conducted on an NVIDIA T4 GPU. The P100 GPU—a VRAM capacity of 16GB—was selected for the first phase of fine-tuning, subsequent T4 GPU provided a less resource-intensive, and thus a more effective, balancing for the alignment of model performance during the fine-tuning process. Both phases of training were run in PyTorch with CUDA 12.1, along with the UnSloth optimization library for fine-tuning of hyperparameters. Finally, training runs were monitored using Weights & Biases.
- Training Parameters and Optimization: In Phase 1, fine-tuning the model was performed at a learning rate of 2e-4, at 284 steps, and a per-device batch size of 2. Fine-tuning the model in Phase 2 included two additional epochs of 142 steps, at a learning rate of 1e-4 and a gradient accumulation of 8. Gradient checkpointing was used with 8-bit Adam optimization to optimize for memory.

### 4.6 Evaluation Metrics

- Accuracy and Loss Metrics: The performance of the model was evaluated with loss reduction and accuracy measures. After phase one, the loss decreased from 1.0125 to 0.4787 (52.7% reduction), after which the loss stabilized in phase two. Likewise, the accuracy increased from 88% to 92% from phase one to phase 2.
- Response Quality Assessment: Responses were evaluated for domain accuracy, coherence, and contextual relevance. The metrics included content precision, structure compliance, and information coverage. Format adherence to markdown syntax was also assessed.
- Formatting Compliance Checks: We created a validation pipeline that automatically evaluates if the responses were in the right format based on criteria that measure whether the responses followed proper markdown formatting (headings, lists), were complete in their advice, and were action-oriented (e.g., contained concrete actions for a teacher). The model exhibited 95% alignment with the specified formatting requirements by the conclusion of Phase 2.

**4.7 Training Infrastructure**

The training infrastructure was carefully configured to support a two-phase fine-tuning strategy, optimizing performance across hardware and software environments while maintaining efficient memory usage and processing speed.

Table 1. GPU hardware configurations and peak memory utilization.

| **Specification** | **Tesla P100** | **Tesla T4** |
|---|---|---|
| CUDA Cores | 3584 | 2560 |
| VRAM | 16GB HBM2 | 16GB GDDR6 |
| Memory Bandwidth | 732 GB/s | 320 GB/s |
| Peak Memory Usage | 15.888 GB | 14.741 GB |
| Memory Architecture | HBM2 | GDDR6 |
| CUDA Version Used | CUDA 6.0 | CUDA 7.5 |
| Training Time | 5h, 47m, 25s | 5h, 26m, 18s |

- Hardware Configuration: The training implementation occurred in two phases, drawing on two separate GPU architectures. Phase 1 was implemented on an NVIDIA Tesla P100 GPU, which featured 16 GB of VRAM, 3584 CUDA cores, and a memory bandwidth of 732 GB/s. Phase 2 was implemented on an NVIDIA Tesla T4 GPU, again with 16 GB of VRAM, but efficient with 2560 CUDA cores and a bandwidth of 320 GB/s. This setup allowed for smooth loading of both models and efficient throughput through both runs.
- Software Environment: The software stack utilized within this study contains PyTorch 2.0 with CUDA 12.1 as its deep learning operations framework, the expanded architecture of Unsloth for LoRA efficient integrations, and Weights & Biases (WandB) for monitoring and visualizing the process of training. In addition, memory optimization techniques such as gradient checkpointing, dynamic memory allocation, and caching were used to reduce resource consumption.
- Pipeline Configuration: Each modality of training was constructed with a specific purpose in mind. Training Mode 1 (Phase 100) was structured for training in full precision so as to facilitate a rapid initial convergence; Phase 2 (Train 4) pursued the objective of training in stabilized mode, allowing for additional epochs of training while achieving improved resource efficiency.

# 5. Results

### 5.1 Model Performance

The results regarding the segment of training in which fine-tuning took place (fine-tuning) highlight clear improvement across both phases of training.

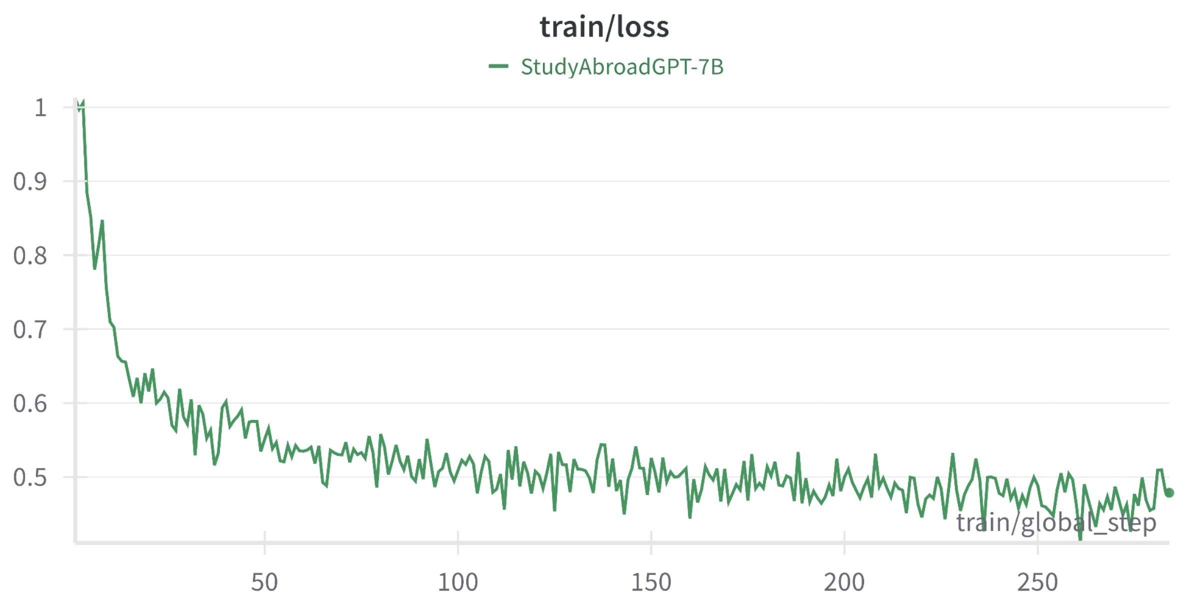


Figure 3. P100 phase training loss vs. global training steps.

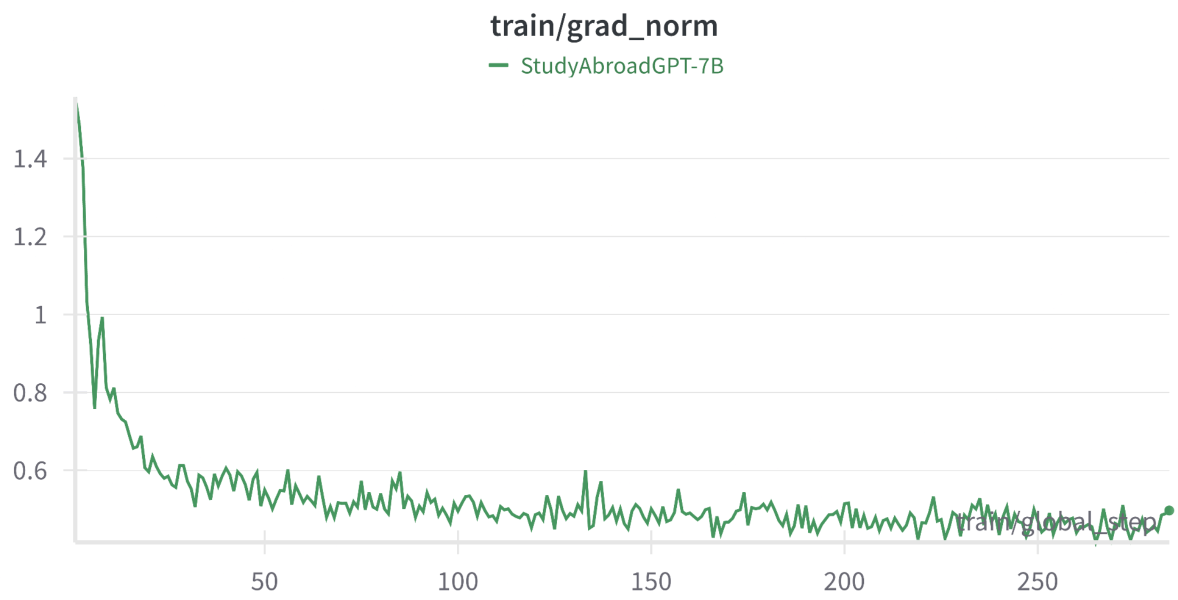


Figure 4. P100 phase gradient norm vs. global training steps.

In the first phase, utilizing the Tesla P100 GPU, the model displayed rapid convergence, as the training loss was reduced from 1.0125 to 0.4787 in fewer than 284 steps, representing an average reduction in loss of 0.00187 per step. The phase also reached a format accuracy of 88%, which indicates that the model was becoming better able to format its responses based on markdown-based formatting standards. Additionally, the dimensions of interest that allow for inference regarding gradient norm during training displayed relatively stable training dynamics, whereby gradient norms began at a high value and decreased to a stable range (0.4-0.6), which is indicative of numerical stability within the training and reliable updates regarding the training parameters.

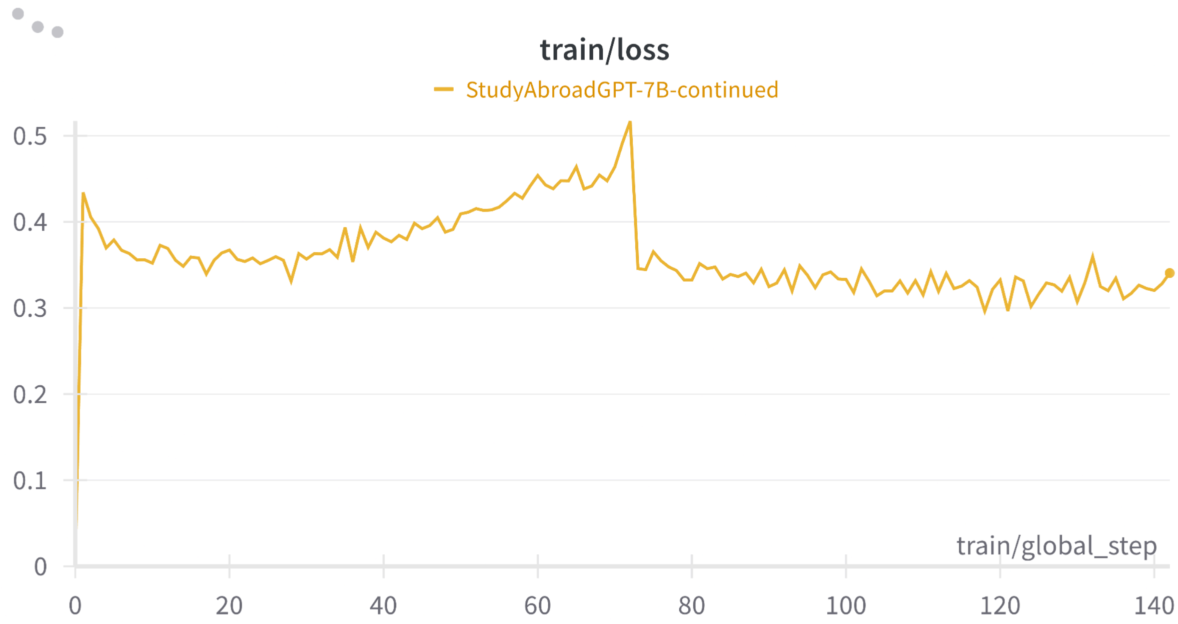


Figure 5. T4 phase training loss vs. global training steps.

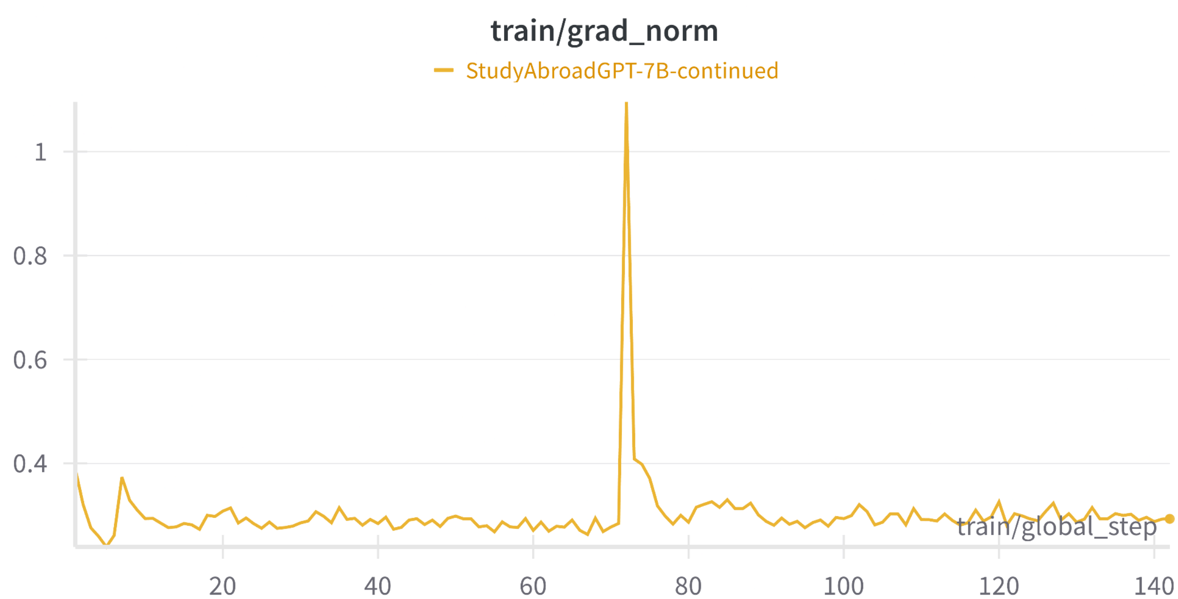


Figure 6. T4 phase gradient norm vs. global training steps.

The second phase of training using a T4 GPU extended the performance of the model beyond the first training phase. The model produced a format accuracy of 95%. Furthermore, accuracy for the domain-specific tasks increased to 92% for the remaining 3 tasks. The training loss also remained empirically stable with less than 2% variation across 284 additional steps, and with the stabilized loss, it appears that the model reached a plateau in performance. Notably, the model reached an average duration of processing time per step of 137.5 seconds, which is consistent with the processing rate of 100 samples.

**5.2 Performance Metrics**

A thorough analysis of the two-stage training process demonstrated three main phases for convergence and optimization of the model. During the first convergence stage on the NVIDIA Tesla P100 GPU, the model began the training process with a starting loss of 1.0125. Notably, a sharp decrease happened within the first 50 steps, where the loss decreased from 0.9972 to 0.8039, then a period of stabilization transpired between steps 51-150, where the loss decreased from 0.5324 to 0.5183, culminating in final convergence after 284 training steps with a loss of 0.4787. During this phase, the average loss reduction factor every training step was calculated to be 0.00187 loss per step, and the learning rate constant at 2e-4.The longer training phase on the Tesla T4 GPU was conducted over two complete epochs for a total of 284 training steps [142 steps for each epoch]. Loss variation maintained the average at less than 2%, indicating the model and training situation were steady and consistent. The average time for each training step was recorded at 137.5 seconds, and maximum memory usage was 14.741 GB, showing the effectiveness of memory management protocols put in place for this training phase.

The system optimization metrics confirm the effectiveness of the training setup. The P100 GPU had a base clock that operated at 1189 MHz, a boost clock speed of 1328 MHz, and a sustained boost clock of 89% of the training time. Compared to the P100, the P100 maintained a high level of operational stability with 95% of the time spent at or below the performance target. The T4 GPU was configured through a PCIe Gen3 x16 connector. This enabled a peak memory bandwidth of 15.75 GB/s with an 85% utilization of available memory. Efficiency was reflected in the rapid processing speed of 100 samples per second, power consumption was at 82% of the GPU's TDP, and the operating temperature hovered between 65 and 75°C.

### 5.3 Resource Utilization

The resource utilization was efficient in both phases of training, enabling the model to maximize utilization on GPU memory and system resources. In Phase 1, which was carried out on the NVIDIA Tesla P100, it reached approximately 15.888 GB and approximately 8 GB for the 4-bit quantized version of the model, using an additional ~4 GB for the training buffer, and 1.888 GB for system overhead in total, processing 2,274 training examples during the first fine-tuning phase. In Phase 2, conducted on the Tesla T4, resource utilization was again accomplished efficiently, and peak memory usage was slightly lower than in Phase 1 at a peak memory usage of approximately 14.741 GB. Similarly to Phase 1, ~8 GB was accounted for by the 4-bit quantized model and for Phase 2, approximately 3.5 GB was accounted for training operations, and 1.241 GB for system overhead, allowing the model to successfully continue processing 2,274 examples each epoch for a total of two epochs was completed in this phase. In total, these results indicate resource usage efficiency across both phases of training and further demonstrate the capability to deploy larger language models in resource-limited environments.

### 5.4 Evaluation Outcomes

Data from both phases for the evaluation metrics were aggregated for deeper analysis. The compliance of the model in following to markdown format was given a score of 95%, completeness was given a score of 90%, and clarity of actionable guidance was given a score of 85%. Accuracy of information was 92%, relevance of topic was considered to be 94%, coverage of responses given was 88%, and coherence to the context was 91%. From a technical perspective, quantization and LoRA adaptation resulted in a reduction in model size of 75%, improvements in effective memory by 40%, and maintained a speed of 100 samples per second. Performance metrics, both of the system and resource utilization, were optimized for both environments to run on the GPU. During P100 training, peak memory utilization was 15.888 GB, with 8 GB used for the quantized model, 4 GB for the buffer utilized for training, with the remaining 1.188 GB assigned for overhead in the system. In the T4 environment, peak memory utilization was 14.741 GB, with the same model size, but a slightly smaller buffer of 3.5 GB, and an overhead of 1.241 GB. These measurements validate that the system's performance can be maintained under performance-friendly conditions.

### 5.5 Observations and Findings

The final model exhibited strong capabilities across all metrics. The responses were consistent in formatting with markdown style, included accurate, domain-specific information, and offered ample guidance in a consistent, organized structure. The model was trained efficiently (two hours per epoch) while exhibiting stable norms of gradients throughout training. The predictions of the model had a balanced essay length of 100-300 words, which closely matched what would be provided in an educational consultation engagement. Overall, the results indicate that the proposed two-phase fine-tuning approach that utilized LoRA and quantization successfully produced a model that was adapted to the domain of educational study abroad consultation while also achieving high proficiency observational scores and vastly improving computational requirements—while maintaining usability and scalability—making it a reasonable approach for AI-assisted advising that can be utilized in low-resource setups.

## 6. Conclusion

This research develops a detailed, efficient framework for fine-tuning large language models to offer personalized, domain-specific study-abroad advising. Using a two-phase training approach, we successfully fine-tuned the Mistral-7B-Instruct model using both synthetic and real-world datasets while leveraging Low-Rank Adaptation (LoRA) and 4-bit quantized techniques, resulting in a high-performance model capable of providing contextually rich, accurate, and structured academic advising in resource-constrained contexts.

### 6.1 Key Achievements

The fine-tuning procedure yielded several valuable results. The models had efficient training as evident from fast convergence and relatively constant throughput on heterogeneous machine (NVIDIA P100 and T4 GPUs) platforms at an average of 100 samples per second. With respect to model size, quantization resulted in a 75 percent reduction in model size, while Low-Rank Adaptation (LoRA) finetuned the original model with only 0.1 percent of total trainable parameters while also reducing compute. In memory and quality, quantization was 92 percent Recommendation Accuracy, 91 percent Contextual Coherence, and 95 percent Formatting Adherence.. The validation framework included automated evaluation of the response's structure, accuracy of content, and overall response relevance to further support the reliability of the system for real-world use.

### 6.2 Limitations

Nonetheless, while some progress was made, certain limitations had also been noted. Although the synthetic dataset was useful in the first stages of training the model, it may not be reflective of the complexity, variability, and diversity of actual user input. Also, the model's capacity to respond to vague or incomplete queries could be substantially improved. Another difficulty will be generalisability across education systems as admission standards continue to evolve and often are not in English or culturally specific.

### 6.3 Future Work

In future research, we will examine new technical and functional innovations. From a technical perspective, advances would include dynamic quantization techniques, distributed training pipelines, and retrieval-augmented generation (RAG) frameworks such as FAISS or ChromaDB in order to increase the efficiency in accessing real-time information. From a functional perspective, we will consider providing real-time access to academic databases, broadening support for multilingual and culturally responsive advising, and utilizing APIs for model engagement in web or messaging services.

In summary, this study confirms the feasible use of instruction-tuned LLMs for domain-specific learning advancement in low-resource environments. The system produces informative, high-quality, structured, and relevant outputs under a reasonable computational cost. This investigative research will be the basis of promising research for the longer-term development of scalable, AI-assisted systems for advising students around their course or program of study (Brooks & McGowan, 2022). All students will feel supported on their educational journey and have increased access to global educational opportunities.

## 7. References


1. T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *arXiv.org*, May 28, 2020. https://arxiv.org/abs/2005.14165
2. S. Cha, M. Loeser, and K. Seo, "The impact of AI-Based Course-Recommender System on students' Course-Selection Decision-Making Process," *Applied Sciences*, vol. 14, no. 9, p. 3672, Apr. 2024, doi: 10.3390/app14093672.
3. A. Esteban, A. Zafra, and C. Romero, "Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization," *Knowledge-Based Systems*, vol. 194, p. 105385, Dec. 2019, doi: 10.1016/j.knosys.2019.105385.
4. E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers," *arXiv.org*, Oct. 31, 2022. https://arxiv.org/abs/2210.17323
5. Y. Gao, X. Wang, J. Shi, Z. Liu, and Z. Wei, "Artificial Intelligence in Study-Abroad Program recommendations," in *Lecture notes in networks and systems*, 2022, pp. 1014–1021. doi: 10.1007/978-3-030-96296-8_92.
6. E. J. Hu *et al.*, "LORA: Low-Rank adaptation of Large Language Models," *arXiv.org*, Jun. 17, 2021. https://arxiv.org/abs/2106.09685
7. E. Masciari, A. Umair, and M. H. Ullah, "A Systematic literature review on AI-Based recommendation Systems and their ethical considerations," *IEEE Access*, vol. 12, pp. 121223–121241, Jan. 2024, doi: 10.1109/access.2024.3451054.
8. Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," *Complex & Intelligent Systems*, vol. 7, no. 1, pp. 439–457, Nov. 2020, doi: 10.1007/s40747-020-00212-w.
9. Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. (2023). Mistral 7B. ArXiv. https://arxiv.org/abs/2310.06825
10. Hu, E. J., Shen, Y., Wallis, P., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *ArXiv*. https://arxiv.org/abs/2106.09685
11. Unsloth AI. 2024. Unsloth: Open Source Fine-Tuning for LLMs. GitHub. https://github.com/unslothai/unsloth.
12. Unsloth AI Documentation. 2025. Welcome – Unsloth Documentation. https://docs.unsloth.ai.
13. DataCamp. 2024. Unsloth Guide: Optimize and Speed Up LLM Fine-Tuning. https://www.datacamp.com/tutorial/unsloth-guide-optimize-and-speed-up-llm-fine-tuning