

Unlocking Parameter-Efficient Fine-Tuning for Low-Resource Language Translation

Tong Su¹ Xin Peng¹ Sarubi Thillainathan²
David Guzmán¹ Surangika Ranathunga³ En-Shiun Annie Lee^{1,4}
¹University of Toronto ²Saarland University ³Massey University
⁴Ontario Tech University

Abstract

Parameter-efficient fine-tuning (PEFT) methods are increasingly vital in adapting large-scale pre-trained language models for diverse tasks, offering a balance between adaptability and computational efficiency. They are important in Low-Resource Language (LRL) Neural Machine Translation (NMT) to enhance translation accuracy with minimal resources. However, their practical effectiveness varies significantly across different languages. We conducted comprehensive empirical experiments with varying LRL domains and sizes to evaluate the performance of 8 PEFT methods with in total of 15 architectures using the SacreBLEU score. We showed that 6 PEFT architectures outperform the baseline for both in-domain and out-domain tests and the Houlby+Inversion adapter has the best performance overall, proving the effectiveness of PEFT methods.

1 Introduction

Advances in large-scale pre-trained language models have transformed the field for high-resource languages (Min et al., 2023), but these data and compute-hungry models are not viable for the more-than-7000 low-resource languages (LRLs) in the world (Stap and Araabi, 2023; Robinson et al., 2023; Zhang et al., 2023). Ideal for the limitations of LRLs, parameter-efficient fine-tuning (PEFT) methods (Houlsby et al., 2019; Pfeiffer et al., 2020b; Hu et al., 2021) are designed to strategically update a small number of parameters within a pre-trained model to be more efficient and adaptable without retraining the entire model. Their architecture significantly saves computational resources and storage space while achieving results comparable to full fine-tuning in downstream tasks (Ruder et al., 2022). Üstün and Stickland (2022) examined the applicability of 4 PEFT methods specifically in the context of language translation. Moreover, it did not address truly LRLs (Üstün and

Stickland, 2022), nor did it incorporate variation in domains that would allow for an assessment of the models' generalization capabilities.

As a result, while the PEFT methods have shown potential in fine-tuning specific tasks, domains, and languages, the effectiveness of this collection of PEFT methods for LRL translation has not been systematically examined. In this paper, we explore the performance of different PEFT architectures in the LRL Neural Machine translation (NMT) by comparing in-domain and out-of-domain test results, as well as training times. We also investigate the effectiveness of PEFT methods in translating LRLs, focusing specifically on their architectures and performance across various datasets.

The contributions of our paper are 1) comprehensive experimentation of PEFT architectures to reveal the suitability of translating non-Latin scripts and LRL pairs; 2) an in-depth assessment of 15 PEFT architectures using 8 distinct methods to evaluate their effectiveness in LRL translation; and 3) a systematic exploration of experimental settings, including variations in dataset domains and sizes, aimed at enhancing model generalization capabilities. As the field continues to advance rapidly, these PEFT guidelines provide practical recommendations for improving LRL translations, thus narrowing the language gap.

2 The PEFT Methods

We focus on the comparative performance of an extensive list of PEFT methods for LRL NMT under various settings (Figure 1), offering a broader and distinctive understanding of adapter utility.

Among all the PEFT methods, some share the same structure. For example, the bottleneck adapters include bottleneck feed-forward layers in each layer of a transformer model. These layers can be added to various positions within transformer blocks. The Houlby adapter (Houlsby et al., 2019)

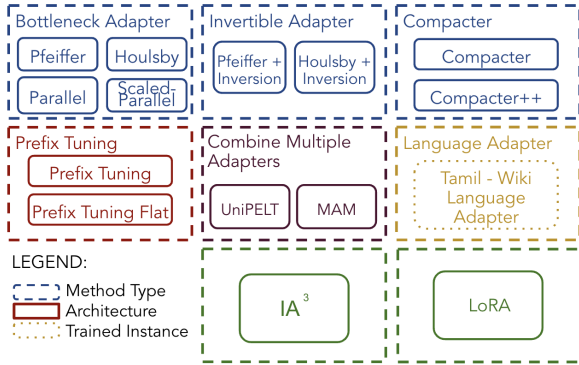


Figure 1: Full list of 8 PEFT methods and 15 architectures. Each color box represents a specific structure appearing in the PEFT methods. The same color represents the PEFT methods that share a similar structure.

adds the layers after both the multi-head attention and feed-forward blocks. The Pfeiffer adapter (Pfeiffer et al., 2020b) only adds the layers after the feed-forward block. The Parallel adapter (He et al., 2021) deploys the layers parallel to the transformer layers. Similarly, the invertible adapters share a similar architecture with bottleneck adapters but with an added invertible adapter layer to the language model embedding layer. The Compacter architecture replaces only the linear down-projection and up-projection with a parameterized hypercomplex multiplication layer (Karimi Mahabadi et al., 2021).

In addition, Prefix Tuning is a lightweight alternative inspired by prompting (Li and Liang, 2021) that introduces additional parameters in the multi-head attention blocks of each transformer layer. The LoRA method allows the training of specific dense layers in a neural network indirectly by optimizing the rank-decomposition matrices of specific dense layers during adaptation with the pre-trained weights frozen (Hu et al., 2021). $(IA)^3$ is built to improve LoRA with modifications. While LoRA uses additive composition, $(IA)^3$ uses element-wise multiplication (Liu et al., 2022).

Some PEFT methods combine multiple methods. The Mix-and-Match (MAM) Adapter combines LoRA, Prefix Tuning, and Parallel adapter to form a new adapter (He et al., 2021). Similarly, UniPELT integrates bottleneck adapters, Prefix Tuning, and LoRA into a unified setup (Mao et al., 2021).

Lastly, the language adapter captures language-specific knowledge for application in various downstream tasks. It is not a distinct adapter architecture; rather, it represents a method of utilizing pre-existing architectures. We expected that this ap-

proach would enhance the model’s performance, given its preexisting familiarity with the language in question. We employed a pre-existing bottleneck adapter for diverse language datasets, training it with Masked Language Modelling on an extensive collection of articles (Pfeiffer et al., 2020c).

3 Experimental Setup

LRLs Selection We chose Sinhala (SI), Tamil (TA), Hindi (HI), and Gujarati (GU) as our primary languages to run our translation task (See Table 1). SI and TA were paired to run the translation task in both directions, and HI and GU were paired.

Language	Family	Joshi class	mBART coverage in tokens (M)
Hindi (HI)	Indo Aryan	4	1715
Gujarati (GU)	Indo Aryan	1	140
Sinhala (SI)	Indo Aryan	1	243
Tamil (TA)	Dravidian	3	595

Table 1: Language details. The smaller the value of the Joshi et al. (2020) class, the more low-resource the language is.

Data Collection The data summary is given in Table 2. More details about the datasets can be found in Appendix A.1. Note that No Language Left Behind (NLLB) (Costa-jussà et al., 2022) corpora are derived from metadata for bitext mining released by Meta AI, which lacks coverage and human quality control, and is only suitable for training purposes. Therefore, we performed an out-of-domain evaluation by using FLORES-101 (Goyal et al., 2022) and FLORES-200 (Costa-jussà et al., 2022) as the test dataset.

Pre-trained Model Selection We performed baseline experiments by fine-tuning all parameters using several pre-trained models, including mBART-50 (Tang et al., 2020), M2M-100 (Fan et al., 2020), and NLLB (Costa-jussà et al., 2022). Model selection process is given in Appendix A.2. Based on our results, we selected mBART-50 as our pre-trained model for the rest of our experiments. The mBART-50 model is a multilingual Sequence-to-Sequence (Seq2Seq) model. Its introduction aims to demonstrate the feasibility of developing multilingual translation models via the process of multilingual fine-tuning.

Experimental Design We also experimented by systematically varying the number of fine-tuned parameters that the method updates with the Housby adapter. It allowed us to investigate the impact of

Dataset	Quality	Languages	Train Size	Test Size
FLORES-101	Sourced from English Wikipedia and translated by professional translators	HI, GU, TA	Test only	1k
FLORES-200	Sourced from web articles and translated by professional translators	SI	Test only	1k
NLLB	Automatically gathered from web sources and monolingual datasets, using web crawls and LASER3 encoders for parallel sentence identification	HI, GU, SI, TA	25k, 100k	2k
Gvt	Parallel government documents dataset with manual cleaning and aligning	SI, TA	25k	2k
Sam	Sourced both from existing corpora and new, diverse data collected via automated web crawling and sentence alignment, with human evaluation ensuring its reliability	HI, GU	25k	2k

Table 2: Dataset statistics

the number of parameters that are updated and then select the most suitable reduction factor for all the PEFT architectures.

The trainer employed in our study is sourced from the Adapter Transformers (Pfeiffer et al., 2020a). Each adapter’s performance was evaluated using the Sacre BiLingual Evaluation Understudy (SacreBLEU) Score (Post, 2018). Training details are given in Appendix A.2.

We evaluated the performance of our PEFT architectures using direct fine-tuning with the pre-trained model as the baseline. In total, we tested 15 PEFT architectures supported by the Hugging Face Adapter Hub (Pfeiffer et al., 2020a) trained on SI-TA 100k NLLB language dataset to identify the best methods for further analysis; both the NLLB test dataset and the FLoRes test dataset were used to test these models. We then narrowed down the selection to the top two methods with the highest SacreBLEU scores from each of the test results, the NLLB and the FLoRes dataset. An additional PEFT architecture was selected based on those that outperformed the baseline for both test datasets and with the shortest training time. Extensive experiments were then conducted with these top-selected methods across additional LRL and dataset sizes to determine the optimal configuration. After all experiments were completed, the average performance was calculated to mitigate any variation due to GPU randomness.

4 Experimental Results

Number of fine-tuned parameters Table 3 shows that the performance of the same PEFT architecture can vary with the number of parameters. Initially, when the reduction factor is set to 2, both the in-domain and out-domain results show improvement over the baseline. Specifically, the in-domain performance increases significantly to **33.34**, while the out-domain performance also sees a modest improvement to 7.62. However, an interesting trend emerges as the reduction factor is further increased to 4: the in-domain test results

begin to decline, dropping to 30.67, while the out-domain results experience only a slight increase (0.07 compared to reduction factor 2). This pattern suggests that increasing the reduction factor beyond 2 may lead to underfitting. Subsequent increases in the reduction factor exacerbate this trend, causing both in-domain and out-domain results to decrease compared to the reduction factor of 2. Therefore, the reduction factor of 2 is considered optimal for the remainder of the experiments, balancing model complexity with performance gains.

Reduction factor	# PEFT parameters	% PEFT parameters	In-domain (SacreBLEU)	Out-domain (SacreBLEU)	Runtime (hours)
-	-	-	30.25	5.52	59.44
2	50,405,376	7.62	33.34	7.62	78.65
4	25,227,264	3.97	30.67	7.69	53.83
8	12,638,208	2.03	31.05	7.52	93.58
16	6,343,680	1.03	26.67	7.12	76.73
32	3,196,416	0.52	23.81	7.35	74.18

Table 3: Comparison of Fine-tuning Results with Different Reduction Factors on 100k NLLB SI-TA with mBART-50 Using the Houlby Adapter. The first line represents the full fine-tuning baseline without using any PEFT architecture, the second line onwards shows fine-tuning with the Houlby Adapter with different reduction factors.

Top-4 Selected PEFT Architectures To evaluate the PEFT architectures’ performance, we compared their in-domain test, out-domain test, and training time for 100k NLLB SI-TA training dataset (Table 4). For methods that did not surpass the baseline in both tests, we inferred that these methods are not suitable for tasks in LRL translation.

For NLLB in-domain testing, the Houlby adapter performs the best at **33.34** (10.20% better than baseline), followed by Scaled-parallel (9.21% improvement). For FLoRes out-of-domain testing, the Houlby adapter remains the best at **7.62** (38.23% better than baseline) followed by Houlby+Inversion adapter (34.51% improvement). The Pfeiffer adapter runs the fastest at **52.59** while outperforming the baseline for both tests.

Domain Similarity of Test Dataset We expanded our training to additional dataset do-

Architecture	In-domain	$\Delta\%$	Out-domain	$\Delta\%$	Runtime (hours)	$\Delta\%$
Baseline	30.25	-	5.52	-	59.44	-
Houlsby	33.34	10.20% [1]	7.62	38.23% [1]	78.65	32.32% [10]
Scaled-parallel	33.04	9.21% [2]	6.62	20.00% [7]	93.68	57.60% [12]
Pfeiffer+Inversion	32.58	7.69% [3]	6.84	24.06% [4]	78.31	31.75% [9]
MAM	33.26	6.62% [4]	6.51	18.08% [8]	95.73	61.05% [13]
Houlsby+Inversion	32.23	6.55% [5]	7.42	34.51% [2]	63.54	6.90% [6]
Pfeiffer	31.24	3.27% [6]	6.96	26.25% [3]	52.59	-11.52% [3]
Language Adapter (TA)	29.98	-0.88% [7]	6.31	14.47% [9]	98.23	65.26% [14]
Parallel	27.63	-8.66% [8]	6.62	20.04% [6]	26.85	-54.83% [1]
Prefix tuning	23.62	-21.93% [9]	6.71	21.72% [5]	77.25	29.96% [8]
LoRA	18.63	-38.41% [10]	5.76	4.45% [10]	58.1	-2.25% [5]
Compacter	13.36	-55.82% [11]	4.27	-22.61% [11]	106.56	79.27% [15]
Compacter++	12.56	-58.49% [12]	4.12	-25.36% [12]	84.22	41.69% [11]
Prefix tuning flat	12.25	-59.50% [13]	3.93	-28.75% [13]	55.29	-6.98% [4]
(IA) ³	11.10	-63.30% [14]	3.63	-34.14% [14]	63.81	7.35% [7]
Unipelt	0.38	-98.74% [15]	0.12	-72.54% [15]	39.47	-33.60% [2]

Table 4: Full list of fine-tuning results with the 100k NLLB SI-TA language dataset. The table shows the predicted SacreBLEU score for both the In-domain test dataset (the NLLB test dataset), the Out-domain test dataset (the FLoRes test dataset), and the models’ training time. $\Delta\%$ represents the percentage increase in terms of the baseline results. **Bold** means that the model’s performance is better than the baseline (higher SacreBLEU score/shorter training time). Underline means that the corresponding PEFT architectures are selected for further testing.

mains (Appendix Table 6). For the in-domain test, Houlsby adapter exhibits superior performance at **31.53**; for the out-of-domain test, Houlsby+Inversion performs best at **10.02** (a 0.1 better than Houlsby). Since the FLoRes out-of-domain test results in a more robust and objective evaluation of the model’s translation performance across many domains (Goyal et al., 2022), we prioritize the out-of-domain results and conclude that the Houlsby+Inversion adapter has the best performance overall. Lastly, in terms of training time (Appendix Table 7), the Pfeiffer adapter has the shortest runtime as expected, saving 8 hours on average compared to the baseline.

5 Discussion

Result Generalization Our results demonstrate the robust generalizability of our PEFT architectures across different training dataset sizes and domains. Figure 2 shows that our model consistently outperforms the baseline, on average, in both in-domain and out-of-domain testing. Specifically for models trained on other domains, the $\Delta\%$ increase over the baseline is over 50%, demonstrating the ability of PEFT methods to excel at tasks beyond their training domain. In terms of training dataset sizes, our selected PEFT architectures show a continuous trend for performance increase compared to the baseline. It is worth noting that our Table 6 in the appendix shows that increasing the training size

has led to improved performance. However, the magnitude of the improvement difference shows diminishing returns, suggesting a potential saturation effect as identified in previous studies (Lee, 2021).

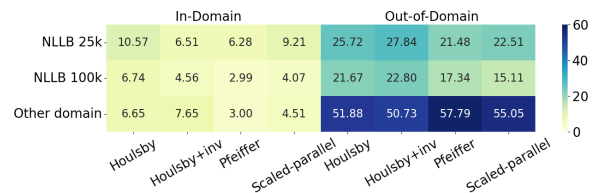


Figure 2: Average $\Delta\%$ compared to baseline for each dataset tested on in-domain and out-of-domain.

Effect of Bottleneck Architecture on LRL The similarity among the outperforming PEFT architectures highlighted in Table 4 is that they all include bottleneck adapters in the architecture. While some other PEFT architectures such as UniPELT and Compacter adapters also adapt the bottleneck architecture, they do not exhibit comparable performance. In the subsequent discussion, we will examine the difference between these architectures and outperforming ones to find out which part of the bottleneck architecture design makes the performance better.

First, UniPELT integrates bottleneck adapters into a unified setup. Compared with the MAM adapter, UniPELT adds the bottleneck only after the Feed-Forward Network (FFN) layers (Mao et al., 2021), while MAM also adds an adapter after the

attention layer (He et al., 2021). This observation suggests that the efficiency of the architecture is not solely determined by the presence of the bottleneck adapter, but also by the specific placement of the adapter within the architecture. This could be because FFN learns task-specific text patterns (Geva et al., 2022), while attention learns pairwise positional interactions. In our LRL translation task, pairwise positional interactions are more important than textual patterns. Removing the adapter from the attention layer may lead to a decrease in the SacreBLEU score.

Second, Compacter modifies the bottleneck adapter but does not achieve good results. The difference between a bottleneck adapter and Compacter is that it replaces the linear down and up projection with a parameterized hypercomplex multiplication (PHM) layer. This layer can break the importance of the parallel position in the translation task. This highlights the importance of the original up-and-down projection layer present in the bottleneck adapters.

Adapter for Domain Adaptation We found that in-domain testing performs better than out-of-domain testing due to memorizing patterns in the dataset, leading to falsely inflated performance. When fine-tuning on a new domain, rapid domain-specific overfitting and catastrophic forgetting reduce the performance on all other domains (Sennrich et al., 2015; Barone et al., 2017; Bapna et al., 2019). However, by freezing the parameters of the original pre-trained model and training only task-specific parameters, the adapter avoids catastrophic forgetting of the knowledge learned during pre-training and can maintain performance when testing in other domains (McCloskey and Cohen, 1989; Lai et al., 2022; Üstün et al., 2021).

Language Family and Pre-Training Size We observed notable disparities in performance among different language pairs (Figure 3). The LRL SI-TA pair demonstrates lower performance with a smaller dataset size (i.e., 25k) but improves as the dataset size increases, suggesting that the amount of training data is a critical factor in enhancing the translation quality for LRL (Lee et al., 2022).

The SI-TA pair yields lower performance compared to the HI-GU pair, underscoring the intricate dynamics of linguistic relationships and the availability of resources (Table 1). Linguistically, HI, GU and SI are part of the Indo-Aryan language family, while TA is Dravidian; thus suggesting the lower performance of SI-TA. Notably, GU’s closer

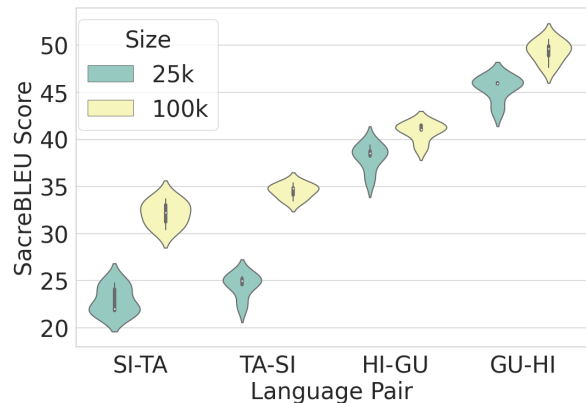


Figure 3: Performance of LRL Translation Pairs by Fine-Tuning Dataset Size (In-Domain only).

linguistic affinity to HI may have facilitated enhancing its performance through cross-lingual transfer, despite its smaller pre-training dataset size. However, its smaller gains due to dataset size increase may be due to the high-resource saturation of HI.

6 Conclusion

Our study delved into a wide range of PEFT methods to identify the most effective ones for LRL-NMT. Particularly focusing on non-Latin scripts and LRL-to-LRL translation pairs, our research stands as a valuable guide for LRL-NMT. We found that certain adapters consistently outperformed others, offering enhanced translation accuracy and efficiency in challenging linguistic contexts. Furthermore, the adapters’ effectiveness was tested and generalized across various dataset domains and sizes, ensuring the applicability of our findings to a broad spectrum of LRL scenarios. Looking ahead, these insights pave the way for further advancements in PEFT methods, aiming to optimize the balance between efficiency and quality in NMT, especially in the challenging context of LRL.

Limitation and Future Work

Language Specific Adapters We tested the PEFT architectures at adapting to our LRLs, and not the specific fine-tuned models of language-specific adapters. We hope this comparison can provide an agnostic baseline for others to follow. Surprisingly, the language adapter we tested does not perform above the baseline; therefore, we need to explore other language-specific fine-tuning strategies. In the future, we will explore more language-specific adapter; but the scope of this study only covers the generic PEFT architectures.

Increase Domain While it is worth noting that three of the four LRLs we have provided translations for belong to the Indo-Aryan language family and the other one is a Dravidian language, we suggest broadening our experimentation to include more diverse languages to increase the credibility of our results. As with dataset sizes of 100k and 25k, we could experiment with sizes in between.

Evaluation Criteria Our assessment of translation performance relied on SacreBLEU scores, but relying on a single metric may not be sufficient to support our conclusions. In future research to evaluate the model’s performance, it would be advantageous to use metrics such as ChrF and COMET, which are reportedly better correlated with human judgments (Dixit et al., 2023). Additionally, the variations between distinct methods lack strong indications. Consequently, statistical significance tests would be fundamental to further confirm the significance of the improvements.

PEFT Composition This paper focuses solely on the impact of a single PEFT architecture. However, there is an ongoing exploration into the potential of combining multiple methods as a composition. AdapterHub recently published a paper that expanded its support to include various composition methods, including stack, fuse, split, and average (Poth et al., 2023).

References

- Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.
- Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. *arXiv preprint arXiv:1707.09920*.
- Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Tanay Dixit, Vignesh Nagarajan, Anoop Kunchukuttan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, et al. 2023. Indicmt eval: A dataset to meta-evaluate machine translation metrics for indian languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14210–14228.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Man-deep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. *Beyond english-centric multilingual machine translation*. *CoRR*, abs/2010.11125.
- Aloka Fernando, Surangika Ranathunga, and Gihan Dias. 2020. Data augmentation and terminology integration for domain-specific sinhala-english-tamil statistical machine translation. *arXiv preprint arXiv:2011.02821*.
- Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. *Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space*. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.
- Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035.
- Wen Lai, Alexandra Chronopoulou, and Alexander Fraser. 2022. *m⁴adapter*: Multilingual multi-domain adaptation for machine translation with a meta-adapter. *arXiv preprint arXiv:2210.11912*.
- En-Shiun Annie Lee, Sarubi Thillainathan, Shravan Nayak, Surangika Ranathunga, David Ifeoluwa Adedani, Ruisi Su, and Arya D McCarthy. 2022. Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation? *arXiv preprint arXiv:2203.08850*.

- Yohan Lee. 2021. [Improving end-to-end task-oriented dialog system with a simple auxiliary task](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohata, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.
- Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. 2021. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*.
- Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.
- Bonan Min, Hayley Ross, Elicor Sulem, Amir Poursan Ben Veysheh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.
- Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020c. [MAD-X: An Adapter-based Framework for Multi-task Cross-lingual Transfer](#). *arXiv preprint*.
- Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.
- Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. [Adapters: A unified library for parameter-efficient and modular transfer learning](#).
- Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2023. [Samanantar: The largest publicly available parallel corpora collection for 11 indic languages](#).
- Nathaniel Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. [ChatGPT MT: Competitive for high- \(but not low-\) resource languages](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 392–418, Singapore. Association for Computational Linguistics.
- Sebastian Ruder, Jonas Pfeiffer, and Ivan Vulić. 2022. [Modular and parameter-efficient fine-tuning for NLP models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 23–29, Abu Dhabi, UAE. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.
- David Stap and Ali Araabi. 2023. [ChatGPT is not a good indigenous translator](#). In *Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)*, pages 163–167, Toronto, Canada. Association for Computational Linguistics.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. [Multilingual translation with extensible multilingual pretraining and finetuning](#).
- Ahmet Üstün, Alexandre Berard, Laurent Besacier, and Matthias Gallé. 2021. Multilingual unsupervised neural machine translation with denoising adapters. *arXiv preprint arXiv:2110.10472*.
- Ahmet Üstün and Asa Cooper Stickland. 2022. When does parameter-efficient transfer learning work for machine translation? *arXiv preprint arXiv:2205.11277*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Xiang Zhang, Senyu Li, Bradley Hauer, Ning Shi, and Grzegorz Kondrak. 2023. [Don't trust ChatGPT when your question is not in English: A study of multilingual abilities and types of LLMs](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7915–7927, Singapore. Association for Computational Linguistics.

A Appendix

A.1 Supplementary Material on Datasets

No Language Left Behind (NLLB) The NLLB (Costa-jussà et al., 2022) corpus consists of translation training datasets for low-resource languages and is automatically created through the process of bitext mining. We employed a selection process based on the LASER score, where we chose the top 100,000 and 25,000 translation pairs from the selected language pair for dataset size variation. However, NLLB lacks coverage and human quality control due to the noisy nature of the entire procedure and is only suitable for training purposes.

Government corpus (Govt) The government document corpus (Fernando et al., 2020) is a multi-way parallel corpus for Sinhala, Tamil, and English. It comprises a range of official Sri Lankan government documents, including annual and committee reports, content sourced from government websites, procurement-related documents, and legislative acts.

Samanantar corpus (Sam) The Samanantar corpus (Ramesh et al., 2023) is the largest publicly available Parallel Corpora Collection for 11 Indic Languages. The data is derived from two sources: existing databases and new data automatically collected through web crawling and sentence alignment techniques.

FLORES The FLORES dataset (Goyal et al., 2022) is a multiway multilingual translation evaluation dataset. FLORES-101 is comprised of translations from 842 unique web articles, comprising a total of 3001 sentences. Because all translations are fully aligned, the resulting dataset allows for a more accurate assessment of model quality on the long tail of LRLs, including the evaluation of many-to-many multilingual translation systems. The professional rigor and reliability of the results are strengthened by using an out-of-domain evaluation of this type, resulting in a more robust and objective evaluation of the model’s translation performance across many domains. FLORES-200 expands the language coverage to twice that of FLORES-101. We used FLORES-200 (Costa-jussà et al., 2022) for Sinhala since it is not in FLORES-101, and dev-test split for both FLORES-101 and FLORES-200.

A.2 Supplementary Material on Experimental Setup

Selection of Pre-trained Models. We conducted experiments with several MT models, such as

mBART-50 (Tang et al., 2020), M2M-100 (Fan et al., 2020) and NLLB (Costa-jussà et al., 2022). Specifically, we fine-tuned these models on the SI-TA 100k NLLB language dataset to identify the most effective methods for further analysis. Both the NLLB test dataset and the FLoRes test dataset were utilized to evaluate the performance of these models. Subsequently, we narrowed down the selection criteria to prioritize models with high SacreBLEU scores, low runtime, and lower computational resources. We found that nllb-200-distilled-1.3B, the largest model that we experimented with, has the best performance in both the in-domain and out-domain test sets (Table 5). However, the superior performance of larger models comes with the caveat of increased hardware requirements, making them less accessible for practitioners, particularly when it comes to LRLs. In contrast, mBART-50 offers a robust alternative that does not demand additional computational resources, making it a practical choice for LRL applications. With these factors into account, we chose mBART-50 for our experiments.

Model	# of parameters	In-domain	Out-domain	Runtime (hours)
m2m100-418M	483,905,536	32.35	6.03	75.96
mbart-large-50	610,879,488	30.25	5.52	59.44
nllb-200-distilled-600M	615,071,744	35.15	9.25	50.49
m2m100-1.2B	1,239,470,080	32.22	6.22	187.91
nllb-200-distilled-1.3B	1,370,636,288	37.75	10.30	60.96

Table 5: Baseline experiments with different pre-trained models with the 100k NLLB SI-TA language dataset.

Choice of Trainer The integration of PEFT methods into language models is facilitated by a modification of AdapterHub, a centralized store of pre-trained adapter modules.

In the context of language translation, the process involves utilizing a translation code to refine the pre-existing model and assess the performance of transformers to translation-oriented assignments. In this case, we used Seq2SeqTrainingArguments.

GPU Details It consists of Dell nodes, each equipped with four NVIDIA V100-32GB GPUs, 32 CPU cores, 32GB of GPU memory, and two Intel Silver 4216 Cascade Lake processors running at 2.1GHz. All GPUs are connected via NVLink and SXM2. They are well suited for processing large language models with a 7.0 capability.

Trainer Setup There are several parameters that we have specified for the execution of the model. For the evaluation strategy, the evaluation is done at the end of each epoch (Wolf et al., 2020). We set the number of training epochs to 40 so that the

Language	Dataset	Size	No PEFT		Houlsby		Houlsby+inv		Pfeiffer		Scaled-parallel	
			In-domain	FLoRes	In-domain	FLoRes	In-domain	FLoRes	In-domain	FLoRes	In-domain	FLoRes
SI-TA	NLLB	25k	21.8171	3.9573	24.7268 (+2.9097)	5.7709	21.6649	5.8532 (+1.8959)	21.9808	5.4773	24.0997	5.3101
		100k	30.3961	5.4352	33.6794 (+3.2833)	7.6977 (+2.2625)	32.2317	7.4188	31.2395	6.9635	33.0374	6.6186
	Gvt	25k	21.2982	1.3255	21.0242	2.2491	21.6247 (+0.3265)	2.1965	19.5961	2.347 (+1.0215)	20.5064	2.0723
TA-SI	NLLB	25k	22.3512	5.3989	25.1825	6.641	25.434 (+3.0828)	7.0094 (+1.6105)	24.5575	6.1987	24.9486	6.4323
		100k	34.0925	7.1264	35.3707 (+1.2782)	8.3163	34.8269	8.6788 (+1.5524)	34.7869	7.9525	33.4139	7.8196
	Gvt	25k	31.9105	2.4346	31.7150	3.2406	31.7034	3.259	28.6959	3.2433	28.86	3.3824 (+0.9478)
HI-GU	NLLB	25k	35.8082	11.2997	39.3775 (+3.5693)	12.3927	38.2209	12.4318	38.7203	12.4832	38.4944	12.807 (+1.5073)
		100k	39.1754	12.0767	41.5658 (+2.3904)	14.2947	41.4993	15.057 (+2.9803)	40.9938	14.5054	41.0432	14.2797
	Sam	25k	11.1118	5.2094	12.6581	9.5945	12.6111	9.0768	12.7405	9.9535 (+4.7441)	12.8279 (+1.7161)	9.9509
GU-HI	NLLB	25k	43.2111	13.9272	45.9313	17.3196 (+3.3924)	45.8927	17.2129	45.9704	17.0236	46.341 (+3.1299)	17.1825
		100k	47.6282	17.5709	50.6256 (+2.9974)	19.3265	49.5878	19.0191	48.826	19.2495	49.9162	19.4532 (+1.8823)
	Sam	25k	14.3543	10.0847	16.4453	12.1565	16.6844 (+6.5997)	13.0219	16.5667	13.0903	16.6316	13.5055 (+3.4208)
Average		29.4296	7.9872	31.5252	9.9167	30.9985	10.0196	30.3895	9.8740	30.8434	9.9012	

Table 6: Comparison of Fine-Tuning Results for Selected PEFT Methods Across Various Language Datasets and Dataset Sizes on the in-domain Test Datasets and FLoRes Test Datasets. In-domain means that the test dataset comes from the same distribution as the training dataset. **Bold** score means that the SacreBLEU score is the highest among all listed fine-tuning experiments within the same dataset.

Language	Dataset	Size	No PEFT	Houlsby	Houlsby+inv	Pfeiffer	Scaled-parallel
SI-TA	NLLB	25k	00-14:22:48	00-22:10:17	00-17:20:46	00-08:41:54	00-16:36:20
		100k	02-23:47:07	03-12:06:21	02-15:32:23	02-04:35:37 (-19:11:30)	03-21:40:44
	Gvt	25k	01-20:35:13	00-23:09:18	01-06:25:42	00-10:29:23 (-01-10:05:50)	00-18:55:42
TA-SI	NLLB	25k	00-09:51:53	00-19:04:15	01-06:57:42	00-21:56:10	00-21:12:29
		100k	03-23:18:56	03-21:35:37	03-00:14:17	03-19:41:43	02-13:40:04 (-01-09:38:52)
	Gvt	25k	02-01:01:03	01-14:06:04	02-02:11:33	00-20:36:14	00-10:26:10 (-01-14:34:53)
HI-GU	NLLB	25k	00-07:42:33	00-17:45:38	00-10:43:29	00-15:47:21	00-06:50:13
		100k	01-05:37:21	01-02:22:44	01-00:18:21	00-19:28:39 (-10:08:42)	00-22:16:01
	Sam	25k	00-16:27:37	00-07:43:53	00-07:27:22	00-05:51:51	00-05:29:49 (-10:57:48)
GU-HI	NLLB	25k	00-07:34:30	00-04:59:17 (-02:35:13)	00-07:20:46	00-05:47:51	00-06:23:02
		100k	00-20:17:54	01-07:19:39	01-02:59:59	00-21:23:38	00-20:35:02
	Sam	25k	00-04:54:57	00-04:54:34	00-05:51:34	00-04:59:19	00-04:46:03 (-00:08:54)
Average		01-06:57:39	01-07:06:28	01-04:57:00	00-23:16:38	01-00:04:18	

Table 7: Comparison of Training Time for Selected PEFT Methods Across Language Datasets and Dataset Sizes. **Bold** time means that the training time is the shortest among listed fine-tuning experiments with the same dataset.

Parameter	Value
Evaluation Strategy	Epoch
Number of Training Epoch	40
Patience	3
Batch Size	2
Metric for Best Model	Evaluation SacreBLEU

Table 8: Full list of trainer parameters used and corresponding value.

model could be finished running in a maximum of 4 days. The patience level is set to 3 based on some small experiments. A lower level of patience will cause the model to stop too early as there is still room for improvement; a higher level of patience will cause overfitting, and the model will only stop until the last epoch; there will be no early stopping, which is not what we expected. Since our task is simple fine-tuning, we set the batch size to 2. A smaller batch size introduces more stochasticity into the training process by updating the model parameters more frequently.

Evaluation Metrics SacreBLEU (Post, 2018) offers benefits over BLEU scores, which cannot be directly compared across papers, as it allows for easy computation of shareable, comparable, and reproducible SacreBLEU scores.

B Direct Fine-Tuning Results With Selected PEFT Architectures Across Different Domains

We selected Houlsby, Houlsby+Inversion, and Scaled-Parallel Adapter for the next experiments based on their performance, with Houlsby emerging as the best performer for both testing results. Pfeiffer adapter was selected for its short training time compared to the baseline. The results displayed in Table 6 indicate that the Houlsby adapter exhibit superior performance over all other methods in the in-domain test with an average SacreBLEU score of **31.5252**. For the FLoRes test dataset, Houlsby+Inversion performs better with an average SacreBLEU score of **10.0196**, a 0.1 difference from Houlsby.

In terms of training time shown in 7, the Houlsby adapter does not have the advantage and even becomes the longest runtime on average. The Pfeiffer adapter, which we chose for its runtime, has the shortest runtime as expected, saving 8 hours on average compared to the baseline.