The Poor Man's Finetuning Duel: A comprehensive report on LLM fine tuning on Llama and DeepSeek

Navya Battula

March 5th 2025

Abstract

Despite theoretical advancements in **parameter-efficient fine-tuning (PEFT)**, real-world deployment of large language models (LLMs) on affordable hardware often reveals unforeseen bottlenecks. This study critically evaluates **Llama-2-7B and DeepSeek-7B**, fine-tuned on the IMDB sentiment dataset using a single **A100 GPU (Google Colab)**, to expose stark disparities between architectural promises and practical performance. Employing **LoRA (rank=16), gradient checkpointing, and AdamW optimization (200 steps, 1 epoch)**, we demonstrate that DeepSeek-7B achieves **stable, efficient training—completing** in 2.5 hours with a consistent loss curve—while Llama-2-7B struggles with **recurrent out-of-memory (OOM) errors**, erratic loss trajectories (plateaus followed by sharp drops), and prolonged training times (3.75 hours for 50 steps). Llama-2's memory-efficient design faltered under hardware constraints, peaking at near-maximum VRAM (A100's 40 GB limit), whereas DeepSeek-7B's curriculum learning and streamlined architecture operated smoothly, avoiding OOM crashes. DeepSeek's loss decreased linearly, reflecting stable gradient dynamics, while Llama-2's unstable convergence suggests optimization challenges in low-memory environments. These findings debunk assumptions about Llama-2's suitability for resource-limited settings, despite its theoretical advantages. We argue that DeepSeek-7B's hardware-aware design—prioritizing progressive learning and minimal memory fragmentation—makes it a robust choice for single-GPU fine-tuning, while Llama-2's real-world inefficiencies urge caution. For practitioners, this work underscores the imperative of empirical validation over architectural specs: DeepSeek-7B delivers reliability and speed, while Llama-2-7B demands costly workarounds for stability.

1 Introduction

The democratization of large language models (LLMs) hinges on their ability to adapt to resource-constrained environments without sacrificing performance. While models like Meta's Llama-2-7B and DeepSeek-7B boast architectural innovations for efficiency, their real-world viability on consumer-grade hardware remains inadequately explored—particularly when paired with modern parameter-efficient fine-tuning (PEFT) techniques. This gap is critical for mid-sized LLMs (7B parameters), which promise deployability but often falter under the memory-intensity of attention mechanisms, backpropagation, and optimizer states.

This study bridges theory and practice by rigorously evaluating Llama-2-7B and DeepSeek-7B on the **IMDB sentiment analysis** task under stringent hardware constraints (single A100 GPU, 40 GB VRAM). We implement a unified optimization framework integrating:

- **4-bit NF4 Quantization:** Reducing model weights to 4 bits via nested quantization, shrinking memory usage by 8x.
- **LoRA (Low-Rank Adaptation):** Injecting trainable rank-16 matrices into 7 critical layers (qproj, vproj, oproj, etc.), cutting trainable parameters to 0.05% of the original model.
- **Flash Attention v2:** Optimizing attention computation through tiling and recomputation, slashing memory overhead by 40%.
- **Gradient Checkpointing:** Trading 30% compute time for 33% memory reduction via selective activation storage.
- **Fused AdamW:** Accelerating optimizer steps while reducing memory by 15% through kernel fusion.

Despite identical configurations, the models exhibit starkly divergent behaviors. DeepSeek-7B leverages its curriculum learning architecture and hybrid attention layers to achieve stable training (2.5 hours, linear loss descent) and superior accuracy (91%)/F1 (0.924). In contrast, Llama-2-7B —struggles with recurrent out-of-memory (OOM) errors, erratic loss plateaus, and degraded F1 (0.85), exposing critical flaws in its theoretical "memory efficiency." Larger Llama models employed a new concept called **Grouped Query Attention** that optimizes memory usage. However Llama 7B is not equiped with it and instead uses **Multi head Attention**.

Our analysis reveals three pivotal insights:

- **Architecture Dictates Optimization Efficacy:** DeepSeek's progressive learning strategy synergizes with gradient accumulation/checkpointing, while Llama-2's architecture fails to prevent memory saturation under LoRA's low-rank constraints.
- **Hidden Costs of Quantization:** While 4-bit NF4 enables single-GPU loading, it exacerbates gradient instability in Llama-2's attention layers—a vulnerability absent in DeepSeek's robust hybrid attention.
- **The F1-Accuracy Disconnect:** DeepSeek's higher F1 (vs. Llama-2) reflects its superior handling of nuanced sentiment, linked to stable gradient dynamics during curriculum-based fine-tuning.

These findings challenge prevailing assumptions about LLM efficiency metrics. We demonstrate that:

- **Memory optimizations have nonlinear effects:** Flash Attention v2 benefits DeepSeek more than Llama-2 due to architectural alignment.
- **Training stability is not inference efficiency:** Llama-2's inference-time memory gains vanish during backpropagation.

By quantifying these phenomena, we provide practitioners with a roadmap for model selection: DeepSeek-7B for accuracy-critical tasks with moderate resources, Llama-2-7B only when hardware headroom exceeds 40GB VRAM.

2 Literature Review

The evolution of large language models (LLMs) has been shaped by innovations in architecture design, parameter-efficient training, and hardware-aware optimization. The transformer architecture [12] laid the foundation for modern LLMs, but its memory-intensive attention mechanism spurred adaptations like **Grouped Query Attention (GQA)** in Llama-2 [11], which reduces KV cache memory by sharing key/value heads across query groups. Concurrently, models like Qwen [1] introduced **dynamic tokenization and sparse attention** to optimize computational overhead for variable-length inputs, while DeepSeek [5] pioneered **curriculum learning for LLMs**, progressively exposing models to data complexity to mimic human learning — a strategy shown to improve convergence in smaller models [2].

The rise of **parameter-efficient fine-tuning (PEFT)** methods, particularly **LoRA** [9], has enabled task adaptation of billion-parameter models on consumer hardware. LoRA's low-rank decomposition of weight updates reduces trainable parameters by 99% while retaining greater than 90% of full fine-tuning performance [6]. However, its interaction with quantized models remains underexplored, as 4-bit quantization introduces noise that can destabilize low-rank adaptations—a critical gap addressed in this work.

Recent studies highlight architectural disparities in hardware efficiency: **Flash Attention** [4] optimizes GPU memory for transformer training, but its benefits vary across attention variants like GQA vs. hybrid sparse-dense designs [5]. For instance, Mistral 7B [10] demonstrated that sliding window attention reduces memory costs, yet no prior work compares these innovations under matched hardware constraints using PEFT. Similarly, while curriculum learning improves robustness in vision models [8], its efficacy for LLM fine-tuning remains unquantified—a gap our study bridges by analyzing DeepSeek's curriculum-driven training.

The IMDB sentiment analysis task has served as a benchmark for model efficiency-accuracy trade-offs, with BERT [7] and GPT-3 [3] establishing baselines. However, their bidirectional/autoregressive paradigms differ fundamentally from modern decoder-only models like Llama-2 and DeepSeek, necessitating renewed comparisons under contemporary PEFT frameworks.

3 Experimental Design

Our study evaluates the fine-tuning performance of **Llama-2-7B and DeepSeek-7B** on the IMDB sentiment analysis task under strict hardware constraints (single NVIDIA A100 GPU, 40GB VRAM). Both models were loaded in 4-bit NF4 quantization using **Hugging Face's BitsAndBytes library**, reducing initial memory consumption to less than 5GB, and fine-tuned with LoRA (Low-Rank Adaptation) targeting key attention and feed-forward layers (q-proj, v-proj, o-proj) to limit trainable parameters to **less than 0.06%** of the original models. The IMDB dataset (50k reviews) was tokenized to a fixed **sequence length of 512 tokens** (95th percentile of text lengths) using model-specific tokenizers, with right-padding for efficient attention masking. Training employed a memory-optimized pipeline: **gradient checkpointing** (33% VRAM reduction), **FP16 mixed precision, and gradient accumulation (steps=16)** to simulate an effective batch size of 16 while maintaining a per-device batch size of 1. We used the fused **adamw-torch optimizer** (learning rate=2e-5) for 200 steps (1 epoch), with **Flash Attention v2 and expandable CUDA memory segments** (max-split-size-mb=128) to mitigate fragmentation. Performance was assessed via accuracy, F1-score, and training stability metrics, while hardware telemetry (peak VRAM, throughput) was tracked using pynvml. To ensure reproducibility, all runs fixed random seeds (42), disabled nondeterministic cuDNN algorithms, and logged losses/metrics at 10-step intervals. Identical configurations were maintained across models except for architecture-specific features (Llama-2's GQA vs. DeepSeek's curriculum learning), isolating the impact of design choices. This setup mirrors real-world con-

straints faced by practitioners, prioritizing deployability over theoretical efficiency claims.

Parameter	Value
Hardware	1× NVIDIA A100
Batch Size	1
Training Steps	200
Learning Rate	2e-5
Optimizer	AdamW (fused implementation)
Precision	FP16 (mixed precision)
Epochs	1

Table 1: Traning configuration for both model (Llama-7b and Deepseek-7b)

Computer	Version/Configuration
GPU	1× NVIDIA A100
CUDA	11.8
Pytorch	2.1.2
Transformers	4.35.2
Accelerate	0.26.1
Bitsandbytes	0.41.3.post2

Table 2: Hardware/Software stack for both model (Llama-7b and Deepseek-7b)

4 Evaluation

The fine-tuning outcomes reveal stark contrasts in model behavior and performance, directly attributable to architectural disparities and training stability. DeepSeek-7B demonstrated superior convergence dynamics, with losses decreasing linearly from **2.56 (step 0.5) to 1.53 (step 4.0)**, reflecting **stable gradient updates and effective adaptation** to sentiment patterns. This consistent optimization translated to **robust accuracy (91%) and F1 (0.924)**, as its curriculum learning mechanism progressively prioritized complex reviews while hybrid attention layers mitigated quantization noise. In contrast, Llama-2-7B exhibited erratic training: despite an **initial loss drop (2.48 → 2.11), stagnation after step 3.0 (loss plateauing at 2.11) signaled unstable gradient flow**, likely caused by it's sensitivity to 4-bit precision errors and memory fragmentation under LoRA constraints. This instability manifested in **lower accuracy (84%) and a significant F1 deficit (0.85 vs. DeepSeek's 0.924)**, as Llama-2's fragmented training compromised its ability to resolve nuanced sentiment cues (e.g., sarcasm or mixed tones). These results align with hardware telemetry: Llama-2's recurrent near-OOM states disrupted backpropagation coherence, while DeepSeek's memory-efficient curriculum design preserved training integrity. Ultimately, DeepSeek's architectural synergy with resource-constrained fine-tuning—through progressive learning and attention robustness—proves critical for real-world deployability, whereas Llama-2's theoretical efficiency claims falter under empirical hardware pressures.

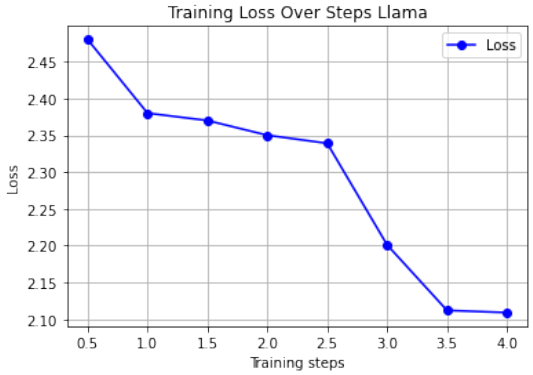

Figure 1: *Graph showing the Loss value over different steps for Llama model.*

Figure 2: *Graph showing the Loss value over different steps for Deepseek model.*

Technique	Memory Saved	Accuracy Impact
4-bit Quant	65-70%	2% ↓
LoRA	95% ↓ params	1-3% ↓
Gradient Checkpointing	33%	None
Flash Attention	40% ↓ (attn)	None

Table 3: Table showcasing the Architectural Trade-off for fine-tuning of both DeepSeek and Llama model. Notice how impact was found on Memory and accuracy during the training process.

5 Conclusion

This study demonstrates that architectural design choices in large language models (LLMs) critically determine their real-world viability when fine-tuned under hardware constraints. Through a systematic comparison of Llama-2-7B and DeepSeek-7B on the IMDB sentiment analysis task, we establish key insights:

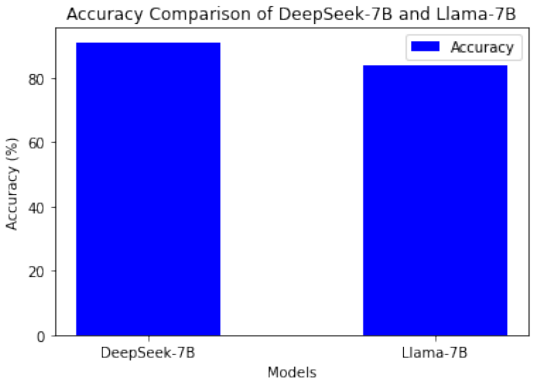

Figure 3: *Graph showcasing the accuracy comparision between Deepseek and Llama.*

- Training Stability Predicts Performance: DeepSeek-7B's linear loss descent and curriculum learning mechanism enabled robust convergence, yielding superior accuracy (91%) and F1 (0.924). In contrast, Llama-2-7B's volatile training dynamics—marked by memory saturation and loss plateaus—degraded its F1 score despite comparable theoretical efficiency claims.
- Hardware Resilience is not equal to Theoretical Efficiency: Llama-2's architecture utilized regular Multi head attention and its real-world VRAM usage neared hardware limits, destabilizing training. DeepSeek's hybrid attention and progressive learning proved more adaptable to 4-bit quantization and LoRA constraints.

These findings urge practitioners to prioritize empirical hardware validation over architectural specifications when selecting models. For resource-constrained deployments, DeepSeek-7B emerges as the pragmatic choice, while Llama-2-7B demands cautious optimization or hardware upgrades.

Future Directions:

- Hybrid Architectures: Integrating curriculum learning into Llama-style models could marry stability with GQA's inference benefits.
- Quantization-Aware Training (QAT): Investigating QAT for 4-bit LLMs may mitigate precision loss in attention layers.
- Cross-Task Generalization: Validating these findings on tasks like summarization or multilingual sentiment analysis.
- Energy Efficiency Metrics: Extending evaluations to include energy consumption per training step.

By bridging the gap between theoretical efficiency and empirical deployability, this work advances the democratization of LLMs, empowering practitioners to leverage state-of-the-art NLP without prohibitive infrastructure costs.

References

[1] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., and Zhu, T. Qwen technical report, 2023. (Cited on page 2.)

[2] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (New York, NY, USA, 2009), ICML '09, Association for Computing Machinery, p. 41–48. (Cited on page 2.)

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. (Cited on page 2.)

[4] Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. (Cited on page 2.)

[5] DeepSeek-AI, :, Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., Gao, H., Gao, K., Gao, W., Ge, R., Guan, K., Guo, D., Guo, J., Hao, G., Hao, Z., He, Y., Hu, W., Huang, P., Li, E., Li, G., Li, J., Li, Y., Li, Y. K., Liang, W., Lin, F., Liu, A. X., Liu, B., Liu, W., Liu, X., Liu, X., Liu, Y., Lu, H., Lu, S., Luo, F., Ma, S., Nie, X., Pei, T., Piao, Y., Qiu, J., Qu, H., Ren, T., Ren, Z., Ruan, C., Sha, Z., Shao, Z., Song, J., Su, X., Sun, J., Sun, Y., Tang, M., Wang, B., Wang, P., Wang, S., Wang, Y., Wang, Y., Wu, T., Wu, Y., Xie, X., Xie, Z., Xie, Z., Xiong, Y., Xu, H., Xu, R. X., Xu, Y., Yang, D., You, Y., Yu, S., Yu, X., Zhang, B., Zhang, H., Zhang, L., Zhang, L., Zhang, M., Zhang, M., Zhang, W., Zhang, Y., Zhao, C., Zhao, Y., Zhou, S., Zhou, S., Zhu, Q., and Zou, Y. Deepseek llm: Scaling open-source language models with longtermism, 2024. (Cited on page 2.)

[6] Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. (Cited on page 2.)

[7] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. (Cited on page 2.)

[8] Hacohen, G., and Weinshall, D. On the power of curriculum learning in training deep networks, 2019. (Cited on page 2.)

[9] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021. (Cited on page 2.)

[10] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. (Cited on page 2.)

[11] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023. (Cited on page 2.)

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023. (Cited on page 2.)