Musale Devendra Kailas, 2026, 14:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Fine-Tuning Large Language Models: A Review Of Full, Parameter-Efficient And Instruction Tuning For Domain-Specific Applications

Musale Devendra Kailas[1], Prof. J. M. Patade[2], Prof. M. C. Kolse[3], Prof. S. K. Said[4]
Department of Artificial Intelligence and Data Science
Jaihind College of Engineering
Pune, Maharashtra, India

Abstract—Large Language Models (LLMs) understand and generate human-like text, but general-purpose models often lack the vocabulary, knowledge and response style needed for specific domains such as healthcare, law, agriculture or customer support. Fine-tuning adapts a pre-trained model to a target task by continuing training on domain-specific data, which is far cheaper than training from scratch. This paper reviews fine-tuning methods for LLMs. It traces transfer learning from universal language model fine-tuning to text-to-text models and explains full fine-tuning, parameter- efficient fine-tuning with adapters, prefix and prompt tuning and low-rank adaptation, quantized fine-tuning, and instruction and preference tuning that align models with human expectations. Tools such as Hugging Face Transformers and PyTorch, and models such as BERT, GPT-style and LLaMA, are discussed. A five-stage proposed pipeline covering dataset collection, preprocessing, model selection, fine-tuning and evaluation is presented as a research design; no implementation results or accuracy claims are made. Research gaps, objectives, a comparison of fine-tuning methods and challenges including data quality, catastrophic forgetting, overfitting, compute cost, evaluation and safety are discussed. The review concludes that parameter-efficient methods make domain adaptation accessible to students and small organizations.

Index Terms—Large language models, fine-tuning, transfer learning, parameter-efficient fine-tuning, LoRA, instruction tuning, Hugging Face, domain adaptation

I. INTRODUCTION

Pre-trained language models learn general language patterns from vast text collections. When a college wants a chatbot that answers questions about its own syllabus, or a hospital wants a model that summarizes discharge notes in its format, the general model is a starting point rather than a solution. Fine-tuning continues training on examples from the target domain so that the model learns its terminology, facts and style.

Howard and Ruder [1] showed with ULMFiT that fine-tuning a pre-trained language model achieves strong results with little labelled data, and BERT [2] made pre-train-then-fine-tune the standard approach in NLP. As models grew to billions of parameters, new methods emerged to make fine-tuning feasible on limited hardware. This paper reviews fine-tuning methods for LLMs and presents a proposed research pipeline. The work is at the research and design stage, so no experimental results are claimed.

II. BACKGROUND AND FUNDAMENTAL CONCEPTS

1. Transfer Learning

A pre-trained model's weights encode general knowledge; fine-tuning adjusts them, or a small set of added parameters, using task data. Fig. 1 classifies fine-tuning approaches.

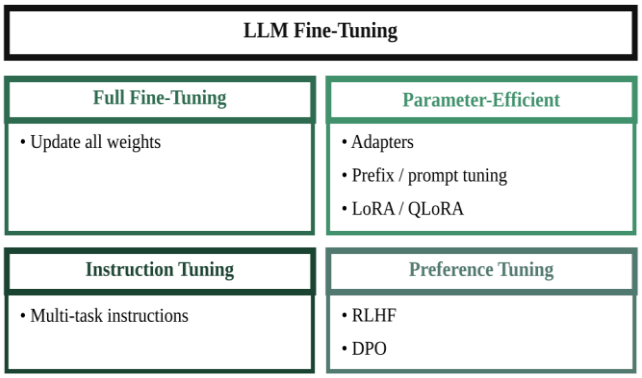


Fig. 1. Classification of LLM fine-tuning approaches.

© 2026 Musale Devendra Kailas, , This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

2. Types Of Fine-Tuning Objectives

Task fine-tuning trains on input–output pairs for one task; instruction tuning trains on many tasks described in natural language; and preference tuning adjusts outputs toward responses humans prefer.

III. LITERATURE SURVEY

Raffel *et al.* [3] introduced T5, which casts every task as text-to-text and systematically compared transfer-learning choices. Houlsby *et al.* [4] proposed adapters, small modules inserted into each layer and trained while the original weights stay fixed, achieving near full fine-tuning performance with few extra parameters.

Li and Liang [5] proposed prefix-tuning, learning continuous vectors prepended to activations, and Lester *et al.* [6] showed that prompt tuning becomes competitive with full fine-tuning as model size grows. Hu *et al.* [7] introduced LoRA, which learns low-rank updates to weight matrices, and Dettmers *et al.* [8] combined LoRA with 4-bit quantization in QLoRA, enabling fine-tuning of large models on a single GPU.

Wei *et al.* [9] showed with FLAN that instruction tuning on many tasks improves zero-shot performance on unseen tasks. Ouyang *et al.* [10] aligned models with human preferences through reinforcement learning from human feedback, and Rafailov *et al.* [11] proposed direct preference optimization as a simpler alternative. Zhou *et al.* [12] found that a small set of carefully curated examples can achieve strong alignment. Kirkpatrick *et al.* [13] described catastrophic forgetting and proposed methods to mitigate it. Table I summarizes these works.

TABLE I Summary of Literature Survey

Sr. No.	Ref.	Year	Methodology	Findings	Research Gap
1	[1]	2018	ULMFiT	Effective LM fine-tuning with little data	LSTM-based
2	[4]	2019	Adapter modules	Few trainable parameters	Inference overhead
3	[3]	2020	T5 text-to-text transfer	Systematic transfer study	Large compute
4	[5]	2021	Prefix-tuning	Lightweight generation tuning	Less stable training
5	[7]	2022	LoRA	Low-rank efficient adaptation	Rank selection
6	[9]	2022	Instruction tuning (FLAN)	Better zero-shot generalization	Needs many tasks
7	[8]	2023	QLoRA	Fine-tuning on one GPU	Quantization effects
8	[11]	2023	Direct preference optimization	Simpler alignment than RLHF	Needs preference data

IV. RESEARCH GAP

Domain datasets are often small, noisy or contain sensitive information.

Fine-tuning can cause forgetting of general abilities.

Evaluation of domain-specific generative quality is difficult and often manual.

Fine-tuned models may still hallucinate domain facts.Few studies address Indian-language domain fine-tuning with limited compute.

V. OBJECTIVES

1. To review fine-tuning approaches for language models.

2. To compare full and parameter-efficient fine-tuning.

3. To examine instruction and preference tuning.

4. To propose a fine-tuning research pipeline using open tools.

5. To identify challenges and evaluation practices.

VI. PROPOSED SYSTEM ARCHITECTURE

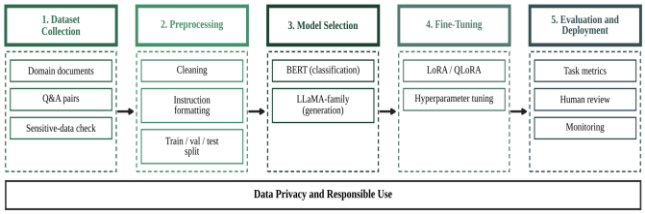


Fig. 2 shows the proposed pipeline.

Fig. 2. Proposed fine-tuning research pipeline.

1. Dataset Collection Layer

Domain documents, question–answer pairs and task examples are collected with permission and checked for sensitive data.

2. Preprocessing Layer

Text is cleaned, de-duplicated and formatted into instruction–response pairs; data are split into training, validation and test sets.

3. Model Selection Layer

A base model such as BERT for classification or a LLaMA-family model for generation is selected according to task and hardware.

4. Fine-Tuning Layer

LoRA or QLoRA adapters are trained with Hugging Face Transformers [14] and PyTorch; hyperparameters are tuned on validation data.

5. Evaluation And Deployment Layer

Models are evaluated with task metrics and human review, compared with the base model, and deployed with monitoring.

VII. METHODOLOGY AND WORKING

Fig. 3 outlines the pipeline. Fine-tuning minimizes the cross-entropy of target tokens given the input,

$$L(\theta) = -\Sigma_t \log P_\theta(y_t \mid y_{<t}, x) \quad (1)$$

and LoRA freezes the original weights W_0 and learns a low-rank update $\Delta W = BA$, so that the adapted layer computes

$$h = W_0 x + BAx \quad (2)$$

with far fewer trainable parameters.

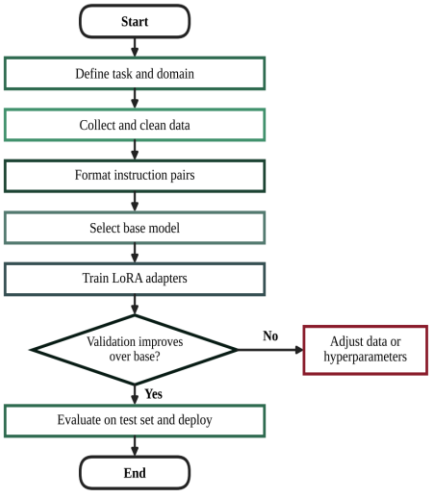


Fig. 3. Proposed fine-tuning workflow.

Planned evaluation includes accuracy or F1-score for classification, ROUGE or exact match for question answering, and human ratings of correctness and helpfulness.

VIII. COMPARATIVE ANALYSIS

Table II compares fine-tuning methods. Full fine-tuning gives maximum flexibility at high cost, adapters and LoRA give most of the benefit with a small fraction of parameters, prompt tuning is lightest, and QLoRA minimizes memory.

Method	Trainable Params	Memory	Performance
Full fine-tuning	All	Very high	Highest flexibility
Adapters	Small	Medium	Near full
Prompt / prefix tuning	Very small	Low	Good at large scale
LoRA	Small	Medium	Near full
QLoRA	Small	Low	Near full

TABLE II Comparison of Fine-Tuning Methods

IX. CHALLENGES AND FUTURE SCOPE

Table III summarizes challenges. Future work includes multilingual fine-tuning for Indian languages, combining fine-tuning with retrieval, efficient evaluation methods and continual fine-tuning without forgetting.

TABLE III Challenges and Mitigation

Challenge	Mitigation
Small or noisy data	Curation; data augmentation
Catastrophic forgetting	PEFT; mixed general data
Overfitting	Validation monitoring; early stopping
Compute cost	LoRA / QLoRA on single GPU
Hallucination	Retrieval grounding; human review

X. CONCLUSION

Fine-tuning is the bridge between general-purpose language models and practical domain applications. The reviewed literature shows how transfer learning evolved into parameter-efficient methods such as adapters and LoRA, and into instruction and preference tuning that align models with users. These methods make it feasible for students and small organizations to build accurate, domain-specific language applications on modest hardware, provided that data quality, forgetting, evaluation and safety are carefully managed.

1. Acknowledgment

The authors express their sincere gratitude to the Department of Artificial Intelligence and Data Science, Jaihind College of Engineering, Pune, for providing the facilities, guidance and encouragement needed to carry out this work. The authors also thank the faculty members and colleagues whose valuable suggestions helped improve the quality of this review.

REFERENCES

1. J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in Proc. 56th Annu. Meeting ACL, 2018, pp. 328–339.
2. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
3. C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.
4. N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in Proc. ICML, 2019, pp. 2790–2799.
5. X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in Proc. 59th Annu. Meeting ACL, 2021, pp. 4582–4597.
6. B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in Proc. EMNLP, 2021, pp. 3045–3059.
7. E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in Proc. ICLR, 2022.
8. T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," in Proc. NeurIPS, vol. 36, 2023.
9. J. Wei et al., "Finetuned language models are zero-shot learners," in Proc. ICLR, 2022.
10. L. Ouyang et al., "Training language models to follow instructions with human feedback," in Proc. NeurIPS, vol. 35, 2022, pp. 27730–27744.
11. R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in Proc. NeurIPS, vol. 36, 2023.
12. C. Zhou et al., "LIMA: Less is more for alignment," in Proc. NeurIPS, vol. 36, 2023.
13. J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," Proc. Natl. Acad. Sci. USA, vol. 114, no. 13, pp. 3521–3526, 2017.
14. T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. EMNLP System Demonstrations, 2020, pp. 38–45.