

Evaluating Parameter-Efficient Fine-Tuning for Cross-Lingual Natural Language Inference: A Case Study on Nepali XNLI

Amir Shapkota

Department of IT Engineering
Nepal College of Information Technology
Kathmandu, Nepal
amir.201402@ncit.edu.np

Sachee Ghimire

Department of IT Engineering
Nepal College of Information Technology
Kathmandu, Nepal
sachee.201431@ncit.edu.np

Abstract—Nepali language remains under-served in multi-lingual Natural Language Processing (NLP), and robust evaluation practices are often missing for low-resource settings. We present a reproducible study of Cross-lingual Natural Language Inference (XNLI) for Nepali using XLM-RoBERTa base, comparing full fine-tuning against parameter-efficient Low-Rank Adaptation (LoRA). Models are fine-tuned on English Natural Language Inference (NLI) data and evaluated on an English-to-Nepali translated subset of XNLI. Our pipeline performs three independent runs (seeds 42/43/44), logs predictions and confusion matrices, aggregates mean $\pm$ std across seeds and reports 95% bootstrap confidence intervals computed directly from per-example predictions. Full fine-tuning achieves 0.7687 accuracy and 0.7689 macro-F1, outperforming a fixed-configuration LoRA ($r=8$, $\alpha=16$, no hyperparameter sweep) by approximately 2.1 points. We discuss why LoRA under-performs in this setting, the likely role of translation artifacts, and practical directions for closing the gap, while emphasizing reproducibility and error analysis over strong claims about absolute accuracy.

Keywords—Parameter-Efficient Fine-Tuning (PEFT), Natural Language Inference, Cross-Lingual Transfer, XLM-RoBERTa, Low-Resource NLP

I. INTRODUCTION

Natural Language Inference (NLI) determining whether a hypothesis is entailed by, contradicts, or is neutral with respect to a premise is a core probe of sentence-level semantics and transfer learning [1]. Multilingual encoders such as mBERT and XLM-RoBERTa (XLM-R), pretrained on massive multilingual corpora, have enabled cross-lingual transfer without explicit alignment, and the XNLI benchmark extended English NLI data into fifteen languages to support zero-shot and translate-test evaluation across language families [2, 5, 6]. Fully fine-tuning such models are computationally expensive, which motivates parameter-efficient fine-tuning (PEFT) methods such as Low-Rank Adaptation (LoRA) that update only a small fraction of parameters while aiming to preserve full fine-tuning accuracy [3, 4].

Nepali, written in Devanagari script and spoken widely across South Asia, is not part of the original XNLI language suite and remains constrained by scarce annotated corpora, domain mismatch with English-centric pretraining data, and a lack of standardized benchmarks. These constraints make it an informative test case for whether PEFT can match full fine-tuning under cross-lingual transfer, and for whether aggregate

accuracy alone can be trusted without multi-seed and per-class evaluation.

This paper takes a reproducibility-first view of Nepali NLI under cross-lingual transfer. Using XLM-R base trained on English NLI data and evaluated on a 500-example English-to-Nepali translated subset of XNLI, we compare full fine-tuning against LoRA under identical training conditions. Our pipeline runs three independent seeds (42, 43, 44), logs per-example predictions and confusion matrices, aggregates mean $\pm$ standard deviation across seeds and reports 95% bootstrap confidence intervals computed directly from predictions. Our contribution is not a new state-of-the-art result, but a transparent, compute-conscious comparison with multi-seed reporting, per-class F1, and bootstrap uncertainty that is, to our knowledge, rarely documented so explicitly for Nepali cross-lingual NLI.

II. RELATED WORKS

A. Cross-Lingual NLI and Multilingual Encoders

The XNLI benchmark established the standard protocol for multilingual NLI evaluation by translating English MultiNLI data into fifteen languages, enabling zero-shot and translate-test transfer [2]. Pires et al. showed that mBERT's multilingual representations transfer across languages without explicit cross-lingual supervision [5], and Conneau et al. demonstrated that scaling multilingual pretraining with XLM-R substantially improves zero-shot XNLI transfer, particularly for low-resource and non-Latin-script languages [6]. These encoders now underpin most cross-lingual NLI pipelines built with modern toolkits such as Transformers and Datasets [7, 8].

B. Parameter-Efficient Fine-Tuning

Because full fine-tuning of large multilingual encoders is computationally expensive [3], adapter-based methods [9] and LoRA [4] insert lightweight or low-rank modules while freezing the backbone, achieving near full fine-tuning accuracy at a fraction of the trainable-parameter cost. QLoRA extends this idea to quantized backbones, enabling adaptation of much larger multilingual models on limited hardware [11]. Dhakal and Baral applied 4-bit-quantized LoRA fine-tuning of mBART for Nepali abstractive summarization and found it competitive with heavier fine-tuning strategies for headline generation [14], suggesting PEFT is a practical default for Nepali NLP tasks beyond classification.

C. Low-Resource and Native-Language NLI Evaluation

Because translation can introduce artifacts that inflate or deflate apparent task difficulty, several efforts build native or

human-verified evaluation sets rather than relying solely on machine-translated benchmarks; IndoNLI, AmericasNLI [13], and similar community efforts prioritize culturally grounded annotation. Htet and Dras applied a comparable translation-and-expert-verification methodology to build Myanmar XNLI, reporting that translate-test transfer for Myanmar, Swahili, and Urdu could exceed zero-shot cross-lingual transfer under XLM-R and mDeBERTa, which underscores that transfer quality is sensitive to both script and translation pipeline [15]. For South Asian languages specifically, Faria et al. compared fine-tuned transformer baselines — mBERT, BanglaBERT, and sahaBERT — against large language models on Bangla XNLI and found that fine-tuned encoders remained competitive with, and often superior to, zero- and few-shot LLMs [16]. Closer to our setting, Nyachhyon et al. introduced the Nepali Language Understanding Evaluation (NLUE) benchmark, extending Nep-gLUE with textual entailment and other NLU tasks, and reported that existing multilingual models still struggle on these added, more complex tasks, with the best multilingual model outperforming the best monolingual Nepali models overall [17].

Prior Nepali-focused work has concentrated on benchmark construction (NLUE) or task-specific fine-tuning (summarization), while cross-lingual NLI studies for other low-resource, non-Latin-script languages (Myanmar, Bangla) have compared transfer strategies but not adaptation efficiency. To our knowledge, no prior study reports a multi-seed, bootstrap-validated comparison of full fine-tuning against LoRA specifically for Nepali XNLI; our study addresses this gap.

III. METHODOLOGY

A. Task and Data

We studied cross-lingual NLI: given a Nepali premise and hypothesis, predict entailment, neutral, or contradiction (mapped to labels 0, 1, and 2 respectively). We adopted the translate-test paradigm of XNLI, training on English NLI data and evaluating on Nepali translations of the evaluation split, so that model capacity and transfer are assessed without target-language supervision [2]. Training used English NLI data following the XNLI protocol; evaluation used a 500-example English-to-Nepali translated subset of XNLI. The Nepali evaluation set was constructed by translating the first 500 examples (a deterministic prefix, not a random sample) of the English XNLI validation split [2] using NLLB-200-distilled-600M (facebook/nllb-200-distilled-600M, eng_Latn→npi_Deva) via the Hugging Face Transformers translation pipeline. No manual post-editing or human validation was performed on the translated premises or hypotheses; the translated text and original XNLI labels were written directly to a JSONL file. We therefore treat this set as a proxy evaluation of cross-lingual transfer rather than a gold-standard, human-verified Nepali benchmark, and report results with this caveat in mind. All data loading and preprocessing used standard tooling to ensure reproducibility and consistency across runs [7, 8]. The resulting 500-item evaluation set is approximately class-balanced (167 entailment, 167 neutral, and 166 contradiction; 33.4%, 33.4%, and 33.2%, respectively), although the selection was deterministic rather than randomized.

B. Models and Adaptation Regimes

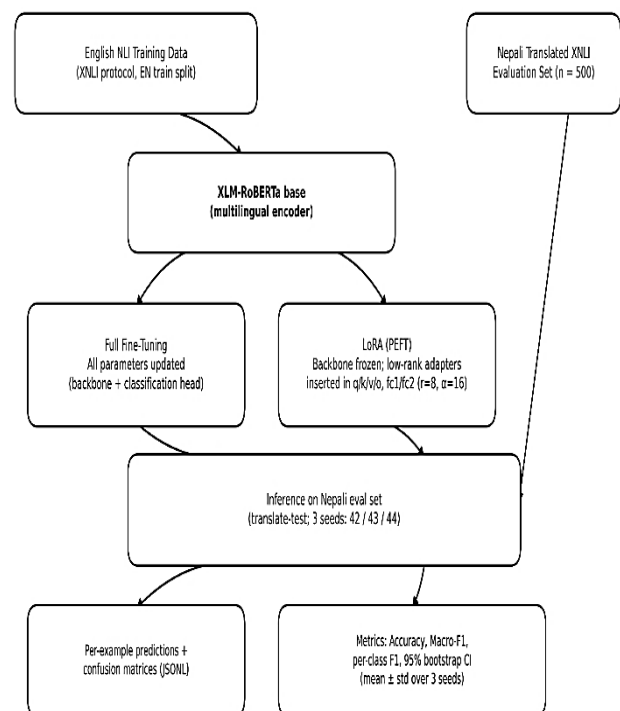
We used XLM-RoBERTa base as the multilingual encoder [6] and compared two adaptation regimes. In full fine-tuning, all model parameters were updated during supervised training.

In LoRA, the base model was frozen and low-rank adapters were inserted into the attention projections and feed-forward layers (q/k/v/o and fc1/fc2) using rank $r = 8$, scaling $\alpha = 16$, dropout 0.1, and bias disabled (bias="none"), adapting only a small fraction of parameters while minimizing memory and compute [4, 9]. Fig. 1 summarizes the overall pipeline: a shared XLM-R backbone is adapted under each regime on English NLI data, and both resulting models are evaluated identically on the Nepali translated test set.

C. Training and Evaluation Protocol

Full fine-tuning and LoRA share the same data splits, epochs, learning-rate schedule, and evaluation cadence, so that differences in outcome can be attributed to the adaptation method rather than to hyperparameter choices; we deliberately avoided per-regime tuning to keep the comparison unbiased. Each regime was trained independently under three seeds (42, 43, and 44) to quantify run-to-run variability, with mixed-precision (fp16) and TF32 enabled where available to accelerate training and reduce memory use, and periodic checkpointing that automatically resumes from the latest valid, uncorrupted state if training is interrupted.

Fig. 1. Pipeline overview: a shared XLM-R backbone is adapted via full fine-tuning or LoRA on English NLI data, then evaluated on the Nepali translated XNLI set across three seeds, producing per-example predictions and aggregated metrics.



For evaluation we reported accuracy as the primary metric alongside macro-F1, per-class F1, and confusion matrices for error analysis, aggregating each regime's three seeds as mean $\pm$ standard deviation and estimating 95% confidence intervals for accuracy via nonparametric bootstrap resampling directly on per-example predictions (1,000 resamples) [10]. For every run we stored per-example predictions (JSONL), confusion matrices, and a run configuration file capturing all the training and evaluation arguments, together with deterministic seeding of data shuffling and initialization, to support exact regeneration and post-hoc analysis. This protocol emphasized compute-conscious, apples-to-apples comparison between full-parameter updates and LoRA while preserving transparency and statistical rigor for Nepali NLI.

Both regimes were trained with an identical optimization configuration to isolate the effect of the adaptation method itself (Table I): AdamW optimizer, learning rate $2e-5$, linear schedule with 6% warmup, weight decay 0.01, per-device batch size 16 (gradient accumulation of 1), maximum sequence length 256 tokens, for 3 epochs, with mixed-precision (FP16) and TF32 enabled identically for both. Evaluation and checkpointing occurred every 300 steps, with logging every 50 steps. LoRA-specific hyperparameters are reported separately in Table II.

TABLE I. TRAINING HYPERPARAMETERS (SHARED)

Parameter	Value
Learning rate	$2e-5$
Optimizer	AdamW (adamw_torch)
LR scheduler	Linear, 6% warmup
Weight decay	0.01
Batch size (train/eval)	16 per device
Gradient accumulation	1
Epochs	3
Max sequence length	256 tokens
Mixed precision	FP16 + TF32
Eval/checkpoint frequency	Every 300 steps
Logging frequency	Every 50 steps

TABLE II. LORA-SPECIFIC HYPERPARAMETERS

Parameter	Value
Rank (r)	8
Alpha (α)	16
Dropout	0.1
Target modules	q_proj, k_proj, v_proj, o_proj, fc1, fc2
Bias	none
Task type	Sequence classification

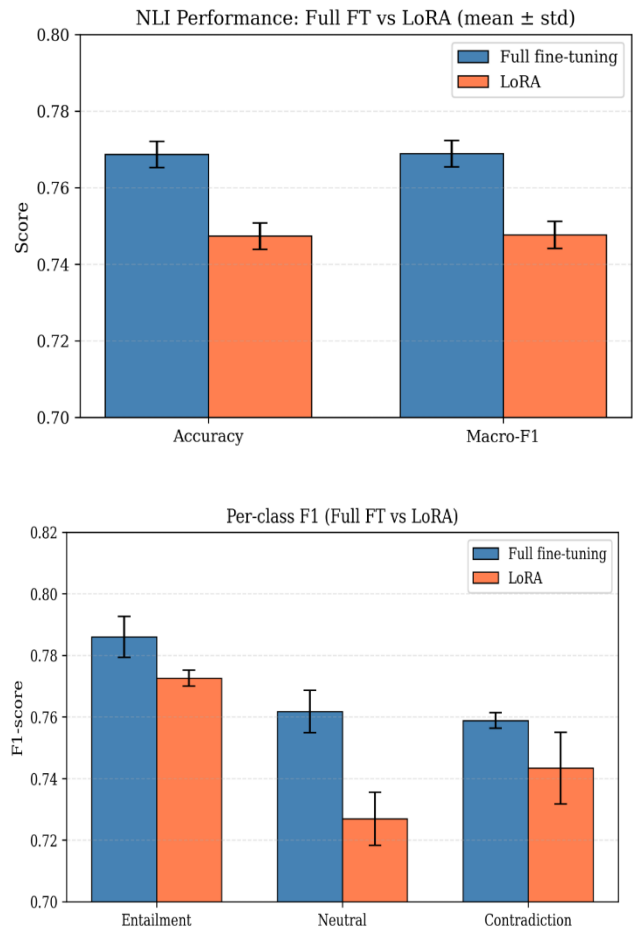
IV. EXPERIMENTAL RESULTS

TABLE III. XNLI EN→NE (TRANSLATED TEST); XLM-ROBERTA BASE FINE-TUNED FOR 3 EPOCHS; AGGREGATED OVER SEEDS 42/43/44

Method	Accuracy (mean $\pm$ std)	Macro-F1 (mean $\pm$ std)
Full fine-tuning	0.7687 ± 0.0042	0.7689 ± 0.0042
LoRA (PEFT)	0.7473 ± 0.0042	0.7476 ± 0.0043

Full fine-tuning outperforms LoRA in all three seeds on the Nepali XNLI evaluation set (500 items), with low variance across seeds and an absolute gap of +0.0214 accuracy and +0.0213 macro-F1 in favor of full fine-tuning (Fig. 2, left). Per-class F1 (Fig. 2, right) shows the same ordering for every class, with the largest gap on the neutral class (0.762 vs. 0.727) and the smallest on entailment (0.786 vs. 0.773), suggesting that LoRA's constrained update space most affects the classes with the subtlest lexical cues.

Table IV reports per-seed 95% bootstrap confidence intervals for accuracy, computed directly from per-example predictions. Given the modest test size ($n = 500$), per-seed intervals partly overlap, but full fine-tuning's accuracy exceeds LoRA's for all three seeds. To assess whether the observed accuracy gap is statistically distinguishable from chance, we conducted a paired analysis directly on per-example predictions.

Fig. 2. Above: aggregate accuracy and macro-F1 (mean $\pm$ std). Below: per-class F1 averaged across seeds.

At the individual-seed level, McNemar's test (continuity-corrected) does not reach significance for any single seed (seed 42: $p=0.126$; seed 43: $p=0.566$; seed 44: $p=0.182$), and per-seed paired bootstrap 95% confidence intervals on the accuracy difference all include zero. This indicates that, taken individually, no single seed provides conclusive evidence of a difference between methods. However, pooling paired predictions across all three seeds ($n=1,500$) yields a paired bootstrap 95% CI for the difference of $[0.002, 0.040]$, excluding zero, and a McNemar's test significant at $p=0.035$. We interpret this as suggestive but not overwhelming evidence: the gap is directionally consistent (full fine-tuning outperforms LoRA in all three seeds) and reaches significance when evidence is pooled, but individual-seed comparisons alone would not support a strong claim of a reliable difference at this sample size ($n=500$ per seed).

TABLE IV. PER-SEED 95% BOOTSTRAP CONFIDENCE INTERVALS FOR ACCURACY (1,000 RESAMPLES)

Method	Seed 42	Seed 43	Seed 44
Full fine-tuning	[0.732, 0.808]	[0.724, 0.802]	[0.730, 0.806]
LoRA (PEFT)	[0.704, 0.782]	[0.716, 0.790]	[0.710, 0.782]

Confusion matrices averaged across seeds (Fig. 3) show similar class balance for both regimes, with the largest disagreements between the neutral and contradiction classes; full fine-tuning shows a lower confusion rate between these two classes than LoRA (12.7% vs. 14.5% for neutral $\rightarrow$ contradiction, and 15.2% vs. 17.6% for contradiction $\rightarrow$ neutral), consistent with the per-class F1 pattern above.

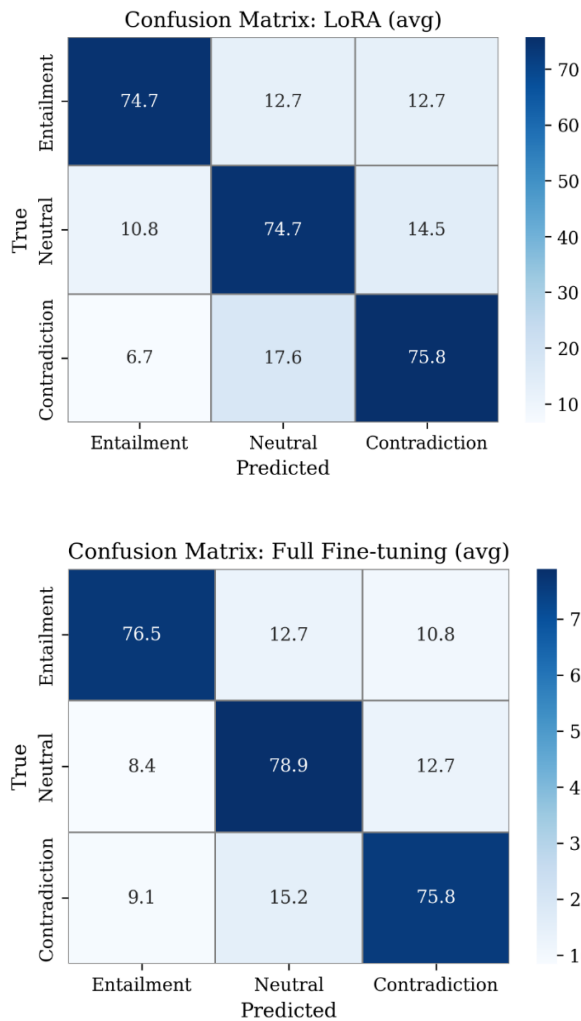


Fig. 3. Confusion matrices averaged across seeds for LoRA (above) and full fine-tuning (below)

Fig. 4 reports training and validation loss alongside validation accuracy per epoch for both regimes, illustrating convergence behavior over the three training epochs. Full fine-tuning's training loss decreases steadily throughout training, reaching roughly 0.25–0.30 by the end of epoch 3, whereas LoRA's training loss plateaus earlier, flattening around 0.50–0.55 from roughly epoch 1 onward. Validation accuracy shows a matching pattern: full fine-tuning rises faster in the first half-epoch and continues to creep upward across all three epochs, ending near 0.76, while LoRA's validation accuracy saturates by around epoch 0.5–0.75 and stays close to 0.73–0.74 for the remainder of training. This is consistent with LoRA's restricted update space reaching its effective capacity early, while full fine-tuning continues to extract gains from additional training. Validation loss tells a complementary story: LoRA's validation loss decreases smoothly and monotonically to a stable plateau near 0.69, whereas full fine-tuning's validation loss is noisier and ticks upward again in the final quarter-epoch even as its training loss keeps falling — a mild divergence between training and validation loss that is a classic early signature of overfitting, though validation accuracy does not yet degrade within the three-epoch budget used here.

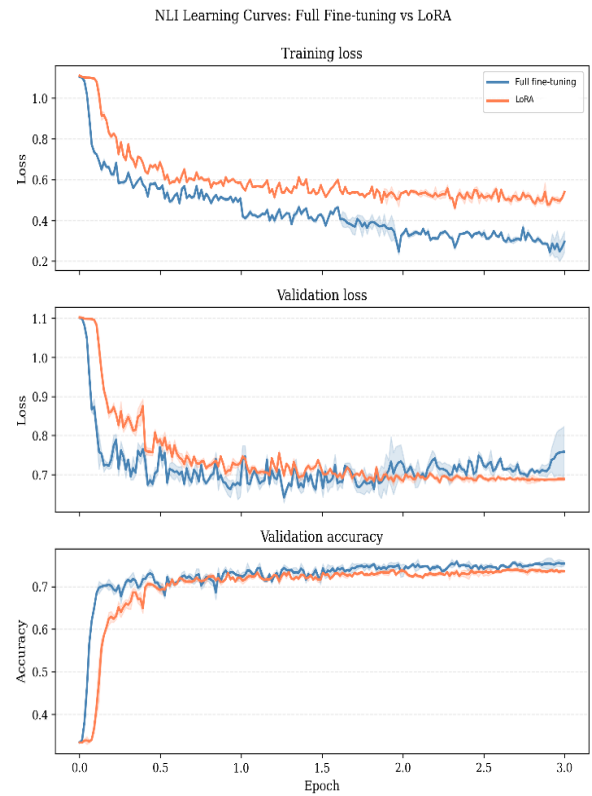


Fig. 4. Training loss, validation loss, and validation accuracy per epoch for full fine-tuning vs. LoRA (representative run; shaded bands show variability across logging steps)

A. Parameter Efficiency

Table V reports trainable parameter counts for each regime, computed identically across all three seeds (architecture is seed-invariant). Full fine-tuning updates all 278,045,955 parameters of XLM-RoBERTa base. LoRA updates only 1,920,003 parameters (0.686% of the base model), comprising 1,327,104 parameters in the inserted low-rank adapter matrices (q_proj/k_proj/v_proj/o_proj, fc1/fc2; r=8) plus 592,899 parameters in the classification head, which we configure as trainable in both regimes via modules_to_save to ensure a fair comparison at the task-specific output layer. This represents a roughly 145× reduction in trainable parameters for a 2.1-point accuracy cost, which may be an acceptable trade-off in memory- or compute-constrained settings.

TABLE V. TRAINABLE PARAMETERS BY ADAPTATION REGIME

Setup	Total params	Trainable params	Trainable %
Full fine-tuning	278,045,955	278,045,955	100.000%
LoRA	279,965,958	1,920,003	0.686%

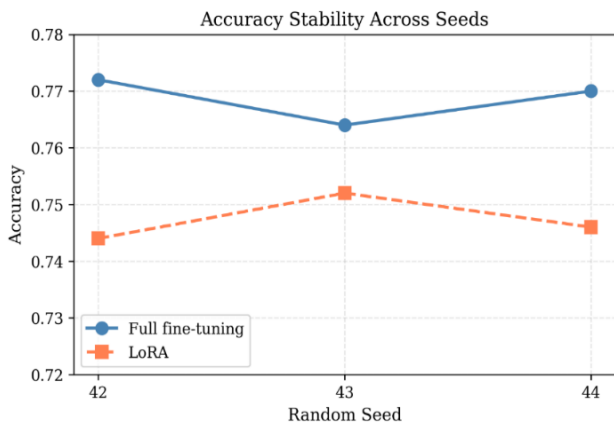


Fig. 5. Run-to-run stability across seeds for each regime

V. DISCUSSION

LoRA's constrained update space likely explains most of the gap. With $r = 8$ and $\alpha = 16$, LoRA updates only a low-rank subspace of the attention and feed-forward projections while the backbone stays frozen; for cross-lingual NLI, where the decision boundary depends on nuanced lexical and semantic shifts between Devanagari and English-centric pretraining, this restricted space can underfit those shifts, whereas full fine-tuning can realign intermediate representations more completely, yielding the observed ≈ 2.1 -point advantage. Coverage of target modules may also matter, we adapted q/k/v/o and fc1/fc2 only, and prior work suggests that expanding targets or excluding sensitive modules such as layer norms can recover additional performance [4, 11]. Because we intentionally kept hyperparameters identical across regimes to isolate the effect of the adaptation method itself, it is also possible that LoRA's true optimum requires a different learning-rate schedule, warmup, or dropout than full fine-tuning [4]; this parity, while necessary for a fair comparison, may subtly favor full fine-tuning.

Translation artifacts in the Nepali evaluation set are a second plausible contributor. Machine translation can flatten pragmatic cues or flip subtle entailment and contradiction signals, such as negation scope or implicature, and the dominant confusion mode we observe between neutral and contradiction is consistent with this kind of noise. Devanagari sub word segmentation may further interact with translation artifacts for named entities, numerals, and idioms [12]. With only 500 evaluation items, even light translation noise can shift class priors and error patterns, and the partly overlapping per-seed confidence intervals suggest that some of the full-fine-tuning/ LoRA gap could be amplified or dampened by these idiosyncrasies rather than reflecting a purely architectural difference. We note that our translated evaluation set underwent no human review, unlike verification-based efforts such as Myanmar XNLI [15] or IndoNLI. This makes MT-introduced artifacts — particularly around negation scope, implicature, and Devanagari sub-word segmentation of named entities or numerals — a plausible and unverified contributor to the neutral/contradiction confusion pattern we observe, rather than a confirmed cause.

We expect targeted LoRA tuning to close much of this gap. A small sweep over rank $r \in \{4, 8, 16, 32\}$, scaling $\alpha \in \{8, 16, 32\}$, and dropout $\in \{0.0, 0.05, 0.1\}$, combined with expanding target modules from attention-only to attention-plus-feed-forward and adjusting the learning rate or warmup specifically for LoRA, would likely recover a meaningful share of the

observed difference while preserving most of LoRA's efficiency advantage.

The learning curves in Fig. 4 add a training-dynamics perspective to this picture. LoRA's early plateau in both training loss and validation accuracy is consistent with a model that has exhausted the capacity of its low-rank update space well before the three-epoch budget ends, suggesting that additional epochs alone would likely yield little further gain without also increasing rank or target-module coverage. Full fine-tuning, in contrast, keeps improving on the training set for the full three epochs, and its validation loss beginning to tick upward in the final quarter-epoch, even as validation accuracy holds steady, is an early overfitting signal rather than a sign of continued genuine improvement; it suggests full fine-tuning's advantage in this study may partly reflect a training budget that is well matched to its capacity but slightly under-explored for LoRA, and that full fine-tuning itself may have limited additional headroom beyond epoch 3 without regularization or early stopping.

These results carry two broader implications for low-resource NLP. First, even a small, translated 500-item evaluation set is sufficient to support multi-seed runs, per-class analysis, and bootstrap confidence intervals on modest hardware, raising the bar for reproducibility in low-resource settings without requiring large GPU clusters; tooling such as Transformers and Datasets has made this kind of standardization practical [7, 8]. Second, a roughly two-point accuracy trade-off for LoRA may be entirely acceptable when memory and time are constrained, since the savings can be reinvested in more seeds or hyperparameter sweeps, whereas full fine-tuning's small edge may still justify its extra cost for mission-critical or benchmark-maximizing use cases; more broadly, translation quality and domain match likely dominate marginal model-capacity improvements at this scale, so investing in higher-fidelity evaluation data human post-editing, targeted challenge sets, dialectal coverage may yield a clearer signal than scaling the model further.

Because Nepali is not part of the original XNLI language suite and our evaluation relies on a 500-item translated set, absolute values here are not directly comparable to the multilingual averages reported for XNLI's original fifteen languages [2, 6]. We therefore emphasize within-study deltas between full fine-tuning and LoRA, together with our uncertainty estimates, and position these results as a compute-conscious baseline for future Nepali cross-lingual NLI work.

VI. CONCLUSION AND FUTURE WORK

We presented a compute-conscious, reproducible evaluation of cross-lingual NLI for Nepali using XLM-R base, comparing full fine-tuning to LoRA. Trained on English NLI and evaluated on a 500-example translated Nepali set, full fine-tuning outperformed LoRA by about 2.1 points on both accuracy and macro-F1 in all three seeds; this gap was not individually significant per seed but reached significance ($p=0.035$) when evidence was pooled across seeds, indicating a modest, directionally consistent advantage for updating all parameters. Together with per-class F1 and confusion matrices, these results suggest full fine-tuning better adapted representations to translation-induced distribution shift, while LoRA remains an attractive, efficient alternative at a small performance cost. All scripts, metrics, predictions, and figures are released to enable exact regeneration and extension. We emphasize that this is a comparison against one untuned, reasonable-default LoRA configuration rather than a general claim about PEFT's ceiling relative to full fine-tuning.

Future work should prioritize tuning LoRA's rank, scaling, dropout, and target-module coverage to test how much of the gap is recoverable, ideally alongside a small human-validated Nepali NLI subset to calibrate absolute performance and disentangle model limitations from translation artifacts. Multi-source training with related Devanagari-script languages such as Hindi, or light domain-adaptive pretraining on Nepali web and news text, may further improve cross-lingual alignment, while expanding the evaluation set beyond 500 items with targeted challenge subsets for negation, discourse markers, numerals, and named entities would sharpen error analysis and calibration reporting. Additionally, since our evaluation set relies on unreviewed machine translation (NLLB-200) with no human post-editing, a natural next step is a small human-validated subsample to quantify translation error rate and disentangle model limitations from translation noise.

REFERENCES

- [1] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. EMNLP*, 2015, doi: 10.18653/v1/d15-1075.
- [2] A. Conneau, G. Lample, R. Rinott, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov, "XNLI: Evaluating cross-lingual sentence representations," *arXiv:1809.05053*, 2018.
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2018.
- [4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *arXiv:2106.09685*, 2021.
- [5] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?," *arXiv:1906.01502*, 2019.
- [6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv:1911.02116*, 2019.
- [7] T. Wolf, L. Debut, V. Sanh, et al., "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP: System Demonstrations*, 2020, doi: 10.18653/v1/2020.emnlp-demos.6.
- [8] Q. Lhoest, A. Villanova del Moral, Y. Jernite, et al., "Datasets: A community library for natural language processing," *arXiv:2109.02846*, 2021.
- [9] N. Hounsby, A. Giurigu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," *arXiv:1902.00751*, 2019.
- [10] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1994.
- [11] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," *arXiv:2305.14314*, 2023.
- [12] A. Rouhe, S.-A. Grönroos, S. Virpioja, M. Creutz, and M. Kurimo, "Morfessor-enriched features and multilingual training for canonical morphological segmentation," in *Proc. 19th SIGMORPHON Workshop*, 2022, pp. 144–151, doi: 10.18653/v1/2022.sigmorphon-1.16.
- [13] K. Kann et al., "AmericasNLI: Machine translation and natural language inference systems for Indigenous languages of the Americas," *Front. Artif. Intell.*, vol. 5, Art. no. 995667, 2022, doi: 10.3389/frai.2022.995667.
- [14] P. Dhakal and D. S. Baral, "Abstractive summarization of low-resourced Nepali language using multilingual transformers," *arXiv:2409.19566*, 2024.
- [15] A. K. Htet and M. Dras, "Myanmar XNLI: Building a dataset and exploring low-resource approaches to natural language inference with Myanmar," *Language Resources and Evaluation*, *arXiv:2504.09645*, 2025.
- [16] F. T. J. Faria, M. B. Moin, A. I. Fahim, P. Debnath, and F. M. Shah, "Unraveling the dominance of large language models over transformer models for Bangla natural language inference: A comprehensive study," *arXiv:2405.02937*, 2024.
- [17] J. Nyachhyon, M. Sharma, P. Thapa, and B. K. Bal, "Consolidating and developing benchmarking datasets for the Nepali natural language understanding tasks," in *Findings of IJCNLP*, 2025, *arXiv:2411.19244*.