Optimizing Mid-Sized Open-Weight Large Language Models: Fine-Tuning Strategies for Cost-Efficient AI Solutions

Nur-Adib Maspo[1*], Ben DAIA Mounir[2], Henrik Eriksson[3]

[1,2]*Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Selangor, Malaysia*

[3]*Cention, Sweden*

[1]*nuradibmaspo@iium.edu.my*
[2]*bendaiamounirads@gmail.com*
[3]*henrik.eriksson@cention.com*

Article history

Received:
7 April 2026

Received in revised form:
20 April 2026

Accepted:
2 May 2026

Published online:
15 June 2026

*Corresponding author
nuradibmaspo@iium.edu.my

Abstract

Large Language Models (LLMs) hold strong potential for enterprise customer service automation, particularly for schema-driven and multi-turn dialogue tasks. Yet, real-world adoption requires high accuracy while operating under strict computational and memory constraints. This study evaluates Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRA) as Parameter-Efficient Fine-Tuning (PEFT) methods for enhancing the LLaMA-8B model in structured automation involving function-call classification and JSON interactions. Experiments compared LoRA, QLoRA, and a zero-shot baseline across function-call accuracy, JSON validity, slot extraction, inference speed, and memory usage. LoRA achieved the best balance, with 84.62% F1-score in function-call accuracy and 95.38% compliance with JSON/schema guidelines. QLoRA reached perfect JSON validity (100%) but performed poorly in recognizing functions, recalling only 6.15%, making it overly conservative for real-world deployment. Both methods struggled with escalation intent detection. Findings highlight the e-off between accuracy and quantization efficiency in PEFT approaches. While QLoRA offers clear benefits in memory and latency, LoRA provides more reliable accuracy with modest resource overhead. This work offers practical guidance for selecting PEFT methods, identifying LoRA as a suitable option for deploying LLMs in structured enterprise customer service automation under resource constraints.

.

Keywords: *Large Language Models, Natural Language Processing, Parameter-Efficient Fine-Tuning (PEFT), Low-Rank Adaptation (LoRA), Quantized Low-Rank Adaptation (QLoRA)*

1. Introduction

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) by enabling systems to generate, interpret, and reason with human-like text across diverse domains. With hundreds of billions of parameters, models such as OpenAI's GPT-3 and GPT-4, Google's PaLM, and Meta's LLaMA have demonstrated advanced abilities in contextual comprehension, reasoning, and creative text generation [3]. Despite these achievements, the practical deployment

* Corresponding author. *nuradibmaspo@iium.edu.my*

of such large-scale models remains challenging due to high computational costs, extensive memory requirements, and limited scalability [2]. For many organizations—particularly small and medium-sized enterprises (SMEs)—these constraints make the adoption of large proprietary models impractical. To bridge this gap, mid-sized open-weight models have emerged as a viable alternative. Models such as LLaMA 3-8B [3] provides a balance between accessibility and performance, offering organizations and individual developers the opportunity to adopt advanced language capabilities without prohibitive resource demands. However, while such models are more affordable and efficient, they often underperform in structured automation tasks. For enterprise environments, where applications frequently rely on function calls, schema adherence, and machine-readable outputs to interact with APIs, this performance gap becomes a critical barrier to adoption.

Parameter-Efficient Fine-Tuning (PEFT) has been proposed as a solution, enabling models to be specialized for domain-specific tasks with significantly lower resource requirements compared to full fine-tuning [1] [16] [17]. Among these techniques, Low-Rank Adaptation (LoRA) [10] and Quantized LoRA (QLoRA) [11] are widely recognized for their efficiency and scalability. Yet, empirical studies that systematically evaluate their effectiveness in structured enterprise tasks remain scarce. This study addresses that gap by evaluating LoRA and QLoRA on the LLaMA 3-8B model for structured automation tasks, with a particular focus on function-call accuracy, schema compliance, latency, and memory efficiency.

The contributions of this paper are threefold: (i) an empirical evaluation of LoRA and QLoRA in enhancing mid-sized open-weight LLMs for structured function-calling tasks; (ii) benchmarking of trade-offs between accuracy, latency, and memory efficiency; and (iii) practical guidance for enterprises and developers, highlighting the suitability of LoRA as a reliable strategy for balancing performance and efficiency in production environments.

2. Literature Review

2.1 Understanding Large Language Models:

LLMs are large-scale neural architectures trained on vast text corpora to perform a variety of sophisticated language tasks. Their scaling into the hundreds of billions of parameters has driven state-of-the-art performance in areas such as reasoning, dialogue, and structured text generation. However, training and deploying such models requires expensive GPUs or TPUs, significant memory resources, and complex infrastructure [5] [9]. For many enterprises, these requirements impose financial and technical barriers, making large-scale LLM adoption impractical. This has motivated research into model compression and fine-tuning techniques to make LLMs more accessible.

2.2. Model Compression Techniques

Numerous model compression techniques have been produced to diminish size and computational requirements of LLMs without drastically compromising accuracy.

a. Quantization reduces precision in parameters and activations (e.g., from FP32 to 8-bit or 4-bit), lowering memory usage and accelerating inference [13][14].
b. Pruning eliminates redundant or non-contributory parameters, resulting in smaller architectures with faster inference [12].
c. Knowledge Distillation transfers knowledge from a large "teacher" model to a smaller "student" model, enabling lightweight deployment [15][16].
d. Weight Sharing reuses the same parameters across multiple layers, reducing the overall number of trainable weights.

While these approaches improve efficiency, they primarily target general model size reduction and do not directly address task-specific structured automation, which is critical for enterprise adoption.

2.3. Parameter-Efficient Fine-Tuning (PEFT)

PEFT represents a complementary approach, where a pre-trained model is adapted to downstream tasks by updating only a small subset of parameters [16]. This method significantly reduces resource demands while maintaining competitive accuracy, making it highly relevant for enterprises operating under resource constraints. Popular PEFT methods include:

a. **LoRA**, which introduces low-rank trainable matrices into transformer layers [4].
b. **QLoRA,** which combines 4-bit quantization with LoRA to enable fine-tuning of large models on consumer-grade GPUs [8].
c. **Adapter Tuning,** which adds small adapter modules between transformer layers [6].
d. Prompt Tuning, which learns trainable prompt embeddings appended to inputs [5].
e. BitFit, which fine-tunes only bias terms in transformer layers [7].

2.4 Comparative Analysis of PEFT Methods

Table 1 summarizes the main PEFT techniques, highlighting their trainable parameter counts, strengths, and limitations. LoRA strikes a balance by significantly reducing trainable parameters (~0.1%–1%) while maintaining accuracy without inference overhead. Prompt Tuning is extremely lightweight (~0.01%) but often underperforms on tasks requiring deep model adaptation. Adapter Tuning (1%–5%) is modular and supports multi-task learning but introduces additional inference latency. BitFit (<0.1%) is simple and effective in some scenarios but lacks the capacity to capture complex task-specific patterns.

Table 1. Comparison of prominent PEFT Methods

Authors	Method	Trainable Parameters	Strengths	Limitations
[4]	LoRA	~0.1%–1%	Significantly reduces trainable parameters, maintains or improves performance in comparison to full fine-tuning, and introduces no additional inference latency.	May not be optimal for all tasks and requires careful selection of rank and scaling factors.
[5]	Prompt Tuning	~0.01%	Extremely parameter-efficient, simple to implement, and effective with large models.	May underperform on complex tasks requiring deep model adaptation.
[6]	Adapter Tuning	~1%–5%	Modular and extensible, allows for task-specific adaptation without affecting the base model, and facilitates multi-task learning.	May require careful integration and introduces additional latency during inference due to added layers.
[7]	BitFit	<0.1%	Extremely lightweight, easy to implement, and surprisingly effective for certain tasks.	May not capture complex task-specific patterns, performance varies across tasks, and limited capacity.

This comparison highlights that while all methods reduce resource costs, LoRA offers the strongest balance between efficiency and adaptability. QLoRA extends this by enabling fine-tuning under memory constraints, achieving near full-precision performance on large models using 4-bit quantization [8].

Although PEFT methods have proven effective across various NLP benchmarks, most evaluations focus on classification or text generation tasks. Relatively few studies examine their role in structured enterprise automation, where accuracy in function calls, schema adherence, and machine-readable outputs is essential. Prompt Tuning and BitFit, while efficient, lack the expressive capacity for such structured tasks. Adapter Tuning, though modular, introduces latency that may hinder enterprise deployments. In contrast, LoRA and QLoRA provide efficiency without significant performance loss, making them promising candidates for resource-constrained business applications. Yet, empirical evidence on their performance in structured automation tasks, particularly in mid-sized open-weight models such as LLaMA 3-8B remains limited. This gap motivates the present study, which systematically evaluates LoRA and QLoRA in terms of accuracy, schema compliance, latency, and memory efficiency, providing guidance on parameter-efficient strategies best suited for enterprise customer service automation.

3. Methodology

The core of this study is built upon a proprietary customer-support dialogue dataset provided by Cention. The research methodology is structured into sequential

stages, as illustrated in Figure 1, with further details discussed in the following sections.

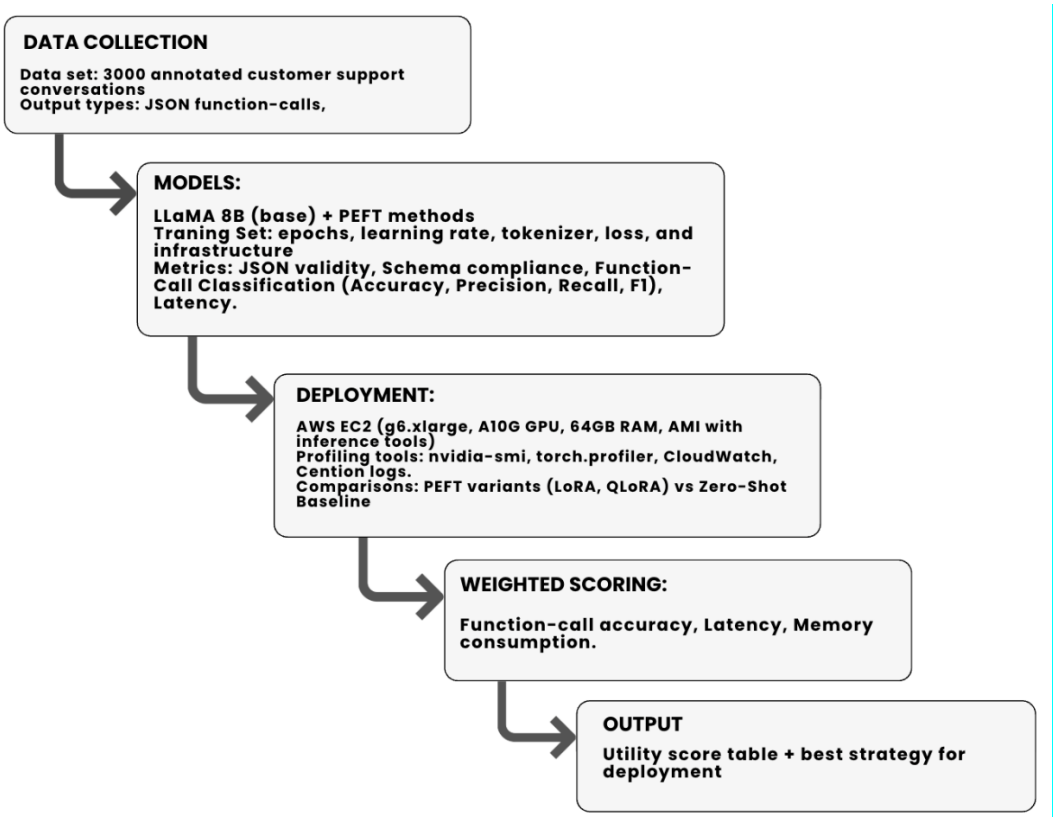

Figure 1. Research workflow and activities

3.1. Dataset

The dataset employed in this study consisted of 3,000 annotated customer support conversations provided by Cention. Each conversation included structured labels in the form of JSON function calls, free-form text, and escalation flags. This ensured the dataset captured both natural conversational dynamics and structured outputs essential for enterprise automation. The annotated data served as the benchmark for evaluating model accuracy, schema compliance, and escalation detection. A custom Python script was used for data preprocessing to convert the raw data into a structured format required for fine-tuning. The data was formatted for "completion" to teach models how to produce set outputs based on input prompts. The fine-tuning setup for LoRA and QLoRA was defined using YAML files. Input-output format structure is presented as per the following sample in table 2.

Table 2. Input-output format structure for customer support conversations

Input prompt structure	Target Output Completion (JSON)

[INST]
Check the status of order id 12345
[/INST]
	{
	"function_call": {
	"name": "get_order_status",
	"parameters": {
	"order_id": "12345"
	}

	} }

[INST]
Send an email to John about the meeting tomorrow
[/INST]
	{
	"function_call": {
	"name": "send_email",
	"parameters": {
	"recipient": "John",
	"subject": "meeting",
	"time": "tomorrow"
	}
	}
	}

3.2 Models

Experiments were conducted using the LLaMA 8B base model fine-tuned with Parameter-Efficient Fine-Tuning (PEFT) methods, specifically Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRA). Both methods were selected due to their proven efficiency in adapting large models to task-specific requirements without incurring the full costs of end-to-end fine-tuning.

3.3 Training Setup

To ensure a fair comparison between LoRA and QLoRA, all training conditions were standardized. The models were trained with the same number of epochs which is 3–5 epochs, with early stopping based on validation structure accuracy, learning rate 1e-4 with linear decay schedule and warmup, batch size is 16 (with gradient accumulation simulating an effective batch size of 64), tokenizer and loss function. Identical infrastructure was used across experiments to eliminate variability from hardware or system-level differences. This setup provided a controlled environment for benchmarking both methods. For the LLaMA 8B model to perform well on function-calling tasks, two PEFT strategies were put into practice.

3.4 Evaluation Metrics

Model evaluation was conducted using a multi-dimensional framework that reflected both predictive quality and deployment feasibility. Key metrics included JSON validity, schema compliance, function-call classification (accuracy, precision, recall, and F1 score), field-level slot extraction (F1 score), inference latency, memory usage, and escalation detection. This comprehensive approach ensured that results captured both technical performance and operational efficiency.

3.5 Deployment Environment

The fine-tuned models were deployed on AWS EC2 g6.xlarge instances equipped with NVIDIA A10G GPUs and 64 GB RAM. Amazon Machine Images (AMIs) preloaded with inference tools provided a consistent runtime environment. Profiling was conducted using nvidia-smi, torch.profiler, AWS CloudWatch, and Cention's internal logging tools. Collected metrics included inference latency, peak VRAM and RAM utilization, parameter overhead compared with the LLaMA 8B base, and disk storage requirements.

3.6 Comparative Benchmarking

Performance comparisons were made between LoRA, QLoRA, and the zero-shot LLaMA 8B baseline. This allowed the study to assess the improvements achieved through fine-tuning and to quantify trade-offs between accuracy and parameter efficiency.

3.7 Weighted Scoring and Integration Review

A weighted scoring system was applied to rank models across function-call accuracy, latency, memory consumption, and ease of integration. Integration complexity was further assessed by examining lines of code required, compatibility with Hugging Face PEFT APIs, and fit within Cention's microservice-based infrastructure. This dual evaluation ensured the methodology addressed both technical performance and real-world deployment considerations.

4. Results and Discussion

This section illustrates the experimental results of fine-tuning the LLaMA 8B model using Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRA) for structured function-calling tasks. The evaluation was conducted on a dataset of 3,500 function-call prompts derived from a combination of open customer service dialogues and custom-generated scenarios designed to capture complex enterprise use cases. A zero-shot baseline test of the LLaMA 8B model revealed 0.00% accuracy and F1 score, confirming the model's inability to perform function-calling tasks without fine-tuning.

4.1 Function Call Classification Performance

4.1.1 LoRA Results

Fine-tuning with LoRA substantially improved function-call performance. The model achieved an accuracy of 84.62%, with a corresponding F1 score of 0.9167, indicating reliable classification across diverse prompts. Furthermore, LoRA maintained a high JSON validity rate of 95.38%, underscoring its ability to produce outputs that conform to schema requirements. These results demonstrate that LoRA effectively adapts mid-sized LLMs to structured automation tasks, striking a strong balance between accuracy and compliance.

4.1.2 QLoRA Results

By contrast, QLoRA exhibited more conservative behavior. While it achieved perfect precision (1.0000) in predicting function calls, its recall was markedly low (6.15%), leading to an overall accuracy of just 6.15%. This suggests that QLoRA was overly cautious, frequently abstaining from predictions unless highly confident. Although such behavior minimizes false positives, it renders QLoRA unsuitable for enterprise applications requiring consistent function-call detection.

4.2 Validity and Schema Compliance

Both fine-tuned models demonstrated strong compliance in generating syntactically valid JSON outputs. QLoRA achieved perfect schema adherence

(100%), while LoRA followed closely with 95.38% compliance. These findings suggest that even under reduced precision, QLoRA retains strong structural reliability, although its function-recognition weakness limits practical applicability.

4.3 Field-Level Slot Extraction

Field-level slot extraction is critical for enterprise applications requiring precise mapping of inputs to structured outputs. LoRA outperformed QLoRA in this aspect, achieving an F1 score of 91.67% compared to QLoRA's 89.26%. While both methods demonstrated strong slot-filling ability, LoRA's superior recall contributed to its higher overall performance, reinforcing its suitability for customer service workflows where comprehensive extraction is essential.

4.4 Latency and Efficiency

In terms of inference efficiency, QLoRA demonstrated slightly lower latency (3.8 s) compared to LoRA (4.3 s) per response. However, the difference of 0.5 seconds is unlikely to be significant in enterprise deployments, where response times within 5 seconds are typically acceptable. Both methods therefore meet practical latency requirements, with QLoRA offering only marginal efficiency gains. A summary of the comparative performance across all evaluation metrics is presented in Table3, highlighting LoRA's superior capability in practical deployment scenarios despite QLoRA's strengths in structural reliability and slightly lower latency

Table 3. Summarizes the performance trade-offs.

Metric/ Evaluation Aspect	LoRA Fine-tuned LLaMA-8B	QLoRA Fine-tuned LLaMA-8B	Interpretation / Implication
Function-Call Accuracy	84.62%	6.15%	LoRA achieves strong classification performance; QLoRA rarely predicts function calls.
Precision	0.88*	1.00	QLoRA avoids false positives but at the cost of missing detections.
Recall	0.96	0.06	LoRA captures most valid function calls; QLoRA fails to trigger reliably.
F1 Score	0.9167	0.1159	LoRA balances precision/recall well; QLoRA collapses due to low recall.
JSON Validity	95.38%	100%	Both produce valid structured outputs; QLoRA is slightly stricter in schema adherence.
Slot Extraction – F1 Score	91.67%	89.26%	LoRA provides more complete field-level information extraction.
Inference Latency	4.3 s	3.8 s	QLoRA is marginally faster; difference not impactful in enterprise SLAs.

Additionally, figure 2 summarizes the comparative performance of LoRA, QLoRA, and the baseline across all evaluation metrics, clearly illustrating the trade-offs between accuracy and efficiency. LoRA demonstrates balanced improvements in accuracy, F1 score, recall, and slot extraction, while QLoRA achieves perfect precision and JSON validity but underperforms in recall and overall accuracy

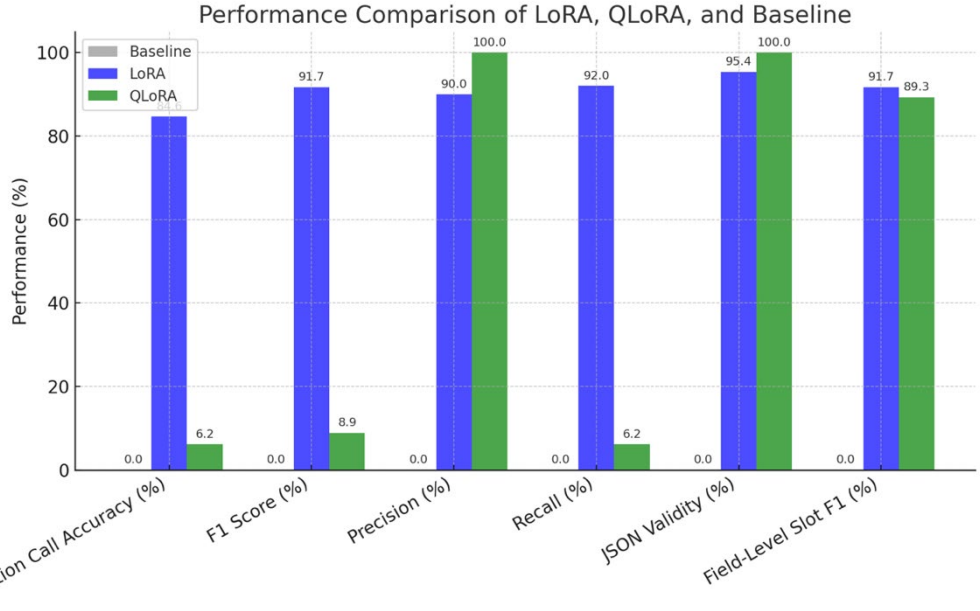

Figure 2. Comparative performance of the baseline, LoRA, and QLoRA models across key evaluation metrics

4.5 Deployment Considerations

Both LoRA and QLoRA are straightforward to integrate into existing pipelines. Their parameter-efficient design reduces computational overhead, making deployment feasible in enterprise environments with limited hardware resources. However, the performance gap in function-call classification suggests that LoRA is currently the more viable option for production scenarios. QLoRA's advantage in memory efficiency may make it attractive in environments with extreme hardware constraints, but its low recall undermines its reliability for mission-critical applications.

Overall, the results confirm that LoRA is a practical and reliable fine-tuning strategy for structured enterprise automation, delivering high accuracy, strong schema compliance, and robust field-level slot extraction with modest latency. QLoRA, while advantageous in memory and latency, exhibited overly conservative function recognition, making it less suitable for real-world deployments despite its excellent JSON compliance. These findings highlight a key trade-off in parameter-efficient fine-tuning: while quantization can reduce resource requirements, it may compromise task-specific accuracy, particularly in structured function-calling contexts. For enterprises, the decision between LoRA and QLoRA depends on deployment constraints—LoRA is recommended for scenarios prioritizing accuracy

and reliability, whereas QLoRA may be considered only in cases where extreme memory efficiency is essential and limited accuracy can be tolerated.

6. Conclusion

This study compared Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRA) for fine-tuning the LLaMA 8B model on structured function-calling tasks in enterprise contexts. Results showed that LoRA achieved robust performance (84.62% accuracy, 92% recall, and 91.67% F1), with strong JSON validity (95.38%) and reliable slot extraction. In contrast, QLoRA delivered perfect precision and JSON compliance but extremely low recall (6.15%), limiting its applicability for production use despite modest latency gains and greater memory efficiency. Both methods failed to detect escalation intent, highlighting an important limitation. Overall, LoRA offers the most effective balance of accuracy, structural consistency, and efficiency, making it the preferred strategy for enterprise deployment. Future work should focus on integrating LoRA with quantization-aware enhancements, developing lightweight auxiliary heads for escalation handling, and implementing Human-in-the-Loop systems during deployment to validate uncertain predictions and continuously refine performance through feedback.

Acknowledgments

The authors gratefully acknowledge the Kulliyyah of Information and Communication Technology, International Islamic University Malaysia (IIUM), for providing facilities and financial support for this research. The authors also wish to thank Cention, Sweden, for supporting this work with additional resources.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

References

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

[2] Metz, L., Freeman, C. D., Harrison, J., Maheswaranathan, N., & Sohl-Dickstein, J. (2022, November). Practical tradeoffs between memory, compute, and performance in learned optimizers. In *Conference on Lifelong Learning Agents* (pp. 142-164). PMLR.

[3] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and efficient foundation language models.

[4] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., ... & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

[5] Lester, B., Al-Rfou, R., & Constant, N. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691.

[6] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP. arXiv:1902.00751.

[7] Ben Zaken, E., Ravfogel, S., & Goldberg, Y. (2022). BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. arXiv:2106.10199.

[8] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, *36*, 10088-10115.Xu, Y., Xie, L., Gu, X., Chen, X., Chang, H., Zhang, H., Chen, Z., Zhang, X., Tian, Q. (2023). QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models.

[9] Lawton, N., Padmakumar, A., Gaspers, J., FitzGerald, J., Kumar, A., Ver Steeg, G., Galstyan, A. (2024). QuAILoRA: Quantization-Aware Initialization for LoRA.

[10] Patil, R. (2025). Analyzing LLaMA3 Performance on Classification Task with LoRA and QLoRA. Applied Sciences.

[11] Gonzalez-Carabarin, L., Huijben, I. A., Veeling, B., Schmid, A., & van Sloun, R. J. (2022). Dynamic probabilistic pruning: A general framework for hardware-constrained pruning at different granularities. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(1), 733-744.

[12] Zhang, J., Zhou, Y., & Saab, R. (2023). Post-training quantization for neural networks with provable guarantees. *SIAM journal on mathematics of data science*, *5*(2), 373-399.

[13] Lee, K., & Park, J. (2022, October). Clipped Quantization Aware Training for Hardware Friendly Implementation of Image Classification Networks. In *2022 19th International SoC Design Conference (ISOCC)* (pp. 370-371). IEEE.

[14] Thrivikram, G. L., Ganesh, V., Sethuraman, T. V., & Perepu, S. K. (2021, July). Efficient knowledge distillation of teacher model to multiple student models. In *2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)* (pp. 173-179). IEEE.

[15] Fu, H., Zhou, S., Yang, Q., Tang, J., Liu, G., Liu, K., & Li, X. (2021, May). LRC-BERT: latent-representation contrastive knowledge distillation for natural language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 14, pp. 12830-12838).

[16] Wang, L., Chen, S., Jiang, L., Pan, S., Cai, R., Yang, S., & Yang, F. (2025). Parameter-efficient fine-tuning in large language models: a survey of methodologies. *Artificial Intelligence Review*, *58*(8), 227.

[17] Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. A. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, *35*, 1950-1965.