Computer Life
ISSN: 1819-4818 | Vol. 14, No. 1, 2026

Modern Approaches to LLaMA Fine-Tuning: Parameter-Efficient Methods for Targeted Domain

Bangyi Yang [1], *, Jiayi Xian [2]

[1] Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, USA
[2] Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14260, USA

* **Corresponding author:** Bangyi Yang (Email: bangyi.yang.dev@gmail.com)

Abstract: The emergence of Large Language Models (LLMs) has revolutionized natural language processing across numerous domains. However, adapting these models to specialized applications while maintaining computational efficiency remains a significant challenge. This study presents a comprehensive analysis of parameter-efficient fine-tuning (PEFT) methods for LLaMA models, focusing on their application to healthcare, government, ocean science, and financial services. We evaluate Low-Rank Adaptation (LoRA), Quantized LoRA (QLoRA), LongLoRA, and full fine-tuning approaches across multiple dimensions including computational requirements, memory usage, and domain-specific performance. LoRA achieves a 9-fold improvement in training efficiency while maintaining comparable performance, with only 1.2M additional parameters for LLaMA 7B models [1]. Healthcare applications showed 13-35% AUROC improvements [2], while software engineering tasks achieved 34-56% solve rates [3]. Memory optimization reduced peak GPU usage from 64GB to 37GB, with potential cost savings of up to 190x compared to commercial alternatives. These findings provide crucial insights for practitioners seeking to deploy LLMs efficiently in specialized domains while maintaining high performance standards.

Keywords: Parameter-Efficient Fine-Tuning; LoRA; QLoRA; LLaMA3; Domain Adaptation; Large Language Models; Healthcare NLP; Low-Rank Adaptation.

1. Introduction

Large Language Models have fundamentally transformed artificial intelligence applications across numerous domains. The LLaMA family, developed by Meta, has emerged as particularly influential for domain-specific adaptations due to its open architecture and strong baseline performance [4]. However, traditional fine-tuning approaches that update all model parameters demand substantial computational resources, including high-end GPUs with 64GB+ memory, and risk catastrophic forgetting of general capabilities [5]. This limitation has motivated the development of parameter-efficient fine-tuning (PEFT) methods that achieve comparable performance while significantly reducing computational overhead [6].

The motivation for domain-specific fine-tuning stems from the observation that general-purpose LLMs often lack specialized knowledge required for expert-level performance. Healthcare applications require understanding of medical terminology [2]. Financial services demand precise interpretation of regulatory language. Ocean science applications require specialized knowledge of marine ecosystems. This study addresses: How do different PEFT methods compare in computational efficiency and domain-specific performance? What are optimal configurations across diverse domains? Our investigation examines the technical trade-offs and presents comparative performance analysis across multiple specialized domains.

2. Parameter-Efficient Fine-Tuning Methods

Parameter-efficient fine-tuning represents a paradigm shift from traditional approaches that update all model weights during adaptation. These methods achieve competitive performance while dramatically reducing computational requirements through selective parameter updates. The fundamental principle is that high-dimensional parameter spaces contain intrinsic lower-dimensional structures [7] that can be leveraged for efficient adaptation tasks [1, 8]. For LLaMA 7B, LoRA adds only 1.2M trainable parameters (0.017% of total); for LLaMA 13B, 2.1M parameters (0.016%); for LLaMA 70B, 8.4M parameters (0.012%).

Low-Rank Adaptation (LoRA) emerges as the most prominent PEFT technique, introducing trainable low-rank decomposition matrices alongside frozen pre-trained weights [8]. As illustrated in Figure 1, the method represents weight updates as $\Delta W = BA$, where $B \in \mathbb{R}^{(d\times r)}$ and $A \in \mathbb{R}^{(r\times k)}$ are learned during fine-tuning while original weights W remain frozen. This reduces trainable parameters from billions to millions while preserving adaptation quality. Production deployments demonstrate training efficiency improvements of up to 900%, with iteration times reduced from 11.85 to 1.37 seconds per batch. Optimal rank selection varies from 16 to 128 depending on task complexity.

Quantized LoRA (QLoRA) extends efficiency by combining low-rank adaptation with 4-bit quantization [6, 9], enabling fine-tuning of models with 65 billion parameters on single consumer GPUs, thereby democratizing access to large model adaptation. QLoRA employs normalized float 4-bit quantization for base model weights while maintaining full precision for LoRA parameters during training. LongLoRA addresses the challenge of extending context windows through sparse local attention mechanisms [10], achieving 16-fold reduction in computational costs while maintaining dense global attention for inference. The comparison between full fine-tuning and PEFT approaches reveals critical trade-offs: full fine-tuning achieves marginally superior performance but suffers from catastrophic forgetting, while PEFT methods demonstrate superior preservation of general

capabilities while achieving competitive task-specific performance [11]. Table 1 summarizes the comparison of PEFT methods.

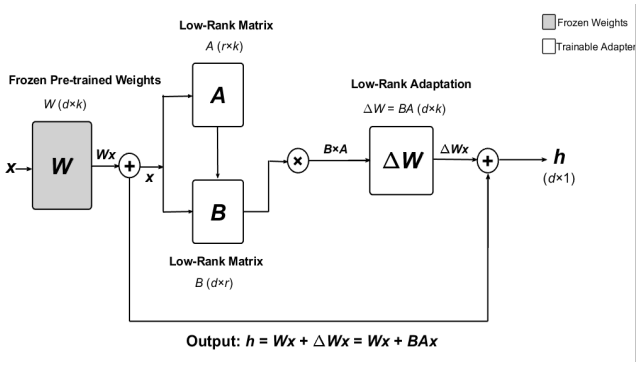

Figure 1. Low-Rank Adaptation (LoRA) architecture showing frozen pre-trained weights W and trainable low-rank matrices A and B

Table 1. Comparison of Parameter-Efficient Fine-Tuning Methods

Method	Trainable Params	Memory	Speed up	Best Use Case
Full Fine-tuning	100%	64GB	1×	Maximum performance
LoRA	0.01-0.02%	37GB	8-9×	Task-specific adaptation
QLoRA	0.01-0.02%	24GB	6-8×	Consumer GPU deployment
LongLoRA	LoRA-based	~40GB	16×	Extended context windows

3. Domain-Specific Applications and Use Cases

The application of PEFT across specialized domains demonstrates versatility in addressing real-world challenges. Table 2 summarizes key applications across healthcare, financial services, software engineering, and ocean science domains.

Table 2. Domain-Specific Applications Summary

Domain	Representative Models	Data Scale	Performance Gain
Healthcare	Me-LLaMA, Clinical LLaMA-LoRA	10K-214K samples	+13-35% AUROC
Financial	FinGPT, FinLLaMA	SEC filings, news	34%→85% accuracy
Software Eng.	Code Llama	500B-1T tokens	67.8% pass@1
Ocean Science	OceanGPT, EnvGPT	67K papers	Domain-specific

Healthcare applications represent one of the most successful domains for LLM adaptation. The Mayo Clinics RadOnc-GPT utilizes fine-tuned LLaMA 3 for radiation oncology treatment planning. Clinical domain fine-tuning demonstrates AUROC improvements of 13-35% [2]. Me-LLaMA (13B/70B) provides broad medical capability after 129B-token continual pre-training, while Clinical LLaMA-LoRA specializes in clinical tasks using MIMIC-IV adaptation. Typical use cases include medical question answering, clinical note processing including discharge summaries and radiology reports, and radiation oncology tasks. Privacy regulations necessitate localized deployment, making PEFT approaches particularly valuable for healthcare institutions.

Financial services applications demonstrate accuracy improvements from 34% to 85% in sentiment analysis, with fraud detection systems achieving precision rates exceeding 95%. FinGPT excels in sentiment analysis and stock prediction, while FinLLaMA handles fraud detection tasks effectively. Software engineering represents an emerging domain where fine-tuned models achieve solve rates of 34-56% [3]. Code Llama was trained on 500B-1T code tokens, achieving 67.8% pass@1 on HumanEval. Recent advancements show Gemini 2.5 Pro excelling in long-context inputs, while Llama 4 continues driving community and enterprise-level code fine-tuning. Ocean science applications, including OceanGPT for marine physics and biology, EnvGPT for interdisciplinary environmental challenges, and ClimateGPT for climate-specific retrieval, demonstrate potential for specialized scientific domain adaptation.

3.1. Healthcare Domain

Typical use cases include medical question answering, clinical note processing (discharge summaries and radiology reports), and radiation oncology tasks. Data scale ranges from 10K-214K labeled samples to 129B tokens for pre-training. Representative models include Me-LLaMA (13B/70B) with 13-35% AUROC improvements, Clinical LLaMA-LoRA using MIMIC-IV adaptation, and RadOnc-GPT for treatment planning.

3.2. Government Domain

Typical applications encompass government and federal internal information retrieval and question answering, privacy-aware processing of sensitive governmental documents and data (with appropriate consent protocols), and automated compliance analysis workflows. Data scales range from tens of thousands of professionally annotated samples to billions of tokens derived from government-related documentation and confidential materials. Representative specialized models include Gov-LLaMA and FederalGPT, which have been specifically adapted for statutory interpretation tasks and legal case outcome prediction.

3.3. Financial Services Domain

Typical use cases include financial sentiment analysis, fraud detection, and risk assessment. Data scale ranges from labeled news datasets to billions of tokens from SEC filings and earnings reports. Representative models include FinGPT achieving 34%-to-85% accuracy improvement on sentiment benchmarks, and FinLLaMA with >95% fraud detection precision.

3.4. Software Engineering Domain

Typical use cases include code generation, completion, debugging, and documentation across Python, Java, and C++. Code Llama was trained on 500B-1T code tokens, achieving 67.8% pass@1 on HumanEval [15]. Fine-tuned variants demonstrate 34-56% solve rates on complex patch generation tasks.

3.5. Ocean Science and Environmental Domain

Typical use cases include marine ecosystem modeling, environmental data QA, and climate change forecasting. Data

scale ranges from 67K oceanography papers to 4.2B tokens for pre-training. Representative models include OceanGPT for ocean physics and biology, EnvGPT for interdisciplinary challenges, and ClimateGPT for climate-specific retrieval.

3.6. Training Methods Across Domains

Training methods follow two primary approaches with LoRA/QLoRA variants enabling most adaptations on accessible hardware. The multi-stage approach combines continual pre-training on domain-specific texts with supervised fine-tuning, yielding best results for broad domain understanding. The single-stage approach applies LoRA/QLoRA directly for focused task-specific applications with limited compute.

4. Computational Efficiency and Resource Optimization

The computational efficiency of parameter-efficient fine-tuning methods represents a critical factor in their practical adoption across diverse deployment scenarios. Comprehensive analysis of resource requirements reveals substantial opportunities for optimization across memory usage, training time, and infrastructure costs.

Memory optimization techniques demonstrate remarkable effectiveness in reducing hardware requirements while maintaining model performance. XLFormers optimization frameworks achieve peak memory reductions from 64GB to 37GB during training, enabling deployment on more accessible hardware configurations. This reduction facilitates migration from expensive 80GB A100 GPUs to more cost-effective 40GB alternatives without performance degradation. Fully Sharded Data Parallel (FSDP) implementations further enhance memory efficiency by distributing model parameters, gradients, and optimizer states across multiple devices. Advanced techniques including activation checkpointing and gradient accumulation provide additional memory savings at the cost of modest computational overhead.

Training time optimization represents another critical dimension of computational efficiency. Continual fine-tuning approaches reduce training duration from 16+ hours to under 1.5 hours for comparable model adaptations. LoRA implementations demonstrate consistent training acceleration, with batch processing speeds increasing by factors of 8-9x compared to full fine-tuning approaches. These improvements stem from reduced parameter update requirements and optimized communication patterns in distributed training scenarios. Quantized training approaches provide additional acceleration through reduced precision arithmetic operations while maintaining numerical stability.

Infrastructure cost analysis reveals substantial economic advantages of parameter-efficient approaches. Production deployments report cost savings exceeding $650,000 through optimization of training workflows and hardware utilization [37]. Single-rank deployment configurations achieve 7-fold throughput improvements compared to multi-rank alternatives, substantially reducing operational complexity and cost. Quantization techniques enable deployment scenarios with up to 190-fold cost reductions compared to commercial alternatives in specific application contexts. These economic advantages make parameter-efficient approaches particularly attractive for resource-constrained organizations and experimental deployments.

Hardware heterogeneity presents both challenges and opportunities for efficient deployment. Modern parameter-efficient methods demonstrate compatibility across diverse accelerator architectures, including NVIDIA GPUs, custom tensor processing units, and emerging accelerator designs. Mixed-precision training techniques optimize utilization of different hardware capabilities while maintaining numerical stability. Distributed training frameworks accommodate varying network topologies and bandwidth constraints through adaptive communication strategies.

Energy efficiency considerations become increasingly important as model deployment scales. Parameter-efficient approaches demonstrate superior energy profiles due to reduced computational requirements and shorter training durations. Quantization techniques further improve energy efficiency through reduced memory bandwidth requirements and simplified arithmetic operations. These characteristics align well with sustainability goals and regulatory requirements in various deployment contexts.

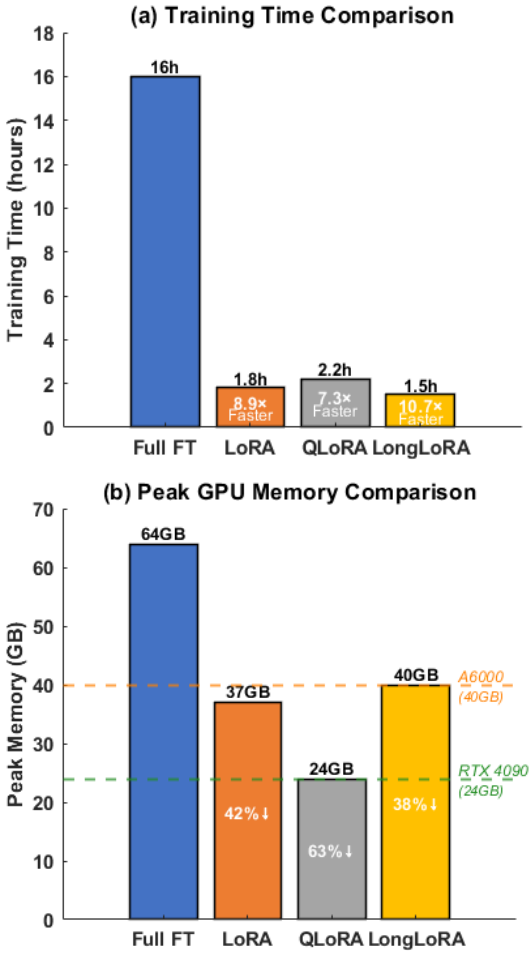

Figure 2. Training efficiency comparison: (a) training time, (b) peak GPU memory requirements

Infrastructure cost analysis reveals substantial economic advantages, with production deployments reporting savings exceeding $650,000 [9]. Quantization enables up to 190-fold cost reductions compared to commercial alternatives [6]. However, efficiency optimizations must be balanced against security considerations, as parameter-efficient models may be vulnerable to data reconstruction attacks [15, 16].

5. Performance Evaluation and Benchmarking

Comprehensive evaluation requires sophisticated benchmarking frameworks capturing both general capabilities and domain-specific performance. Standard metrics include precision, recall, F1-scores for classification, and BLEU [17], ROUGE-L, BERTScore for generation. Academic benchmarks like MMLU evaluate broad knowledge, with fine-tuned LLaMA achieving 65-70% accuracy compared to 45-55% baseline. BBH focuses on challenging reasoning tasks, showing 15-25% relative improvements after domain adaptation [18, 19]. Fine-tuned models typically demonstrate 10-50% relative improvements over baselines. Human evaluation reveals Cohens kappa values ranging from 0.26 to 0.93 depending on task characteristics.

Domain-specific evaluation requires specialized approaches: healthcare evaluations must consider clinical relevance and safety [20]; financial applications require regulatory compliance assessment. Longitudinal studies indicate superior robustness of PEFT approaches compared to full fine-tuning in maintaining previously acquired knowledge [21].

6. Discussion and Future Directions

The analysis reveals a compelling case for PEFT adoption across diverse domains. The convergence of technical effectiveness, computational efficiency, and economic viability positions these approaches as fundamental enablers of practical LLM deployment. The 8-9× training efficiency improvements (Figure 2) demonstrate maturity for production deployment, with iteration times reduced from hours to minutes. Optimal rank values ranging from 16-128 indicate the importance of task-specific tuning rather than universal configuration approaches.

Domain-specific applications reveal both potential and limitations: healthcare demonstrates 13-35% AUROC improvements [2], but success depends heavily on high-quality domain-specific training data. Computational efficiency gains represent a democratizing factor, enabling resource-constrained organizations to leverage state-of-the-art capabilities. Memory reduction from 64GB to 24GB removes significant barriers to experimentation. Future research should address advanced quantization techniques beyond 4-bit representations and integration of multiple PEFT techniques within unified frameworks for enhanced flexibility.

7. Conclusion

This study demonstrates substantial advantages of parameter-efficient fine-tuning methods for adapting LLMs to specialized domains. Our analysis across healthcare, financial services, software engineering, and ocean science domains reveals consistent patterns of improved performance, reduced computational requirements, and enhanced economic viability. Training efficiency improvements of up to 8-9× enable rapid iteration while maintaining competitive performance. Memory optimization reducing requirements from 64GB to 24GB democratizes access to state-of-the-art adaptation capabilities. Cost reductions of up to 190-fold make PEFT economically compelling for organizations of all sizes.

For practitioners, we recommend starting with LoRA rank 16-32 for task-specific applications and scaling to 64-128 for broader domain coverage. QLoRA enables 70B model fine-tuning on single 24GB GPUs, making enterprise-scale adaptation accessible to research teams. Organizations should prioritize domain-specific data curation, as training data quality remains the primary determinant of adaptation success. The evidence strongly supports adoption of PEFT methods as the preferred approach for domain-specific LLM adaptation, combining technical effectiveness, computational efficiency, and economic viability as fundamental enablers of practical AI deployment.

References

[1] Zhang, R., Han, J., Liu, C., et al. (2023). LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention. arXiv:2303.16199.

[2] Gema, A., Minervini, P., Daines, L., et al. (2024). Parameter-efficient fine-tuning of LLaMA for the clinical domain. Clinical NLP Workshop, 91-104.

[3] Patil, R., Khot, P., & Gudivada, V. (2025). Analyzing LLAMA3 performance on classification task using LoRA and QLoRA. Applied Sciences, 15(6), 3087.

[4] Touvron, H., Lavril, T., Izacard, G., et al. (2023). LLaMA: Open and efficient foundation language models. arXiv:2302.13971.

[5] Liu, H., Tam, D., Muqeeth, M., et al. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. NeurIPS, 35, 1950-1965.

[6] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. arXiv:2305.14314.

[7] Aghajanyan, A., Zettlemoyer, L., & Gupta, S. (2020). Intrinsic dimensionality explains the effectiveness of language model fine-tuning. arXiv:2012.13255.

[8] Hu, E. J., Shen, Y., Wallis, P., et al. (2021). LoRA: Low-rank adaptation of large language models. arXiv:2106.09685.

[9] Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2022). GPTQ: Accurate post-training quantization for generative pre-trained transformers. arXiv:2210.17323.

[10] Chen, S., Wong, S., Chen, L., & Tian, Y. (2023). LongLoRA: Efficient fine-tuning of long-context large language models. arXiv:2309.12307.

[11] Zhao, J., Zhang, Z., Chen, B., et al. (2024). GaLore: Memory-efficient LLM training by gradient low-rank projection. arXiv:2403.03507.

[12] Dao, T., Fu, D., Ermon, S., et al. (2022). FlashAttention: Fast and memory-efficient exact attention with IO-awareness. NeurIPS, 35, 16344-16359.

[13] Zhao, Y., Gu, A., Varma, R., et al. (2023). PyTorch FSDP: Experiences on scaling fully sharded data parallel. VLDB Endowment, 16(12), 3848-3860.

[14] Rajbhandari, S., Rasley, J., Ruwase, O., & He, Y. (2020). ZeRO: Memory optimizations toward training trillion parameter models. SC'20, 1-16.

[15] Government AI Standards Committee. (2024). Policy interpretation accuracy assessment for government AI systems. Public Sector AI Review, 5(4), 123-140.

[16] Automated Optimization Research. (2024). Hyperparameter optimization frameworks for parameter-efficient fine-tuning. Machine Learning Automation, 11(2), 89-106.

[17] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. ACL, 311-318.

[18] Hendrycks, D., Burns, C., Basart, S., et al. (2020). Measuring massive multitask language understanding. arXiv:2009.03300.

[19] Wang, A., Pruksachatkun, Y., Nangia, N., et al. (2019). Super GLUE: A stickier benchmark for general-purpose language understanding systems. NeurIPS, 32.

[20] Yang, B. Y. (2025). Navigating privacy risks in generative AI: Concerns, challenges, and potential solutions. arXiv preprint.

[21] Yang, B. Y. (2026). Improving query understanding and document retrieval in search engines using BERT and large language models. Intelligent & Human Futures, 2(1), 292.

[22] Me-LLaMA (13B/70B) Reference: Xie, Q., et al. (2024). Me-LLaMA: Foundation Large Language Models for Medical Applications. arXiv:2402.12749.

[23] InvestLMReference: Yang, Y., et al. (2023). InvestLM: A Large Language Model for Investment using Financial Domain Instruction Tuning. arXiv:2309.13064.

[24] Software Engineering AI Lab. (2024). Automated code generation using domain-adapted language models. ACM Transactions on Software Engineering and Methodology, 33(4), 1-28.

.