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

Anonymous ACL submission

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 the Houlsby+Inversion adapter outperforms the baseline, 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 the need to retrain the entire model. Their architecture resulted in significant savings in computational resources and storage space while achieving results comparable to full fine-tuning in downstream tasks (Ruder et al., 2022).

While the above PEFT methods showed their advantages for 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 explored different PEFT architectures' performance in the LRL Neural Machine translation (NMT) by comparing in-domain test, out-of-domain test, and training time. We also investigated how PEFT methods can succeed in translating LRLs, specifically their structure and how they performed across different 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) systematic study of experimental settings such as dataset domains and size for generalization. As the field continues to advance rapidly, these PEFT guidelines provide practical recommendations for improving LRL translations, thus narrowing the language gap.

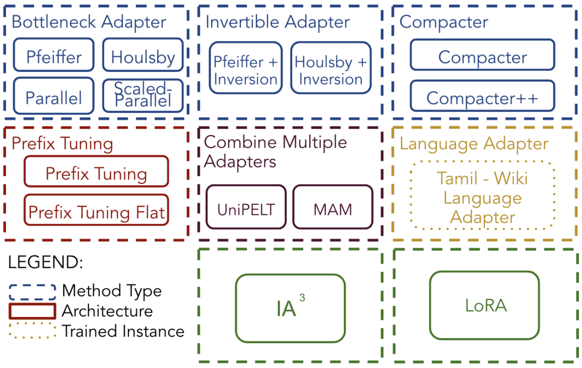

Figure 1.1: Full list of 8 PEFT methods and 15 architectures. Each color box represents a specific structure appearing in the PEFT methods. The same color represents the PEFT methods share similar sturcture

2 The PEFT Methods

We focus on the comparative performance of an extensive list of PEFT methods for LRL NMT under various settings (Figure 1.1), 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 Houlsby adapter (Houlsby et al., 2019) 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- 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 was 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 approach 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. Note that No Language Left Behind (NLLB) (Costa-jussà et al., 2022) corpora lacks coverage and human quality control, and is only suitable for training purposes. Therefore, we performed an out-of-domain evaluation by using FLoRes (Goyal et al., 2022) as the test dataset.

Dataset	Quality	Languages	Train Size	Test Size
FLoRes	Sourced from English Wikipedia and translated by professional translators	HI, GU, SI, TA	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

Experimental Design The pre-trained model we used is the mBART-50 model from Facebook (Tang et al., 2020). 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.1.

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 Results on the Effect of Various 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 3). 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 Houlsby 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 Houlsby adapter remains the best at **7.62** (38.23% better than baseline) followed by Houlsby+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 domains (Appendix Table 5). For the in-domain test, Houlsby adapter exhibited superior performance at **31.53**; for the out-of-domain test, Houlsby+Inversion performed best at **10.02** (a 0.1 better than Houlsby). Since the FLoRes out-of-domain test resulted 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 6), the Pfeiffer adapter has the shortest runtime as expected, saving 8 hours on average compared to the baseline.

Result Generalization Our results demonstrate the robust generalizability of our PEFT architectures across different training dataset sizes and domains. Figure 4.1 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 Δ% 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 showed a continuous trend for performance increase compared to the baseline. It is worth noting that our Table 5 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).

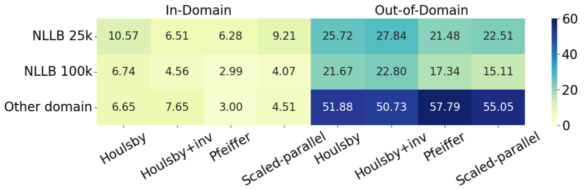

Figure 4.1: Average Δ% compared to baseline for each dataset tested on in-domain and out-of-domain

Language Family and Pre-Training Size We observed notable disparities in performance among different language pairs (Figure 4.2). 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).

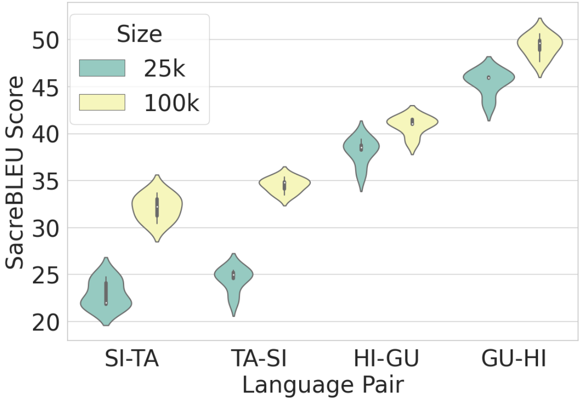

Figure 4.2: Performance of LRL Translation Pairs by Fine-Tuning Dataset Size (In-Domain only)

The SI-TA pair yielded 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

Method	In-domain	Δ%	Out-domain	Δ%	Runtime (hours)	Δ%
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	**33.04**	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 3: 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. Δ% 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.

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.

5 Discussion of Impact on Architectures

Bottleneck Architecture on LRL Our analysis reveals that the top-performing adapters employ the bottleneck adapter architecture, establishing a clear correlation between this design and performance.

Firstly, when fine-tuning with bottleneck, only task-specific parameters are fine-tuned. The original features are projected into a smaller dimension, thus preventing overfitting a large number of parameters to the limited dataset on LRL translation. This finding is consistent with that of (Bapna et al., 2019; Cooper Stickland et al., 2021a,b), which finds that bottleneck adapter controls the parameter count to keep at least the performance of the parent model.

Secondly, the skip connections facilitate information flows by bypassing one or more layers to allow input to be added directly to the output of the skipped layers. Parameters of projected layers initialized to near zero as near-identity initialization (Philip et al., 2020; Cooper Stickland et al., 2021a). The adapter module does not introduce significant input changes, allowing graduate learning. When fine-tuning downstream tasks with limited LRL data, parameters are not easily influenced by individual data, allowing for stable training.

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 reduces 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).

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 did 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 cover 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).

Ethics Statement

The code we used to run fine-tuning with methods is publicly available in AdapterHub (Pfeiffer et al., 2020a). The NLLB, Sam, and FloRes datasets we used for running the experiments are all publicly available. We received the Gvt corpus from Fernando et al. (2020).

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*.

Asa Cooper Stickland, Alexandre Berard, and Vassilina Nikoulina. 2021a. Multilingual domain adaptation for NMT: Decoupling language and domain information with adapters. In *Proceedings of the Sixth Conference on Machine Translation*, pages 578–598, Online. Association for Computational Linguistics.

Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2021b. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3440–3453, Online. Association for Computational Linguistics.

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.

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*.

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, Yuanzhi 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. $m^4adapter$: Multilingual multi-domain adaptation for machine translation with a meta-adapter. *arXiv preprint arXiv:2210.11912*.

En-Shiun Lee, Sarubi Thillainathan, Shravan Nayak, Surangika Ranathunga, David Adelani, Ruisi Su, and Arya McCarthy. 2022. Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 58–67, Dublin, Ireland. Association for Computational Linguistics.

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 Mohta, Tenghao Huang, Mohit Bansal, and Colin A 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, Elior Sulem, Amir Pouran Ben Veyseh, 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*.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470.

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, Aravinth 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 Dubai, 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*.

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 professionally translated sentences from the domain of Wikipedia and was obtained by taking samples from Wikimedia's List of Articles Every Wikipedia Should Have, covering various topics across different fields of knowledge and human activities. 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 construction using semi-automatic procedures (Goyal et al., 2022) and is only suitable for training purposes.

Government corpus (Gvt) The government document corpus (Fernando et al., 2020) is a multiway 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.

A.2 Supplementary Material on Experimental Setup

Choice of Pre-trained Model The mBART-50 model (Tang et al., 2020) is a multilingual Sequence-to-Sequence (Seq2Seq) model. Its introduction aimed to demonstrate the feasibility of developing multilingual translation models via the process of multilingual fine-tuning. Instead of singularly fine-tuning in a single direction, a pre-trained model undergoes simultaneous fine-tuning across many directions. The mBART-50 model is derived from the original mBART architecture and has been expanded to incorporate an additional 25 languages. This augmentation enables the development of multilingual machine translation models that can handle a total of 50 languages.

Paraemter Name	Value
Evaluation Strategy	Epoch
Number of Training Epoch	40
Patience	3
Batch Size	2
Metric for best model	Evaluation SacreBLEU

Table 4: Full list of trainer parameters used and corresponding value

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

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 5: 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 6: 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.

of transformers to translation-oriented assignments. In this case, we use 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 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 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 5 indicate that the Houlsby adapter exhibited 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 6, 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.