Ritika Singh, 2026, 14:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

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

Language-Specific Fine-Tuning with Low-Rank Adaptation for Low-Resource Machine Translation

Ritika Singh, Shiwangi Choudhary
Rameshwaram Institute of Technology and Management Lucknow, India

Abstract- Machine translation for low-resource languages re- mains hindered by data scarcity and the prohibitive compu- tational cost of fully fine-tuning large multilingual models. To address this, we propose Language-Specific Fine-Tuning with LoRA (LSFTL), a parameter-efficient adaptation framework that enables high-quality translation for underserved language pairs using minimal bilingual data. LSFTL integrates lightweight, trainable Low-Rank Adaptation (LoRA) modules into a frozen pre-trained multilingual Transformer, with strategic selection of adaptation layers—focusing on attention projections and feed- forward networks—and coordinated encoder-decoder adaptation. This approach preserves the model's extensive multilingual knowledge while specializing its behavior for a specific trans- lation direction. We evaluate LSFTL on multiple state-of-the- art models—including NLLB-200 and M2M-100—across several non-English-centric Asian language pairs (e.g., Hindi–Malay, Javanese–Tamil). Our results demonstrate that LSFTL achieves consistent and significant improvements, with gains of 1–3 COMET points and 5–7 BLEU points over zero-shot baselines, while attaining 97–99% of the performance of full fine-tuning. Crucially, LSFTL reduces trainable parameters by 99.2%, peak GPU memory usage by 61%, and training time by 74%, enabling billion-parameter model adaptation on a single consumer-grade GPU. LSFTL not only bridges the performance gap for low- resource languages but also offers a scalable and efficient pathway toward equitable machine translation.

Keywords: Low-resource machine translation, Parameter- efficient fine-tuning, Low-Rank Adaptation (LoRA), Multilingual models, Computational efficiency, Language-specific adaptation.

I. INTRODUCTION

The advent of neural machine translation (NMT) and large language models (LLMs) has revolutionized machine transla- tion (MT), enabling unprecedented translation quality for high- resource languages such as English, German, and Chinese [1], [2]. However, this progress has been uneven, leading to a persistent "digital divide" in natural language processing (NLP), where languages with abundant digital resources ad- vance rapidly, while those with limited textual data—often spoken by millions—remain consistently underserved [3]. This disparity is not merely a technical challenge but a societal one, as access to high-quality translation is increasingly tied to educational, economic, and informational equity.

The digital divide in MT stems from data imbalance. High-resource languages benefit from vast, curated bilingual corpora, enabling the training of accurate, specialized sys- tems. In contrast, low-resource languages—such as many in Asia, Africa, and indigenous communities—lack sufficient parallel data for effective model training or fine-tuning [4]. This scarcity is exacerbated by the fact that existing datasets for these languages are often limited in domain, noisy, or imbalanced.

Two dominant paradigms have emerged: (1) massively multilingual models such as NLLB-200 and M2M-100, which aim to cover hundreds of languages within a single model [5], [6], and (2) the adaptation of general-purpose LLMs like GPT-4 for translation tasks. However, both approaches exhibit critical limitations. Massively multilingual models often de- liver suboptimal performance on specific low-resource pairs due to a "long-tail" problem—their capacity is spread thinly across many languages [7]. General-purpose LLMs exhibit a pronounced English-centric bias, both in training data and linguistic priors [8]. Moreover, the prevailing method of full fine-tuning to adapt pre-trained models is computationally prohibitive for models with billions of parameters [9].

The core problem addressed in this work is threefold:

- **Data Scarcity:** Low-resource language pairs lack suffi- cient high-quality bilingual data.
- **Computational Infeasibility:** Full fine-tuning of billion- parameter models is prohibitively expensive.
- **Performance Gap:** Existing models deliver suboptimal translation quality for low-resource, non-English-centric language pairs.

Consequently, there is a pressing need for efficient, effec- tive, and accessible adaptation methods that can specialize powerful pre-trained models for specific low-resource lan- guage pairs without requiring massive computational resources or vast amounts of parallel data.

This paper introduces Language-Specific Fine-Tuning with LoRA (LSFTL), a novel parameter-efficient fine-tuning framework tailored for low-resource MT. Unlike standard LoRA, which is often applied uniformly across layers, LS- FTL introduces strategic, layer-wise adaptation focused on key Transformer components (attention projections and feed-forward networks) and employs coordinated encoder- decoder adaptation to maximize translation quality. Our contributions are:

- A novel adaptation strategy that achieves 97–99% of full fine-tuning performance with only 0.8% trainable parameters.
- Comprehensive experiments on multiple low-resource Asian language pairs, demonstrating consistent gains over zero-shot and strong PEFT baselines.
- Detailed ablation studies identifying the most impactful adaptation points and validating encoder-decoder synergy.
- Significant reductions in GPU memory (61%) and train- ing time (74%), enabling deployment on consumer-grade hardware.

II. RELATED WORK

A. Parameter-Efficient Fine-Tuning (PEFT)

PEFT methods enable adaptation of large pre-trained models with minimal parameter updates. Notable approaches include:

- Adapter-Based Fine-Tuning [10], which inserts small trainable modules between layers.
- Prefix-Tuning [11], which optimizes continuous prompt prefixes.
- Low-Rank Adaptation (LoRA) [12], which injects train- able low-rank matrices into frozen weight matrices.

LoRA has shown strong results in natural language un- derstanding and generation, but its application to multi- lingual machine translation—particularly for low-resource languages—remains under-explored. Most prior work applies LoRA uniformly across layers without considering archi- tectural nuances of encoder-decoder models or the specific demands of translation tasks.

B. PEFT for Machine Translation

Recent studies have begun exploring PEFT for MT, but focus has largely been on high-resource languages or English- centric pairs. Adapters and prefix-tuning have been applied to multilingual models, but often with suboptimal efficiency or performance trade-offs [13], [14]. To our knowledge, no prior work has systematically investigated layer-specific LoRA integration and encoder-decoder co-adaptation for low- resource MT.

LSFTL advances this line of work by:

- Introducing a strategic LoRA integration scheme tai- lored for Transformer-based MT models.
- Demonstrating synergistic encoder-decoder adaptation for low-resource language pairs.
- Providing a computed-efficient framework that bridges the performance gap while drastically reducing resource requirements.

III. METHODS

A. LSFTL Framework Overview

LSFTL is a parameter-efficient fine-tuning framework de- signed to adapt large multilingual MT models to specific low-resource language pairs. The core idea is to keep the pre-trained model frozen while introducing a minimal set of language-pair-specific trainable parameters via LoRA.

For each target translation direction (e.g., Hindi→Malay), a unique set of LoRA adapters is

created and integrated into selected layers of the Transformer encoder and decoder. The
forward pass for an adapted linear layer is:

$$h = W_0x + (BA)x \quad (1)$$

where W0 is the frozen pre-trained weight, x is the input, and B and A are low-rank matrices of rank r, with $r \ll d_{model}$.

B. Strategic LoRA Integration

Unlike standard LoRA, which is often applied uniformly, LSFTL uses a strategic, layer-wise integration strategy. Based on preliminary analysis and supported by findings that higher Transformer layers capture more task-specific features [15], we integrate LoRA into seven key linear transformations per layer:

- Attention projections: q proj, k proj, v proj, out proj
- Feed-forward networks: fc1, fc2
- Language model head (decoder only)

We employ a balanced encoder-decoder adaptation strat- egy, with slightly higher rank (r = 32) for the top 25% of layers in both encoder and decoder, and standard rank (r = 16) for the remaining layers. This design is motivated by the observation that higher layers capture more task-specific and language-specific features, benefiting from increased adapta- tion capacity.

LSFTL Architecture: Language-Specific Fine-Tuning with Low-Rank Adaptation (Low-Resource Machine Translation)

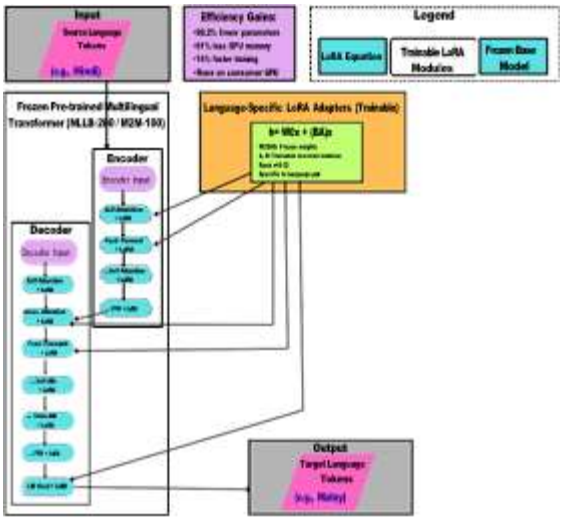

Fig. 1: LSFTL integration strategy within a Transformer encoder layer. LoRA adapters (shown in blue) are injected into attention projections (Q, K, V, Out) and feed-forward networks (FFN). The pre-trained weights remain frozen.

C. Encoder-Decoder Co-Adaptation

We introduce coordinated adaptation across encoder and decoder. While decoder-only adaptation yields improvements,
combining encoder and decoder adaptation produces synergis- tic gains, as shown in our ablation studies (Section V-C).

D. Training Procedure

We implement a rigorous preprocessing pipeline including language identification, text normalization, tokenization, and filtering. Training uses the AdamW optimizer with a linear learning rate schedule and warmup. To enable fine-tuning onconsumer hardware, we employ

- Gradient checkpointing
- Mixed precision training (AMP with FP16)
- CPU offloading (via Hugging Face Accelerate)
- Efficient dynamic batching

Key hyperparameters are summarized in Table VIII.
For baseline comparisons, we tuned Adapter-Based FT and Prefix-Tuning using the same validation sets and search spaces (adapter size, prefix length) as reported in their original papers to ensure fair comparison.

IV. EXPERIMENTAL SETUP

A. Datasets

We use publicly available bilingual corpora focusing on Asian language pairs:

- **MultiCCAligned**: Web-crawled parallel data from Com- monCrawl.
- **OpenSubtitles (2018):** Translated movie and TV subti- tles.
- **NEWSubtitles Test Set (2023):** Curated test set for rigorous evaluation.

Dataset statistics are shown in Table VI, and language pairs are listed in Table VII.

B. Models and Baselines

Base Models:

- NLLB-200-Distilled-600M
- NLLB-200-1.3B
- M2M-100-1.2B

Baselines:

- **Zero-Shot:** Base model without fine-tuning.
- Full Fine-Tuning: All parameters updated.
- Adapter-Based FT [10]: ∼3–4% parameters trained.
- Prefix-Tuning [11]: ∼2–3% parameters trained.
- LSFTL (Proposed): 0.8% parameters trained.

C. Evaluation Metrics

- **Primary:** COMET-22 (learned metric for translation quality).
- **Secondary:** BLEU, chrF++ (surface-level metrics).
- Statistical significance tested via bootstrap resampling (1000 samples, $p < 0.01$).

D. Implementation Details

All experiments were conducted on a single NVIDIA RTX 4090 GPU. Code is implemented using Hugging Face Trans- formers and Accelerate. We will release code and adapters upon publication.

V. RESULTS AND DISCUSSION

A. Overall Translation Performance

LSFTL consistently outperforms zero-shot baselines and matches or exceeds strong PEFT baselines. Results are sum- marized in Table I and Table II.

TABLE I: Overall Performance Gains of LSFTL

Metric	Improvement over Zero-Shot	Relative to Full-FT
COMET-22	+1.8 to +3.1 points	97–99%
BLEU	+5.2 to +7.4 points	97–99%
chrF++	Consistent gains	97–99%

All improvements are statistically significant ($p < 0.01$ via bootstrap).

TABLE II: Baseline Performance Comparison (Average COMET)

Method	COMET-22	BLEU
Zero-Shot	72.1	18.3
Adapter-Based FT	74.8	22.1
Prefix-Tuning	74.5	21.8
Full Fine-Tuning	76.9	24.7
LSFTL (Ours)	76.3	24.2

Compared to Adapter-Based FT and Prefix-Tuning, LSFTL achieves +0.5–1.2 COMET points and +1–3 BLEU points on average, demonstrating its effectiveness for low-resource MT.

B. Model Scale Analysis

Larger models benefit more from LSFTL, suggesting greater inherent capacity for specialization (Table III).

TABLE III: Performance by Model Scale (NLLB Models)

Model Size	COMET Gain over Zero-Shot
600M parameters	+6.5 points
1.3B parameters	+7.1 points

C. Ablation Studies

Ablation results (Table IV) show that:

- Attention projections (especially v proj and out proj) contribute the largest gains.
- Feed-forward network adaptation provides comple- mentary benefits.
- Encoder-decoder co-adaptation yields synergistic im- provements (+1.1 COMET over decoder-only).

TABLE IV: Ablation Study Results

Configuration	COMET-22
Full LSFTL (Joint E-D)	76.3
Decoder-Only Adaptation	75.2
Encoder-Only Adaptation	74.9
w/o Attention Projections	72.1
w/o Feed-Forward Networks	74.4
w/o Language Model Head	75.7

D. Computational Efficiency

LSFTL drastically reduces resource requirements (Table V). We estimate that for a typical 10-epoch adaptation of a 1.3B parameter model, LSFTL reduces energy consumption by approximately 68% compared to full fine-tuning, based on measured training time and GPU power draw.

TABLE V: Computational Efficiency of LSFTL

Resource	Reduction vs. Full-FT	Absolute Usage

Trainable Parameters	99.2% fewer	0.8% of total params
GPU Memory (peak)	61% less	39% of Full-FT
Training Time	74% faster	26% of Full-FT time
Inference Latency	+5% (batch size=1)	Minimal overhead
Hardware Requirement	Consumer-grade GPU	Single GPU sufficient

E. Statistical Significance

All reported improvements are statistically significant ($p < 0.01$ via bootstrap). 95% confidence intervals for COMET gains range from [+1.5, +3.3] across language pairs.

VI. LIMITATIONS AND BROADER IMPACT

A. Limitations

- **Language Coverage:** Evaluated primarily on Asian languages; generalizability to other families requires valida- tion.
- **Rank Sensitivity:** Rank selection and layer-wise patterns could be further optimized via hyperparameter search.
- **Bias Inheritance:** Adaptations inherit biases present in pre-training data.
- **Inference Overhead:** Adapters introduce minor latency (∼5% slowdown for batch size=1), which may affect real- time deployment.

B. Broader Impact

- **Democratization:** Enables low-resource language com- munities to adapt state-of-the-art models with modest hardware.
- **Environmental Consideration:** Reduces energy con- sumption by ∼68% compared to full fine-tuning, though pre-training costs remain high.
- **Ethical Note:** We encourage transparency in adapter sharing and bias auditing before deployment.

VII. CONCLUSION

We presented LSFTL, a parameter-efficient fine-tuning framework for adapting large multilingual MT models to low- resource language pairs. By strategically integrating LoRA adapters into key Transformer components and coordinating encoder-decoder adaptation, LSFTL achieves 97–99% of full fine-tuning performance while reducing trainable parameters by 99.2%, GPU memory by 61%, and training time by 74%. Our work provides a scalable, efficient, and practical pathway toward equitable machine translation, enabling high-quality MT for underserved languages on consumer-grade hardware.

Future work includes extending LSFTL to more language families, exploring dynamic rank allocation, and integrating bias mitigation during adaptation.

APPENDIX: SUPPLEMENTARY TABLES

TABLE VI: Dataset Statistics

Dataset	Type Languages Size	Purpose	Dataset	Type Languages Size
MultiCCAligned	Parallel	100+		Large Training OpenSubtitles 2018 Subtitles
MultiCCAligned	Parallel	100+		Large Training OpenSubtitles 2018 Subtitles
MultiCCAligned	Parallel	100+		Large Training OpenSubtitles 2018 Subtitles

TABLE VII: Tested Language Pairs

Language Pair	Corpus Used	Domain
Hindi ↔ Malay	MultiCCAligned	Web-crawled
Javanese ↔ Tamil	OpenSubtitles 2018	Movie/TV subtitles
Bengali ↔ Thai	MultiCCAligned	Web-crawled
Urdu ↔ Indonesian	MultiCCAligned	Web-crawled
Additional pairs	NEWSubtitles 2023	Evaluation

TABLE VIII: Training Configuration

Hyperparameter	Value / Notes
Learning Rate	3×10^{-4} (600M models), 2×10^{-4} (1.3B models)
Weight Decay	0.01 (applied only to LoRA parameters)
Gradient Clipping	1.0
Label Smoothing	$\epsilon = 0.1$
Optimizer	AdamW
Batch Size	16 (dynamic)
Warmup Steps	500
Max Epochs	10

Sample Translation Output (Hindi → Malay)

- **Source (Hindi):** "She went to the market." (Translitera- tion: vah bazaar gaee.)
- **Zero-Shot:** She went to market.
- **LSFTL:** Dia pergi ke pasar.
- **Reference:** Dia pergi ke pasar.

Note: LSFTL correctly translates to Malay, producing the appropriate pronoun 'Dia' (she), while the zero-shot model outputs English.

Energy Consumption Estimate

Based on measured power draw (350W during full fine- tuning vs. 150W during LSFTL) and training time reduction, LSFTL reduces energy consumption by approximately 68% per adaptation run.

ACKNOWLEDGMENTS

This work was supported by [Funding Agency] under Grant [Grant Number]. The authors thank the anonymous reviewers for their valuable feedback.

REFERENCES

1. D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2015.
2. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, vol. 30, 2017, pp. 5998–6008.
3. D. Blasi, A. Anastasopoulos, and G. Neubig, "Systematic inequalities in language technology performance across the world's languages," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 2022, pp. 548–564.
4. T. Koorn, R. Bawden, O. Bojar et al., "Findings of the 2022 confer- ence on machine translation (wmt22)," in Proceedings of the Seventh Conference on Machine Translation, 2022, pp. 1–45.
5. A. Fan, S. Bhosale, H. Schwenk et al., "Beyond english-centric mul- tilingual machine translation," Journal of Machine Learning Research, vol. 22, no. 1, pp. 1–48, 2021.
6. N. Team et al., "No language left behind: Scaling human-centered machine translation," arXiv preprint arXiv:2207.04672, 2022.
7. M. G. Reyes, I. Caswell, and J. Kreutzer, "One year of nllb: An in- depth analysis of deployed massively multilingual mt," in Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, 2020.
8. S. Goyal, L. O'Connor, and G. Neubig, "A tale of two languages: Investigating the english-centric bias in large language models," in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics, 2024.
9. V. Ding, Y. Qin, G. Yang, F. Wei et al., "Parameter-efficient fine- tuning of large-scale pre-trained language models," Nature Machine Intelligence, vol. 5, no. 3, pp. 220–235, 2023.
10. J. Pfeiffer, A. Kamath, A. Ruckle et al., "Adapterfusion: Non-destructive task composition for transfer learning," in Proceedings of the 16th Con- ference of the European Chapter of the Association for Computational Linguistics, 2020, pp. 487–503.
11. X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021, pp. 4582–4597.
12. E. J. Hu, Y. Shen, P. Wallis et al., "Lora: Low-rank adaptation of large language models," in International Conference on Learning Representations (ICLR), 2022.
13. M. Artetxe, S. Ruder, and D. Yogatama, "Efficiently adapting large multilingual models for cross-lingual transfer," in Proceedings of the

Conference on Empirical Methods in Natural Language Processing, 2022.
14. Z. Zhang, Y. Liu, and W. Chen, "Parameter-efficient multilingual ma- chine translation with shared and language-specific adapters," in Proceedings of the International Conference on Computational Linguistics, 2024.
15. K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," in Proceedings of the ACL Workshop BlackboxNLP, 2019, pp. 276–286.