

How Much is Too Much? Exploring LoRA Rank Trade-offs for Retaining Knowledge and Domain Robustness

Darshita Rathore Vineet Kumar Chetna Bansal Anindya Moitra

PayPal Artificial Intelligence

PayPal, Bengaluru, India

{drathore, vkumar32, cbansal, amoitra}@paypal.com

Abstract

Large language models are increasingly adapted to downstream tasks through fine-tuning. Full supervised fine-tuning (SFT) and parameter-efficient fine-tuning (PEFT) methods, such as Low-Rank Adaptation (LoRA), are two dominant approaches. While PEFT methods are widely used for their computational efficiency, the implications of their configurations (e.g., rank) remain under-explored in downstream Q&A tasks and generalization. In this work, we perform a comprehensive evaluation across multiple reasoning and recall datasets, conducting a rank sweep to quantify the trade-off between SFT and PEFT. We also compare the accuracy of PEFT and SFT models across in-domain and out-of-domain adaptation, highlighting distinct generalization behavior and task-specific forgetting. We demonstrate that LoRA achieves competitive and in some cases superior performance compared to SFT, particularly on reasoning tasks at specific rank values. Additionally, we analyze the internal representations via spectral features and layer-wise attention structures, offering insights into representational drift and structural changes in attention patterns.

1 Introduction

Large Language Models (LLMs) have become indispensable for a wide range of use cases, including text generation, machine translation, summarization, question answering, data synthesis & insights generation and software development, to name a few. Beyond these core tasks, LLMs are increasingly embedded in AI-powered agents for more complex, real-world workflows such as document understanding, data extraction, financial analysis, legal research, and web-based intelligence gathering (Kumar et al., 2025; Minaee et al., 2025). Their ability to operate across diverse domains with minimal supervision has led to rapid adoption in enterprise and production settings.

Despite their impressive capabilities, aligning LLMs with specific domains or use cases typically requires task adaptation via fine-tuning. Full supervised fine-tuning (SFT) – where all model parameters are updated can improve performance, especially in high-stakes domains like law, finance, and medicine. However, SFT is computationally and memory intensive, often rendering it impractical at scale due to the size of modern models. Moreover, full fine-tuning poses challenges related to catastrophic forgetting (Haque, 2025) and particularly when adapting to multiple tasks or clients in dynamic production environments.

Parameter-Efficient Fine-Tuning (PEFT) methods, most notably Low-Rank Adaptation (LoRA) (Hu et al., 2022), have emerged as effective and scalable alternatives to full supervised fine-tuning. By injecting trainable low-rank matrices into the attention and feedforward layers of the model, LoRA enables fine-tuning with significantly fewer parameters, often without compromising task performance. This efficiency allows practitioners to maintain lightweight, domain-specific adapters while reusing a shared base model, thereby reducing both training and deployment costs.

While LoRA and other PEFT methods have been the focus of numerous empirical and theoretical investigations, most existing studies either emphasize absolute performance gains or analyze specific tasks in isolation. Few have systematically compared the structural and behavioral changes induced by LoRA with those resulting from full supervised fine-tuning (SFT) in a model-agnostic manner. However, a unified understanding of how different fine-tuning strategies, particularly LoRA with varying rank configurations, affect internal representations, generalization, and forgetting across reasoning and factual recall tasks remains lacking. In this work, we address this gap through a comprehensive evaluation framework that connects performance metrics with interpretability and model

dynamics.

Building on this direction, we present a comprehensive study of LoRA rank selection and its effects on model behavior and performance. Our main contributions are as follows:

1. We systematically evaluate how varying the LoRA rank affects downstream performance across multiple datasets and domain setups.
2. We compare SFT and PEFT approaches to assess whether full fine-tuning offers consistent benefits over parameter-efficient methods, especially for recall & reasoning tasks.
3. We examine cross-domain performance degradation post fine-tuning, quantifying both forgetting and loss of generalization.
4. We analyze how internal representations, attention patterns, and layer-level drift differ from the base model after SFT and PEFT.

2 Related Works

Full supervised fine-tuning (SFT) of large language models is a computationally expensive process. This has motivated extensive research on parameter-efficient fine-tuning (PEFT) techniques that adapt models without updating all weights.

Han et al. (2024) provides a comprehensive taxonomy of PEFT approaches, categorizing them into four primary families: *additive methods* (e.g., adapters, soft prompts), *selective methods* (e.g., parameter masking), *reparameterized methods* (e.g., low-rank decomposition), and *hybrid approaches*. These methods substantially reduce memory and computational cost while maintaining downstream accuracy, enabling rapid task specialization even for billion-parameter models. Among these, low-rank reparameterization has emerged as a particularly compelling trade-off between efficiency and representational flexibility, allowing practitioners to inject compact task-specific capacity with minimal inference overhead.

Low-Rank Adaptation (LoRA) (Hu et al., 2022) injects trainable low-rank matrices $\Delta W = BA$ into existing linear projections and learns only A, B while freezing the pretrained weights. The learned update can be merged into the base weights at inference time, introducing no additional latency. The original work demonstrated competitive accuracy with orders of magnitude fewer trainable

parameters, but offered limited guidance on how to select the rank parameter across task families.

Subsequent analyses by Biderman et al. (2024) compared LoRA to full supervised fine-tuning and argued that LoRA “learns less and forgets less” highlighting that the effective update induced by SFT often possesses substantially higher intrinsic rank and thus greater capacity to both specialize and overwrite pretrained knowledge. *Catastrophic forgetting* – a phenomenon where a model trained on a new task drastically forgets previously learned information remains a central challenge in model adaptation (Haque, 2025). LoRA has been shown to mitigate forgetting on out-of-domain tasks, offering a favorable alternative. However, prior studies typically fixed the LoRA rank r to a small constant, leaving open the quantitative relationship between r and both in-domain and cross-domain behavior under-explored.

Ren et al. (2024) further explored this issue and demonstrated that LoRA reduces forgetting compared to full fine-tuning, particularly when applied selectively. They introduced *Interpolation-based LoRA* (I-LoRA), which leverages mode connectivity and a dual-memory learning mechanism to balance plasticity and stability. These approaches reveal that the geometry of the adaptation subspace strongly influences retention and transfer, motivating finer-grained analyses of how capacity constraints shape representational change.

Generalization under domain shift remains critical for achieving robust real-world performance. Complementary work on understanding *where* fine-tuning alters model representations has provided deeper insight into PEFT behavior. Hao et al. (2020) used Jensen–Shannon divergence and Singular Vector Canonical Correlation Analysis (SVCCA) to show that BERT fine-tuning predominantly modifies upper layers, leaving lower layers largely intact across tasks. Such layer-wise analyses motivate studying how low-rank updates affect internal representations differently from full fine-tuning.

Our study builds upon these lines of research by systematically examining the *rank–performance trade-off* in LoRA and its consequences for knowledge retention, reasoning ability, and cross-domain generalization.

3 Methodology

3.1 Models and Fine-Tuning Configuration

For our experiments, we used two instruction-tuned language models: **LLaMA-3.1-8B-Instruct** (Grattafiori et al., 2024) and **Qwen-2.5-7B-Instruct** (Yang et al., 2024). Both models belong to the latest generation of open-source foundation models and demonstrate strong performance across a wide range of standard benchmarks. Their instruction-following capabilities and architectural improvements make them well-suited for evaluating diverse Q&A tasks.

3.2 Tasks and Datasets

For our experiments, we considered three broad families of question-answering (Q&A) tasks: general knowledge, mathematical reasoning, and domain-specific specialized tasks. These include both free-form text generation and multiple-choice question formats. Specifically, we used the GSM8K (Cobbe et al., 2021), MMLU (Hendrycks et al., 2021), and MedMCQA (Pal et al., 2022) datasets to represent the three task categories. GSM8K focuses on grade-school level mathematical reasoning, MMLU (Massive Multitask Language Understanding) covers a wide range of general knowledge domains, and MedMCQA targets medical domain expertise. The data distribution across the datasets is as follows: GSM8K consists of 7,473 training and 1,319 test examples; MMLU includes 99,842 training and 14,042 test samples; MedMCQA contains 182,822 training and 6,150 test instances. Furthermore, we incorporated MathQA (Amini et al., 2019) (with 29,837 training and 3,589 test samples) and LegalMCQ (940 training samples) to diversify the reasoning, domain-specific, and cross-domain evaluation. More details on data distribution are detailed in Section A.1.

3.3 Evaluation Metrics

All of our evaluation datasets consist of question-answer (Q&A) formats. Except for GSM8K, all are structured as multiple-choice questions, which allows for reliable and consistent performance measurement through exact answer matching. This design choice was intentional: multiple-choice formats enable clear answer boundaries and reduce ambiguity in evaluating correctness. In case of GSM8K, where answers are free-form numeric responses with CoT style reasoning preceding it, we

leverage the dataset’s annotation convention where the final answer is always prefixed with the token `###`. This allows for straightforward extraction of the predicted answer using regular expressions. For all datasets, we consider a prediction correct if the generated answer string exactly matches the ground-truth answer string. We adopt accuracy (the proportion of correctly answered questions) as our primary evaluation metric.

4 Experimental Setup

4.1 Performance Trade-offs

To evaluate the performance of the base models, various LoRA configurations, and the full SFT models, we fine-tune each model on the training split of the respective datasets and evaluate them on the corresponding test sets¹.

Just with a small fraction of trainable parameters, LoRA achieves competitive downstream performance (Shuttleworth et al., 2025).

To evaluate the trade-off between model performance and fine-tuning, we train and assess each model in three configurations:

- **Base model:** Original pre-trained *off-the-shelf* base models were evaluated in a zero-shot setting using prompt-based inference, serving as a baseline.
- **LoRA fine-tuning:** LoRA configurations (except rank and alpha), including target modules (Key, Query, Value, and the Output layer), dropout, etc., were kept constant across all the experiments to enable a controlled and directly comparable evaluation. We sweep across five adaptation ranks $r \in \{8, 16, 32, 64, 128\}$ to analyse performance trends across varying levels of trainable parameter capacity. Setting $\alpha = 2 \times r$ has been empirically shown to improve results (Shuttleworth et al., 2025) and avoid intruder dimensions with better generalization (Biderman et al., 2024).
- **Full-SFT:** Standard full supervised fine-tuning of all model weights, representing the upper bound in terms of adaptation flexibility and computational cost.

Further, hyperparameters such as the number of epochs, learning rate, optimizer, scheduler type,

¹Except for MedMCQA, where ground truth answers are not available in the test set.

and maximum sequence length were kept constant during training. For inference, the parameters were matched to each model’s training configurations.

The accuracy metric used for evaluation is defined in Section 3.3, and the results are summarised in Table 1.

4.2 Knowledge Retention, Forgetting & Out-of-Domain Generalization

Along with the task accuracy, we evaluated how much pre-trained knowledge is retained after fine-tuning for each model. Prior work has shown erosion of encoded world knowledge in language models with an increase in the amount of fine-tuning data (Dou et al., 2023). To quantify this, we evaluate models both before and after fine-tuning on specific downstream tasks.

The experiments included:

- As a proxy for factual retention, evaluating the model on knowledge-intensive tasks (e.g., MMLU).
- Comparing performance and generalization drop on benchmarks between LoRA and Full-SFT configurations.
- Evaluating on unseen domains (e.g., legal QA, math QA) with models fine-tuned on a specific task (e.g., MedMCQA).

The quantitative results from these experiments are presented in Tables 1, 2, and 3. A detailed analysis and interpretation of these results, including model-wise and task-wise trends, is provided in the Discussion section (Section 5).

4.3 Training & Inference Infrastructure

Fine-tuning was performed on a compute cluster equipped with 4x NVIDIA H100 GPUs (80GB each) connected via NVLink, enabling high-throughput training for LLMs. We used mixed-precision training (bfloat16 where supported, otherwise fp16) to optimise GPU memory usage and computational speed. The unsloth (Daniel Han and team, 2023) framework was employed for efficient LoRA fine-tuning with gradient checkpointing and support for large batch sizes via gradient accumulation. Distributed training was handled using HuggingFace’s Trainer (von Werra et al., 2020) API with PyTorch’s DDP backend.

For inference, we utilized vLLM (Kwon et al., 2023), an optimised inference engine that supports

paged attention and continuous batching, allowing for significantly faster and memory-efficient evaluation of the fine-tuned models. This setup enabled low-latency serving and efficient evaluation across multiple datasets and model variants.

5 Discussion

5.1 Performance Trade-offs: LoRA’s Efficiency and Efficacy

The experimental results in Table 1 highlight LoRA’s significant role as an effective and scalable alternative to full SFT. Across a majority of datasets and models, LoRA configurations consistently deliver substantial performance improvements over the base models, often exceeding those of Full SFT variants. This is consistent with the premise that PEFT methods can enable adaptation with significantly fewer parameters without compromising task performance. A particularly noteworthy observation comes from the performance on the MMLU dataset. For both LLaMA-3.1-8B-Instruct and Qwen-2.5-7B-Instruct, LoRA configurations consistently outperform Full SFT. This outcome challenges the intuitive assumption that updating all model parameters through Full SFT would invariably lead to superior performance due to greater learning capacity. Instead, for general knowledge tasks like those in MMLU, which cover a wide range of domains, the constrained adaptation space of LoRA appears to act as a regularizer and by injecting only inherently low-rank matrices, LoRA limits the degrees of freedom for adaptation, potentially preventing the model from drastically altering its core knowledge base or overfitting to the specific training distribution. This preservation of broader pre-trained knowledge, coupled with targeted adaptation, led to better generalization and was more effective than a full fine-tune.

Findings of our experiments also highlight that there is no single rank that uniformly outperforms others. However, the variability is minimal in certain classes of tasks like pure recall (MMLU & MedMCQA) – all ranks achieve almost similar performance; however, for more involved reasoning and math-based tasks, some ranks are better than others. This variability indicates the interplay between the nature, complexity of the task and data distribution in downstream performance. Tasks requiring more nuanced or extensive adaptations might benefit from a slightly higher rank, whereas

Model	Dataset	Base Model	PEFT Model (Rank r)					Full SFT
			$r = 8$	$r = 16$	$r = 32$	$r = 64$	$r = 128$	
Llama-3.1-8B-Instruct	MMLU	36.95%	57.39%	57.44%	57.21%	57.24%	57.20%	53.03%
	GSM8K	81.65%	65.35%	67.93%	69.83%	71.11%	70.43%	56.33%
	MedMCQA	45.45%	51.90%	50.83%	51.67%	51.44%	51.67%	49.62%
Qwen-2.5-7B-Instruct	MMLU	31.15%	65.46%	65.87%	66.04%	66.15%	65.66%	60.90%
	GSM8K	58.30%	66.94%	68.99%	71.80%	74.98%	70.05%	71.34%
	MedMCQA	11.30%	27.24%	32.09%	21.40%	25.49%	21.17%	27.69%

Table 1: Model Performance Comparison: Base Model vs PEFT (LoRA Rank Sweep) vs Full SFT

simpler tasks or those where the base model already possesses strong foundational abilities might require less adaptation.

Another interesting finding was with the GSM8K mathematical reasoning dataset. For LLaMA-3.1-8B-Instruct, both LoRA and SFT resulted in a significant decrease in accuracy compared to base model performance. This unexpected degradation suggests the latest model with good performance and strong instruction following capability may already exhibit superior performance, and SFT/PEFT is not beneficial. It is also likely that fine-tuning on specific tasks could induce biases which lead to a general loss of mathematical abilities. This underscores the importance of carefully evaluating the base model for downstream tasks, as fine-tuning may not always be required.

Model	Trained on	Evaluated on	Base	LoRA
LLaMA	MedMCQA	LegalQA	34.25%	58.19%
		MathQA	31.20%	21.04%
		GSM8K	81.65%	74.37%
	GSM8K	MedMCQA	45.45%	46.51%
		Legal	34.25%	34.47%
Qwen	MedMCQA	LegalQA	49.46%	63.94%
		MathQA	25.97%	28.36%
		GSM8K	58.30%	77.48%
	GSM8K	MedMCQA	11.30%	26.56%
		Legal	49.46%	57.02%

Table 2: Cross-task generalization performance of Base vs LoRA fine-tuned models on various QA datasets.

Model	Trained on	Evaluated on	Base	LoRA
LLaMA	GSM8K	MathQA	31.20%	22.32%
Qwen	GSM8K	MathQA	25.97%	32.74%

Table 3: Inter Domain generalization: Trained on a domain and evaluated on a similar domain but different distribution

5.2 Generalization Capabilities: Cross & Inter-Domain

When models are trained on one domain and evaluated on another, LoRA-tuned models often exhibit robust generalization, sometimes even showing improvements over the base model. For example, Training on MedMCQA and evaluation on LegalQA, a significant performance improvement is observed (approximately 25% and 15% for LLaMA-3.1-8B-Instruct and Qwen-2.5-7B-Instruct, respectively). This suggests LoRA is not merely memorizing the facts specific to the training domain, rather it is learning more abstract and transferable skills. We conjecture this is true in the case of MedMCQA and LegalQA because both of these are factual recall-based tasks. However, this doesn’t always hold in general; we also observed instances of negative transfer. For LLaMA-3.1-8B-Instruct, fine-tuning on MedMCQA led to a decrease in accuracy when evaluated on MathQA, dropping by 10%. This indicates a clear risk of catastrophic forgetting. The adaptation process for one task, particularly when the domains are fundamentally different (e.g., factual recall in medicine vs. reasoning in mathematics), might optimize the model’s parameters in a way that conflicts with or overwrites internal representations crucial for other capabilities.

Even within similar domains like mathematical reasoning, differences in data distribution between GSM8K and MathQA lead to LoRA adapters learning task-specific features that do not generalise to slightly different problem sets. Our experiments also highlight model-specific generalization behaviors. When trained on GSM8K and evaluated on MathQA, Qwen-2.5-7B-Instruct demonstrated an improvement in accuracy while LLaMA-3.1-8B-Instruct experienced a decrease. This divergence suggests that the underlying architectural differences between LLaMA-3.1-8B-Instruct and Qwen-

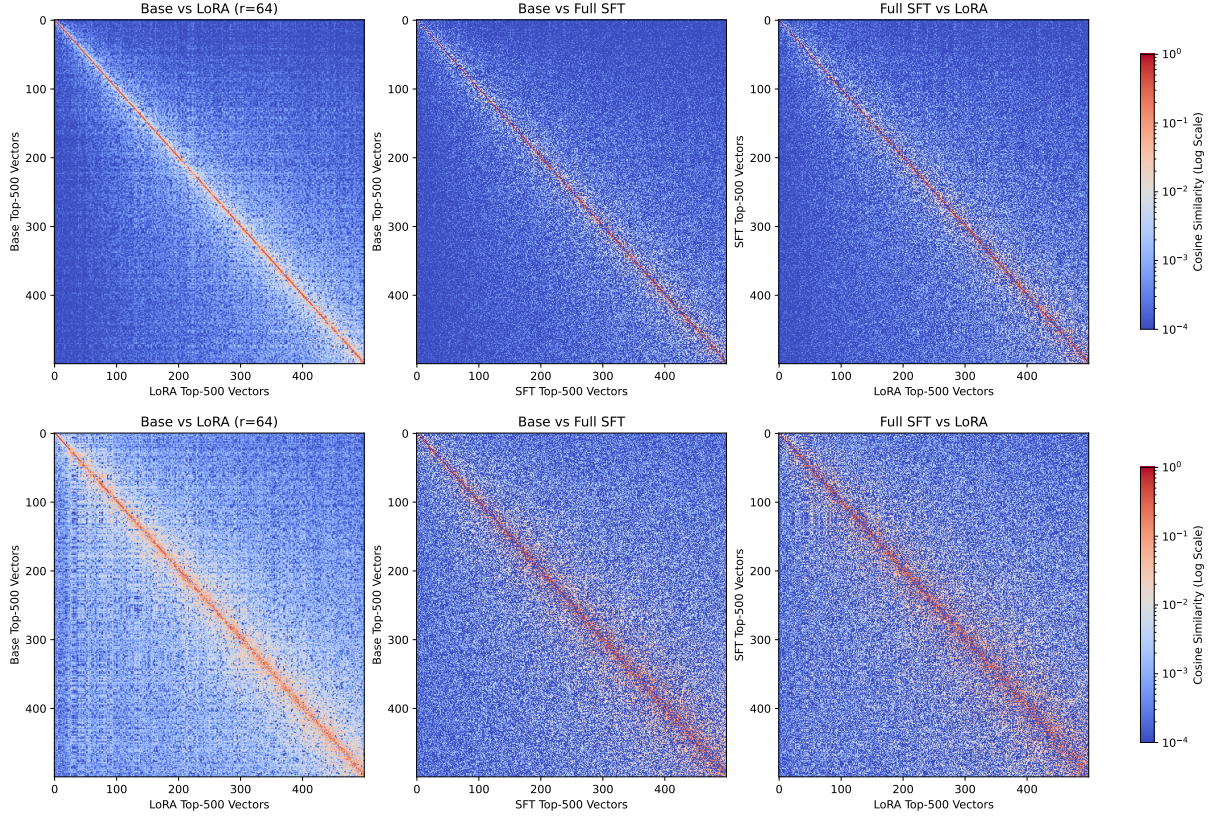


Figure 1: Similarity Heatmaps: Full SFT and for LoRA LLaMA GSM8K (top) and Qwen MMLU(bottom)

2.5-7B-Instruct, such as specific attention mechanisms, normalisation layers, influence how effectively LoRA can adapt and generalise.

5.3 Interpretability Analysis

Spectral Features of Weight Matrices We investigated how fine-tuning alters the fundamental characteristics of weight matrices by examining the similarity between their singular vectors before and after the fine-tuning process. We compute the cosine similarity between the top 500 singular vectors obtained via singular value decomposition (SVD) of the weight matrices to capture spectral shifts induced by adaptation. Figure 1 presents these similarity heatmaps for both LLaMA-3.1-8B-Instruct and Qwen-2.5-7B-Instruct models. The observed patterns indicate that the learning dynamics differ substantially between full supervised fine-tuning (SFT) and parameter-efficient fine-tuning (PEFT), suggesting distinct modes of representation change. Full SFT, by modifying all model parameters, allows for a holistic and potentially more drastic reshaping of the entire representation space. While this can lead to superior optimisation for a specific task, it might also result in greater catastrophic forgetting of pre-trained knowledge. In contrast,

LoRA, through the adaption of low-rank matrices into specific layers, induces more targeted changes and preserves the existing structure.

Attention Head Ablation As part of our interpretability analysis, we perform attention head ablation to identify which attention heads contribute most to the model’s output. For a given input, we systematically zero out individual attention heads and measure the drop in the log-probability of the correct answer. This is an established approach in many interoperability studies (Zhou et al., 2024; Michel et al., 2019). A larger drop indicates that the head is more critical to the model’s prediction. This approach allows us to quantify the functional importance of specific heads and track how this importance shifts across fine-tuning methods (LoRA vs. SFT) and task types (reasoning vs. recall). By comparing the ablation maps across models, we gain insight into how fine-tuning redistributes or reinforces focus on certain attention pathways. Figure 2 reveals that only a small subset of heads contribute significantly to task performance. The heatmaps show concentrated impact in mid-to-late layers, indicating that LoRA and SFT models rely on different attention pathways.

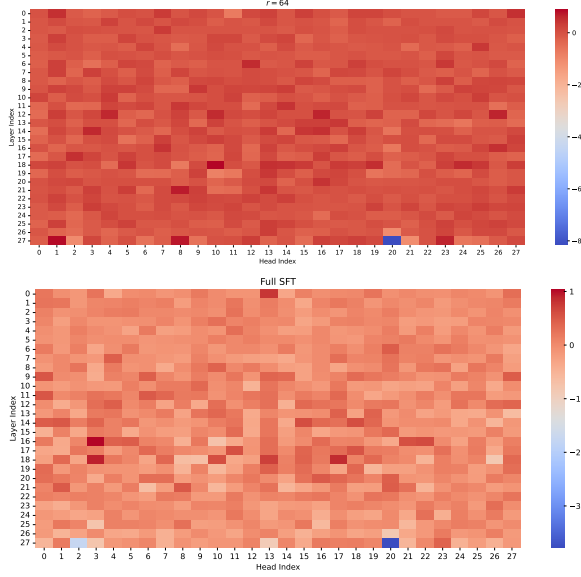


Figure 2: Log-Probability Drop After Attention Head Ablation for Qwen LoRA (top) and SFT (bottom)

Frobenius (l_2) Norm The Frobenius norm provides a proxy quantifying the overall magnitude of parameter changes during model adaptation. In the context of Low-Rank Adaptation (LoRA), this norm represents the cumulative strength of weight updates across the model, serving as a direct measure of how significantly the adaptation process modifies the base model’s parameters. Formally, the Frobenius norm of a matrix is calculated as the square root of the sum of squared elements, effectively capturing the total magnitude of all parameter changes:

$$\|\Delta W\|_F = \sqrt{\sum_{i,j} (\Delta W_{ij})^2}$$

For LoRA adaptations, where weight updates are decomposed into low-rank matrices ($\Delta W = B \times A$), the norm quantifies the effective strength of these adaptations while accounting for their interaction effects.

As evident in Figure 3, Frobenius norm exhibits approximately logarithmic growth with increasing LoRA rank across all model-task combinations. MMLU adaptations consistently show the highest Frobenius norms, suggesting that general knowledge tasks require more substantial modifications. GSM8K shows the lowest norm values for both models, indicating that mathematical reasoning capabilities might require more focused, rather than expansive, changes.

Looking at this from a model lens – LLaMA-

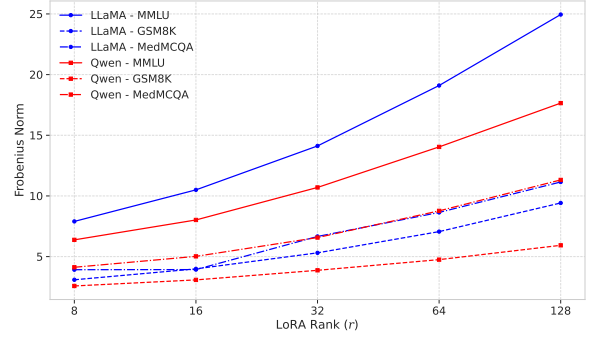


Figure 3: Quantifying Low-Rank Adaptation Impact: Frobenius Norm Scaling with Rank

3.1-8B-Instruct models demonstrate steeper growth in Frobenius norm with increasing rank compared to Qwen-2.5-7B-Instruct models, particularly for MMLU and GSM8K tasks, suggesting that LLaMA-3.1-8B-Instruct’s architecture might be more responsive to increasing parametric capacity during adaptation. Both models show similar patterns for MedMCQA adaptations, with Qwen-2.5-7B-Instruct exhibiting slightly higher norms at higher ranks, potentially indicating better alignment with medical domain adaptation.

6 Conclusion

This study advances a unified understanding of Parameter-Efficient Fine-Tuning (PEFT), specifically LoRA, by addressing gaps in how it performs across recall and reasoning tasks. Prior work often focused on isolated benchmarks, whereas we systematically evaluate how varying LoRA rank affects performance, generalization, and internal representations.

Consistent with the *no-free-lunch* principle (Mitchell, 1997), there is no single universally optimal fine-tuning recipe for large language models. The effectiveness of adaptation depends on the interplay between task type, domain characteristics, and deployment constraints. Rather than proposing a one-size-fits-all rule, our contribution lies in establishing strong, evidence-based defaults that practitioners can reliably start from. Across reasoning and factual-recall tasks, we demonstrate that LoRA provides a computationally efficient fine-tuning method that preserves general knowledge while maintaining competitive downstream performance. Empirically, intermediate ranks ($r = 32-64$) offer a balanced operating point between representational capacity and stability, achieving robust performance. We view these recommendations not as

prescriptive choices, but as practical anchors that can be adapted to specific application contexts and model architectures.

Limitations

This study provides a comprehensive analysis of LoRA and SFT; however, it is subject to certain limitations that also suggest avenues for future research. On LoRA configuration choices, the current methodology states that setting $\alpha = 2 \times r$ has been empirically shown to improve results and avoid ‘intruder dimensions’ (Shuttleworth et al., 2025) with better generalization. This specific choice was applied consistently across experiments to ensure fairness. Future work could explore the impact of varying α independently of r or investigate other LoRA variants and their respective optimal configurations. This study focused on two specific instruction-tuned LLMs. Future work could extend this to a wider range of models (e.g., larger models, different architectures, non-instruction-tuned models) and compare LoRA against other PEFT variants (e.g., Prefix-tuning, Prompt-tuning, Adapter, QLoRA) to provide a more comprehensive understanding of the PEFT landscape. Furthermore, while diverse Q&A tasks were covered, exploring other NLP tasks such as text generation, summarisation, or classification could yield additional insights into fine-tuning trade-offs. For deeper interpretability, future research could move beyond spectral features and attention ablation to explore other methods, such as neuron activation analysis, concept activation vectors, or causal mediation analysis, to gain a more granular understanding of how fine-tuning alters specific model behaviors. For observed negative transfer or performance degradation, future work could propose and evaluate mitigation strategies, such as multi-task fine-tuning with LoRA, selective LoRA application, or incorporating advanced regularization techniques.

Acknowledgments

The authors thank the anonymous reviewers of AACL IJCNLP 2025 and OpenReview ARR for their insightful feedback and suggestions. The authors also dedicate this work to the memory of Prof. Pushpak Bhattacharya, whose vision, generosity, and teachings have inspired countless NLP researchers.

References

- Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. [MathQA: Towards interpretable math word problem solving with operation-based formalisms](#). 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)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.
- Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, and 1 others. 2024. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Michael Han Daniel Han and Unsloth team. 2023. [Unsloth](#).
- Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, and 1 others. 2023. [Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin](#). *arXiv preprint arXiv:2312.09979*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. [Parameter-efficient fine-tuning for large models: A comprehensive survey](#). *arXiv preprint arXiv:2403.14608*.
- Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Investigating learning dynamics of bert fine-tuning. In *Proceedings of the 1st conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th international joint conference on natural language processing*, pages 87–92.
- Naimul Haque. 2025. [Catastrophic forgetting in llms: A comparative analysis across language tasks](#). *Preprint*, arXiv:2504.01241.
- Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. [Aligning ai with shared human values](#). *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,

- Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Vineet Kumar, Ronald Tony, Darshita Rathore, Vipasha Rana, Bhuvanesh Mandora, . Kanishka, Chetna Bansal, and Anindya Moitra. 2025. **Genicious: Contextual few-shot prompting for insights discovery**. In *Proceedings of the 8th International Conference on Data Science and Management of Data (12th ACM IKDD CODS and 30th COMAD)*, CODS-COMAD '24, page 405–409, New York, NY, USA. Association for Computing Machinery.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. **Are sixteen heads really better than one?** *Preprint*, arXiv:1905.10650.
- Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. **Large language models: A survey**. *Preprint*, arXiv:2402.06196.
- Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill. Free PDF available: <https://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/mlbook.html>.
- Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. **Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering**. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.
- Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*.
- Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. 2025. **Lora vs full fine-tuning: An illusion of equivalence**. *Preprint*, arXiv:2410.21228.
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Galouédec. 2020. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>.
- Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Hao-ran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. **Qwen2.5 technical report**. *ArXiv*, abs/2412.15115.
- Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. 2024. On the role of attention heads in large language model safety. *arXiv preprint arXiv:2410.13708*.

A Appendix

A.1 Dataset Distribution

Table 4 summarises the size and availability of splits across all datasets used in our experiments. MMLU serves as a general knowledge benchmark, while GSM8K and MATHQA target reasoning and numerical comprehension. MedMCQA and LegalMCQ cover domain-specific QA for the medical and legal domains, respectively.

Table 4: Dataset Statistics

Dataset	Split(s)	Number of Records
GSM8K	train, test	7,473 / 1,319
LegalMCQ	train	940
MATHQA	train, dev, test	29,837 / 4,475 / 3,589
MedMCQA	train, validation, test	182,822 / 4,183 / 6,150
MMLU	train, validation, dev, test	99,842 / 1,531 / 285 / 14,042

A.2 SFT and PEFT Configuration

Full-SFT: For full supervised fine-tuning, we train all model parameters using the adamw_8bit optimizer with a learning rate of 5e-5 and weight decay of 0.01. Training is performed using a per_device_train_batch_size of 2 and gradient_accumulation_steps of 4, yielding an effective batch size of 8 per update step. We fine-tune for 3 epochs, using a linear learning rate scheduler with 10% warmup steps. Mixed precision is enabled, automatically selecting between fp16 and bf16 based on hardware support.

Evaluation and checkpointing are conducted every 500 steps. Training logs and metrics are reported via TensorBoard. All models are trained with a maximum sequence length of 2048 tokens.

PEFT with LoRA via Unsloth: We apply parameter-efficient fine-tuning using the unsloth framework (Daniel Han and team, 2023), which wraps HuggingFace’s PEFT and TRL libraries (von Werra et al., 2020). We use LoRA with attention projection layers (q_proj, k_proj, v_proj, o_proj) as target modules - this matches the architecture of LLaMA-3 and Qwen models.

Key LoRA hyperparameters:

- $r = 32$

- `lora_alpha = 64`
- `lora_dropout = 0.0`
- `bias = "none"`

Gradient checkpointing is enabled via `use_gradient_checkpointing = "unsloth"`, improving memory efficiency. We do not use any quantisation techniques (i.e. `loftq_config = None, use_rslora = False`).

For training, we use the `SFTTrainer` class provided by Unsloth with:

- `max_seq_length = 2048`
- `dataset_text_field = "text"`
- `packing = False` (no input packing used)
- `dataset_num_proc = 2` (parallel preprocessing)

We use the same tokeniser, data splits, and stopping criteria across both SFT and LoRA runs for consistency.

A.3 System Prompt for Training & Evaluation

To guide model behavior across tasks, we prepend task-specific system prompts during both training and evaluation. These prompts are designed to reflect the dataset domain and expected output style.

For example:

- **MMLU (Knowledge):** "You are a helpful AI assistant that specializes in multiple-choice questions. Solve this MCQ and provide the correct option."
- **GSM8K / MathQA (Reasoning):** "You are a math expert. Solve the problem step-by-step and return the final answer."
- **MedMCQA:** "You are a medical assistant. Carefully analyse the question and provide the correct option."
- **LegalMCQ:** "You are a legal expert. Read the question and choose the most accurate answer."

All prompts are applied consistently across training and evaluation to ensure stable behavior and performance alignment.

A.4 Layer-wise Norm Distribution

To better understand how different LoRA configurations affect adaptation across the model layers, we visualise the layer-wise norm distributions of the LoRA weights for LLaMA fine-tuned on the MMLU dataset.

Figure 4 presents the l_2 norm of the injected LoRA deltas (adapter weights) across transformer layers for different LoRA ranks (r). These plots help identify which layers are more sensitive to adaptation and how this sensitivity varies with rank.

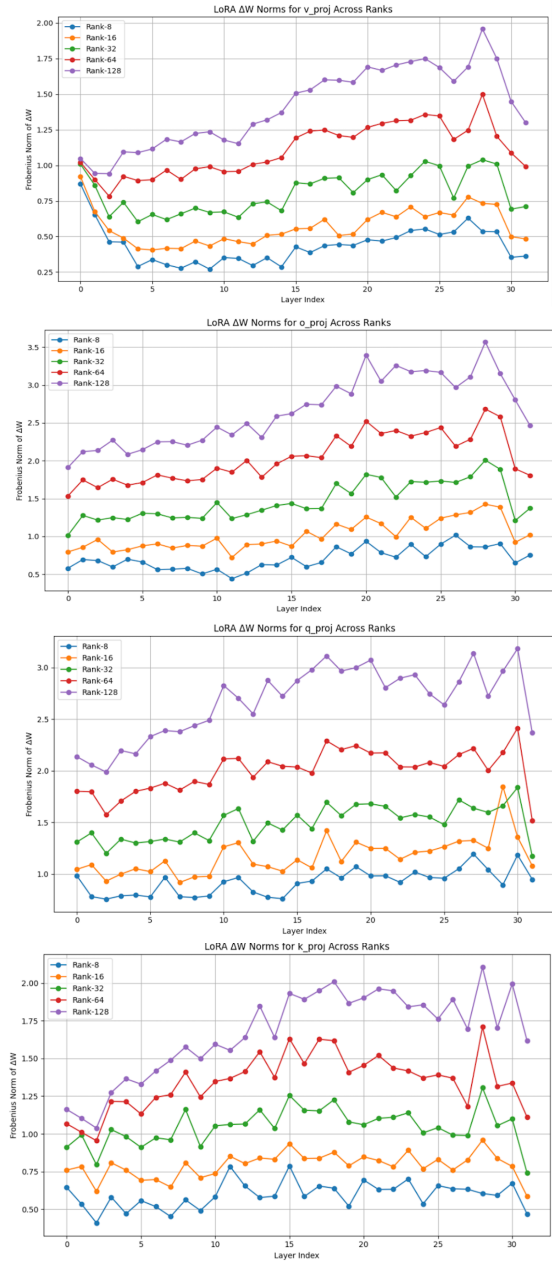


Figure 4: Comparison of layer-wise LoRA adapter norm distributions across different ranks $r \in \{8, 16, 32, 64, 128\}$ for LLaMA fine-tuned on MMLU.

As observed, the middle and upper transformer

blocks tend to accumulate more change as the rank increases, suggesting that LoRA adaptation is non-uniform across layers. This aligns with prior findings that later layers contribute more to task-specific reasoning and learning ([Hao et al., 2020](#)). The layer norm trend can inform future decisions on layer selection for targeted PEFT.