Int. J. Sci. R. Tech., 2026 3(7)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

Parameter-Efficient Fine Tuning Technique In Lightweight Large Language Models: A Comprehensive Review Of Architectures, Techniques, Applications, Challenges And Future Directions

Mahiba Jeyaraj*, Sujata Deshmukh

Department of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, Mumbai University, Maharashtra, India.

ABSTRACT

The use of PEFT with Lightweight LLMs had started from 2021 with the introduction of LoRA techniques. Making and adapting the large language models (LLMs) to new tasks through full fine-tuning is often considered computationally high, particularly in lightweight, edge-deployed variants. Thus the Parameter-efficient fine-tuning (PEFT) technique has emerged as a solution where updating of only a small fraction of parameters are done while the pretrained architecture of the model remains frozen. This review integrates the PEFT literature published between 2024 to 2026 and organizes the methods into six families: additive, reparameterization, quantization-aware, selective/sparse, gradient-projection and mixture-of-experts approaches. The recent advances including the DoRA, AdaLoRA, VeRA, QLoRA and its successors, GaLore-style gradient projection and sparse LoRA mixture-of-experts architectures are summarized along with open challenges like the residual accuracy gap relative to full fine-tuning, hyperparameter sensitivity, adapter-quantization interaction and multi-adapter serving. Applications such as the on-device inference, federated personalization, healthcare, multimodal instruction tuning and multilingual adaptation are also reviewed. A comparative table summarizes the trainable-parameter and inference overhead across the different representative methods. The review concludes that PEFT techniques has become the primary mechanism for adapting lightweight LLMs to real-world use and identifies automatic configuration search, hardware-software co-design and by standardized benchmarking as priorities for future work.

Keywords: Parameter-efficient fine-tuning; Low-rank adaptation; LoRA; Quantization; Lightweight language models.

INTRODUCTION

The rapid scaling of large language models has produced systems with remarkable capabilities, but it has widened the gap between what state-of-the-art models can do and what ordinary practitioners can afford to fine-tune. Full fine-tuning of an LLM requires storing and updating billions of parameters, along with their optimizer states and gradients, which in practice can demand memory many times larger than the model's own footprint. This cost is especially limiting for lightweight LLMs intended for edge devices, mobile phones, or single-GPU research settings, where both training and inference must operate under strict memory, latency, and energy budgets.

Parameter-efficient fine-tuning (PEFT) addresses this tension by freezing the majority of a pretrained model's weights and introducing a small number of trainable parameters, through additional modules, low-rank updates, or selective parameter subsets, that are optimized for a downstream task. Since the introduction of adapter tuning[1] and the popularization of Low-Rank Adaptation (LoRA)[2], PEFT has become the default strategy for customizing LLMs efficiently. The last two years have seen an especially rapid expansion of this space[3,4], with dozens of LoRA variants, quantization-aware adapters, gradient-projection methods, and mixture-of-experts extensions proposed at major venues throughout 2024, 2025, and into 2026.

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Alongside adapter-based approaches, a parallel line of work has revisited how the optimizer itself is memory-constrained. GaLore[28] showed that projecting gradients, rather than weights, into a low-rank subspace allows full-parameter training to proceed under a fraction of the usual optimizer memory, and follow-up work such as Q-GaLore[29] combined this idea with INT4 projection matrices to push the memory floor even lower. This expands the PEFT taxonomy beyond adapter- and weight-centric methods into optimizer- and gradient-centric ones, which we incorporate into the taxonomy below.

This review focuses specifically on the intersection of PEFT and lightweight LLMs: models and adaptation methods designed to run and be trained under constrained compute, memory, or connectivity. It provides an updated taxonomy of PEFT methods reflecting work published since 2024, consolidates the practical challenges that remain unresolved, and surveys emerging applications, including multimodal and clinical instruction tuning, that illustrate why PEFT is central to making lightweight LLMs practically useful, closing with directions the field appears to be converging on.

2. METHODOLOGY

This review was conducted through a structured literature search on various scopus indexed databases like arXiv, ACL Anthology, IEEE Xplore and Springer/Nature indexed using the search terms such as “parameter-efficient fine-tuning”, “low-rank adaptation”, “LoRA”, “QLoRA”, “quantized fine-tuning”, “gradient low-rank projection”, “mixture of LoRA experts”, “multimodal parameter-efficient fine-tuning” and “LLM fine-tuning”. The search was limited to papers published or revised between 2021 to July 2026 where priority were given with substantial citation traction or from established research groups. All the research sources were selected for detailed review, collecting their method-development papers, empirical comparison studies and application-focused on generating reports.

2.1 Comparative Summary of Representative Methods

Table 1 compares all the trainable-parameter, inference-time overhead and core mechanisms of representative PEFT methods as discussed above.

Method	Family	Trainable Params	Inference Overhead	Key Idea
Adapter tuning	Additive	~1-4% of backbone	Small (extra layers)	Bottleneck FFN modules inserted between transformer layers
Prefix / Prompt tuning	Additive	<0.1%	Small (extra tokens)	Trainable continuous vectors prepended to inputs/hidden states
LoRA	Reparameterization	0.1-1%	None (mergeable)	Low-rank matrices B, A approximate the weight update
DoRA	Reparameterization	~0.1-1%	None (mergeable)	Decomposes W into magnitude and direction; low-rank update on direction only
AdaLoRA	Reparameterization	0.1-1% (adaptive)	None (mergeable)	Importance-based dynamic rank

				allocation across layers
VeRA	Reparameterization	<0.1%	None (mergeable)	Shared frozen random low-rank matrices with trained scaling vectors
QLoRA	Quantization-aware	0.1-1% (adapters)	None (adapters mergeable)	4-bit NormalFloat backbone + double quantization + LoRA adapters
BitFit	Selective/Sparse	<0.1%	None	Only bias terms are updated
IA3	Selective/Sparse	<0.05%	Minimal	Learned per-layer rescaling vectors on activations
GaLore	Gradient-projection	Full-rank weights, low-rank grad.	None (no adapters at inference)	Projects gradients/optimizer states into a low-rank subspace
LoRAMoE / MixLoRA	Mixture-of-experts	Scales with # experts	Moderate (routing)	Multiple LoRA experts routed per token or task

Table 1: Comparison of representative parameter-efficient fine-tuning methods.

The methods were grouped into taxonomy families which are based on their underlying mechanism such as additive, reparameterization-based, quantization-aware, selective, gradient-projection or mixture-of-experts and cross-cutting challenges and applications were extracted by synthesis of research across the selected sources where methods reported to overlapping trainable-parameter and inference-overhead characteristics. These were consolidated into a single comparative table (Table 1) to support direct comparison.

2.2 *Taxonomy of Parameter-Efficient Fine-Tuning Methods*

PEFT methods can be organized into six broad families based on how they introduce trainable capacity: additive methods, reparameterization-based (low-rank) methods, quantization-aware methods, selective/sparse methods, gradient-projection methods, and mixture-of-experts extensions. Figure 1 summarizes this taxonomy and representative methods within each family.

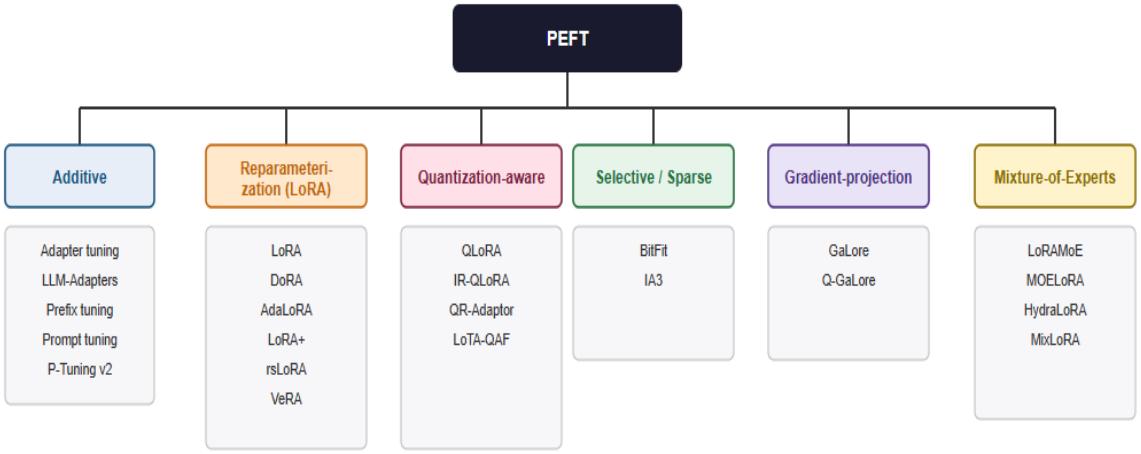

Figure 1: Taxonomy of parameter-efficient fine-tuning methods for lightweight LLMs.

2.3 *Why PEFT Matters for Lightweight LLMs*

Lightweight LLMs, typically ranging from under one billion to roughly eight billion parameters, are designed to fit within the memory and compute envelopes of consumer GPUs, laptops, and increasingly, smartphones. Even at this reduced scale, full fine-tuning remains expensive relative to available hardware: optimizer states for adaptive methods such as Adam roughly double or triple the memory required beyond the model weights themselves, and every fine-tuned variant of the model must be stored separately. The PEFT methods reduces the trainable parameters typically by one percent of the base model which corresponds to shrink the optimizer state, gradient memory and checkpoint size.

Beyond memory savings, PEFT offers flexibility because the base model is frozen where a single pretrained backbone can support many lightweight, swappable adapters for different tasks, users or domains without duplicating the full architecture of the model. This property is particularly valuable for lightweight LLMs deployed at the edge, and it has made PEFT the natural fit for federated learning, where clients must exchange only small adapter updates rather than full model weights.

2.4 *Different Methods and Taxonomy of PEFT techniques*

2.4.1 Additive Methods

Additive methods insert new trainable components into an otherwise frozen network. Adapter tuning[1], later extended for LLMs through adapter families such as LLM-Adapters[5], inserts small bottleneck feed-forward modules between transformer layers. Prefix tuning[6] and prompt tuning[7] instead prepend trainable continuous vectors to the input or hidden states at each layer, leaving the transformer body untouched. P-Tuning v2[8] showed that deep prompt tuning across all layers can match full fine-tuning performance across a range of scales and tasks. These methods are simple to implement and support multi-task composition, but they typically add a small amount of inference latency because the extra modules or tokens must be processed at run time.

2.4.2 Selective and Sparse Methods

Selective methods fine-tune a small, chosen subset of the existing parameters rather than adding new ones. BitFit[18] updates only the bias terms of a transformer, achieving competitive performance on many tasks with an extremely small trainable footprint. IA3[19] instead learns a small set of per-layer rescaling vectors that multiply activations, keeping trainable parameters minimal. These approaches are attractive for the most constrained lightweight deployments, but they generally trail low-rank methods in expressiveness on more complex tasks.

2.4.3 Gradient-Projection and Hybrid Optimizer-Centric Methods

A distinct branch of memory-efficient training targets the optimizer rather than the weights or activations directly. GaLore[28] observes that the gradient matrix of a linear layer becomes approximately low-rank over

the course of training, and periodically computes a low-rank projection of the gradient via SVD, projecting the optimizer's moment estimates into this subspace rather than storing them at full rank. Unlike LoRA, GaLore keeps the model's full-rank weights trainable, so it does not introduce the small residual accuracy gap that low-rank adapters can leave relative to full fine-tuning, while still reducing optimizer-state memory substantially. Q-GaLore[29] extends this idea with INT4-quantized projection matrices and layer-adaptive low-rank gradients, further lowering the memory floor enough to fine-tune a 7-billion-parameter model on a 16 GB consumer GPU. Because gradient-projection methods and adapter-based methods make different memory-versus-fidelity trade-offs, several 2025-2026 papers have begun exploring hybrids that apply GaLore-style projection to some layers and LoRA-style adapters to others, though this remains an early-stage research direction.

2.4.4 Mixture-of-Experts and Multi-Adapter Extensions

A fast-growing branch of PEFT research combines low-rank adapters with mixture-of-experts (MoE) routing, treating individual LoRA modules as experts that are selectively activated per token or per task. LoRAMoE[20] introduces localized load-balancing routing to mitigate catastrophic forgetting during instruction tuning, while MOELoRA[21] targets multi-task settings by learning to route among task-specialized adapters. HydraLoRA[22] proposes an asymmetric architecture with a shared down-projection and multiple specialized up-projections, reducing parameter duplication. MixLoRA[23] targets high-throughput serving of many adapters. These MoE-PEFT hybrids improve multi-task capacity but most cannot be losslessly merged back into the base model, reintroducing some of the inference overhead that plain LoRA was designed to avoid.

2.4.5 *Quantization-Aware PEFT*

Quantization-aware PEFT methods combine low-rank adaptation with reduced-precision storage of the frozen backbone, directly targeting the memory bottleneck that limits fine-tuning of larger models on constrained hardware. QLoRA[14] quantizes the pretrained model to 4-bit precision using a NormalFloat data type, applies double quantization to compress the quantization constants themselves, and uses paged optimizers to avoid memory spikes, enabling fine-tuning of models with tens of billions of parameters on a single consumer-class GPU while closely matching full 16-bit fine-tuning performance. Figure 3 illustrates the end-to-end QLoRA pipeline and its principal follow-on refinements.

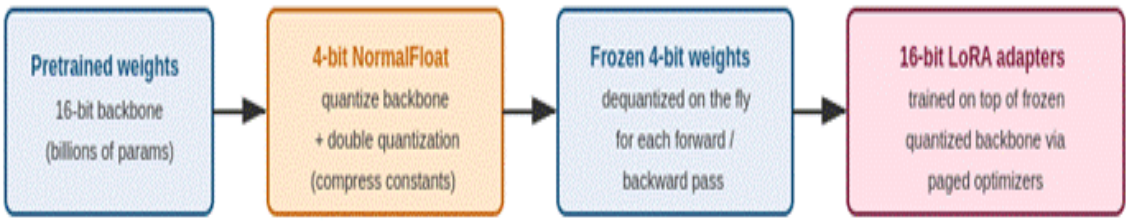

Figure 3: Quantization-aware PEFT: the QLoRA pipeline and its refinements.

Because quantization introduces representation error that interacts poorly with naively initialized low-rank adapters, follow-up methods refine how adapters are initialized or trained on top of quantized weights. IR-QLoRA[15] uses entropy-maximizing quantization together with data-driven adapter initialization to better absorb quantization error. QR-Adaptor[16] performs a joint, gradient-free search over per-layer bit-width and rank, reporting accuracy that can exceed both standard QLoRA and full-precision LoRA at matched memory budgets. LoTA-QAF[17] targets lossless ternary adaptation so that adapters can be merged directly into quantized weights without precision loss.

2.4.6 *Reparameterization-Based Methods: LoRA and Its Variants*

Low-Rank Adaptation (LoRA)[2] represents the weight update for a linear layer as the product of two small low-rank matrices, which are trained while the

original weight matrix stays frozen; after training, the low-rank update can be merged into the original weights, adding no inference overhead. This property has made LoRA the most widely adopted PEFT method and the basis for a large family of successors published through 2024–2026. Figure 2 explains the core LoRA update rule with DoRA's magnitude-direction decomposition.

Several variants target the residual accuracy gap between LoRA and full fine-tuning. DoRA[9] decomposes the pretrained weight matrix into magnitude and direction components, applying low-rank updates only to the directional component, which more closely mirrors how full fine-tuning updates weights and yields consistent gains over vanilla LoRA, particularly at low ranks, without added inference cost.

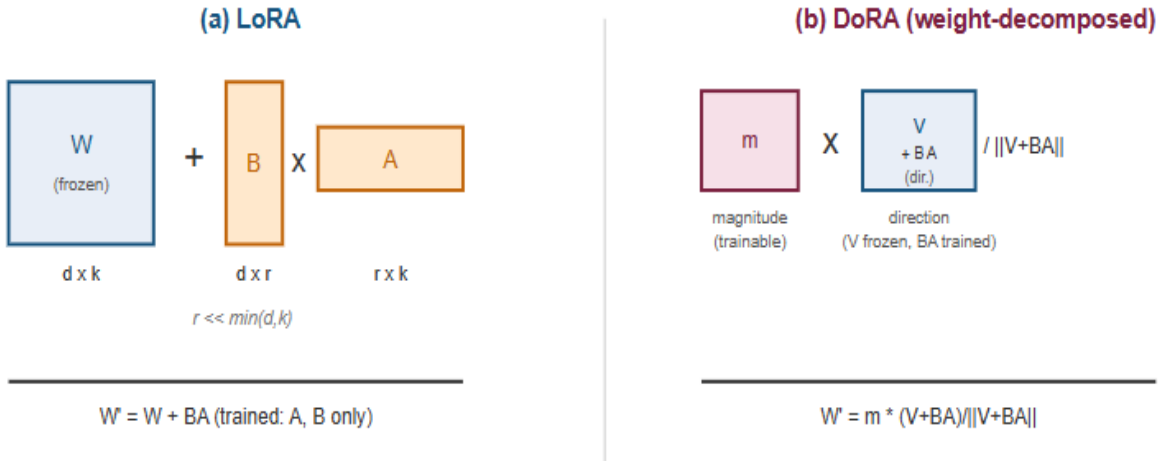

Figure 2: LoRA vs. DoRA low-rank weight update mechanism.

Other variants address the structural inefficiencies in the original LoRA formulation. The AdaLoRA[10] adjusts and allocates the rank budget across the layers based on importance scores rather than fixing an uniform rank. LoRA+[11] shows that using different learning rates helps LoRA's two matrices improve the convergence. rsLoRA[12] explains about the scaling factor applied to low-rank updates by proposing a rank-stabilized scaling rule so that the performance does not degrade when the rank increases. VeRA[13] shares a single pair of frozen random low-rank matrices across all layers and trains only lightweight per-layer scaling vectors, substantially cutting trainable parameters relative to standard LoRA.

3. RESULTS AND DISCUSSION

3.1 Multimodal and Cross-Domain Applications of PEFT

PEFT has also become the default adaptation strategy for multimodal large language models (MLLMs), where a vision or audio encoder is connected to a frozen language backbone and only a lightweight connector plus LoRA-style adapters on the language model are trained. PALO[30], a multilingual vision-language model built on the LLaVA architecture, applies LoRA to the language model component during instruction tuning across ten languages, reporting substantial gains for low-resource languages while keeping training cost close to that of monolingual fine-tuning. In clinical applications, LDP[31] fine-tunes a Qwen2-VL backbone with LoRA and aligns outputs to clinical reporting standards, reducing training compute by several hundred-fold relative to full fine-tuning while matching or exceeding baseline report-generation quality on expert evaluation. A broader survey of PEFT for pretrained vision models[32] documents a parallel line of adapter, prompt, and low-rank methods developed specifically for vision transformers, many of which have since been re-purposed for the vision encoders inside MLLMs. Together, these results indicate that the parameter-efficiency principles established for text-only LLMs transfer directly to multimodal settings, where the cost of full fine-tuning is compounded by the size of both the language and perception backbones.

3.2 Challenges

Despite rapid progress in the field of Lightweight LLMs there still are several challenges that comes across the PEFT literature such as:

• Residual accuracy gap: theoretical and empirical comparisons[24] continue to find that PEFT methods do not always match full fine-tuning performance on harder tasks, motivating capacity-expanding variants such as DoRA and higher-rank, MoE-based, or gradient-projection approaches.

• Multi-adapter serving and composition: efficiently storing, routing, and merging large adapter populations at inference time, without linear growth in memory or latency, remains an active systems problem.

• Hallucinating, forgetting and interference: sequential or multi-task adaptation can degrade previously learned capabilities, motivating routing-based mitigations such as LoRAMoE's load-balancing scheme.

• Heterogeneity in federated and edge settings: when PEFT is deployed across federated clients or heterogeneous devices, differences in data distribution and compute budget across clients complicate aggregation[25].

• Evaluation standardization: the number of LoRA variants published since 2024 has outpaced the availability of standardized, comparable benchmarks.

3.3 Applications

A defining trend of 2025–2026 is the movement of PEFT from server-side training toward genuine on-device execution. Frameworks such as MobileFineTuner[26] provide end-to-end pipelines for fine-tuning small models directly on commodity phones under tight RAM budgets, supporting both full and LoRA-based fine-tuning.

Because PEFT updates are small, they are naturally suited to federated learning. Heterogeneous LoRA methods[27] allow clients with different compute budgets to train adapters of different ranks while still contributing to a shared global model, addressing heterogeneity identified as a central open problem in federated foundation models[25].

PEFT, and QLoRA in particular, has also enabled lightweight domain adaptation in resource-constrained institutional settings such as clinical decision support, and has lowered the barrier to adapting LLMs for low-resource and non-Latin-script languages, where full fine-tuning is rarely feasible for smaller research groups. As reviewed above, this extends naturally to multimodal clinical and multilingual vision-language applications, where PEFT keeps both the language and perception components trainable within a single consumer-GPU budget.

3.4 Future Directions

Automatic and joint configuration search, over rank, bit-width, and layer placement simultaneously[16], is likely to replace today's largely manual tuning. Hardware-software co-design for on-device training[26] will need to mature to handle larger lightweight models within mobile power and thermal budgets. Unifying PEFT with mixture-of-experts routing while preserving LoRA's losslessly-mergeable, zero-overhead deployment property remains unresolved, and federated PEFT will likely continue to develop rank- and resource-heterogeneous protocols so that low-end edge devices can meaningfully participate. Gradient-projection methods such as GaLore and Q-GaLore point toward a further convergence of adapter-based and optimizer-based memory efficiency, and hybrids that combine the two remain an open area. Finally, the field would benefit from standardized, task-diverse benchmarks and reporting practices for PEFT methods, particularly as multimodal and multilingual applications broaden the evaluation surface beyond text-only NLP tasks.

CONCLUSION

The Parameter-efficient fine-tuning has changed from being a memory-saving methods to a foundational requirement for making the lightweight LLMs use them for trainable and deployable with spanning cloud fine-tuning of small models, federated personalization, multimodal instruction tuning and making it available on-device adaptation on phones. The year range from 2024 to 2026 has seen a vast amount of progress in the field AI to expand its research beyond the original LoRA formulation with weight-decomposition methods, quantization-aware

approaches, gradient-projection techniques and mixture-of-experts extensions by addressing all different facts and solution for obtaining the accuracy-efficiency trade-off. The challenges that still remains is closing the difference in the accuracy gap, automating the configuration choices, serving many adapters efficiently, mitigating hallucination or accidental forgetting and standardizing the evaluation which are less about whether PEFT is useful and talks more about how far its efficiency and advantages can be pushed without sacrificing the capability that motivated fine-tuning processes.

ACKNOWLEDGEMENT

The authors of the research paper declare that no external funding was received for this work and also like thank their institution for providing access to the literature review databases used for this review.

REFERENCES

1. Wang L, Chen S, Jiang L, Pan S, Cai R, Yang S. Parameter-efficient fine-tuning in large language models: a survey of methodologies. Knowl Inf Syst. 2025.
2. Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. QLoRA: efficient finetuning of quantized LLMs. Adv Neural Inf Process Syst. 2026.
3. Han Z, Gao C, Liu J, Zhang J, Zhang SQ. Parameter-efficient fine-tuning for large models: a comprehensive survey. arXiv preprint arXiv:2403.14608; 2024.
4. Zhou T, Xiang S, et al. LDP: parameter-efficient fine-tuning of multimodal LLM for medical report generation. arXiv preprint arXiv:2512.10750; 2025.
5. Zhou Y, et al. QR-Adaptor: efficient adapter fine-tuning via joint bitwidth and rank optimization. arXiv preprint arXiv:2505.00385; 2025.
6. Li XL, Liang P. Prefix-tuning: optimizing continuous prompts for generation. Proc ACL. 2021.
7. Xu Z, et al. LoTA-QAF: lossless ternary adaptation for quantization-aware fine-tuning. arXiv preprint arXiv:2505.18724; 2025.
8. Liu X, Ji K, Fu Y, Tam W, Du Z, Yang Z, et al. P-Tuning v2: prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602; 2021.
9. Liu SY, Wang CY, Yin H, Molchanov P, Wang YCF, Cheng KT, et al. DoRA: weight-decomposed low-rank adaptation. Proc ICML. 2024;32100-21.
10. Zhang Q, Chen M, Bukharin A, Karampatziakis N, He P, Cheng Y, et al. AdaLoRA: adaptive budget allocation for parameter-efficient fine-tuning. Proc ICLR. 2023.
11. Hayou S, Ghosh N, Yu B. LoRA+: efficient low rank adaptation of large models. arXiv preprint arXiv:2402.12354; 2024.
12. Kalajdzievski D. A rank stabilization scaling factor for fine-tuning with LoRA. arXiv preprint arXiv:2312.03732; 2023.
13. Kopiczko DJ, Blankevoort T, Asano YM. VeRA: vector-based random matrix adaptation. Proc ICLR. 2024.
14. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: low-rank adaptation of large language models. Proc ICLR. 2022.
15. Qin H, Ma X, Zheng X, Li X, Zhang Y, Liu S, et al. IR-QLoRA: accurate LoRA-finetuning quantization of LLMs via information retention. arXiv preprint arXiv:2402.05445; 2024.
16. Hu Z, Wang L, Lan Y, Xu W, Lim EP, Bing L, et al. LLM-Adapters: an adapter family for parameter-efficient fine-tuning of large language models. Proc EMNLP. 2023;5254-76.
17. Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning. Proc EMNLP. 2021.
18. Zaken EB, Ravfogel S, Goldberg Y. BitFit: simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199; 2021.
19. Liu H, Tam D, Muqeeth M, Mohta J, Huang T, Bansal M, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. Adv Neural Inf Process Syst. 2022;35.
20. Dou S, Zhou E, Liu Y, Gao S, Shen W, Xiong L, et al. LoRAMoE: revolutionizing mixture of experts for maintaining world knowledge in language model alignment. arXiv preprint arXiv:2312.09979; 2023.
21. Liu Q, et al. MOELoRA: an MoE-based parameter efficient fine-tuning method for multi-task medical applications. arXiv preprint arXiv:2310.18339; 2023.

22. Tian C, Shi Z, Guo Z, Li L, Xu C. HydraLoRA: an asymmetric LoRA architecture for efficient fine-tuning. arXiv preprint arXiv:2404.19245; 2024.
23. Li D, et al. MixLoRA: enhancing large language models fine-tuning with LoRA-based mixture of experts. arXiv preprint arXiv:2404.15159; 2024.
24. Liu Y, Xu X, Nie E, Wang Z, Feng S, Wang D, et al. Look within or look beyond? A theoretical comparison between parameter-efficient and full fine-tuning. arXiv preprint arXiv:2505.22355; 2025.
25. Fan T, Gu H, Cao X, Chan CS, Chen Q, Chen Y, et al. Ten challenging problems in federated foundation models. IEEE Trans Knowl Data Eng. 2025;37(7):4314-37.
26. Geng J, Zhao L, Lu Y, Luo B. MobileFineTuner: a unified end-to-end framework for fine-tuning LLMs on mobile phones. arXiv preprint arXiv:2512.08211; 2025.
27. Cho YJ, Liu L, Xu Z, Fahrezi A, Barnes M, Joshi G. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. Proc EMNLP. 2024;12903-13.
28. Zhao J, Zhang Z, Chen B, Wang Z, Anandkumar A, Tian Y. GaLore: memory-efficient LLM training by gradient low-rank projection. Proc ICML. 2024.
29. Zhang Z, Jaiswal A, Yin L, Liu S, Zhao J, Tian Y, Wang Z. Q-GaLore: quantized GaLore with INT4 projection and layer-adaptive low-rank gradients. arXiv preprint arXiv:2407.08296; 2024.
30. Rasheed H, Maaz M, Shaji A, Shaker A, Khan S, Cholakkal H, et al. PALO: a polyglot large multimodal model for 5B people. arXiv preprint arXiv:2402.14818; 2024.
31. Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, de Laroussilhe Q, Gesmundo A, et al. Parameter-efficient transfer learning for NLP. Proc ICML. 2019.
32. Xin Y, Luo S, Zhou H, Du J, Liu X, Fan Y, Li Q, Du Y. Parameter-efficient fine-tuning for pre-trained vision models: a survey. arXiv preprint arXiv:2402.02242; 2024.

HOW TO CITE: Mahiba Jeyaraj*, Sujata Deshmukh, Parameter-Efficient Fine Tuning Technique In Lightweight Large Language Models: A Comprehensive Review Of Architectures, Techniques, Applications, Challenges And Future Directions, Int. J. Sci. R. Tech., 2026, 3 (7), 1085-1093. https://doi.org/10.5281/zenodo.21701977