Investigating Adapters for Parameter-efficient Low-resource Automatic Speech Recognition

Ahnaf Mozib Samin[†§] **Shekhar Nayak**[§] **Andrea DeMarco**[†] **Claudia Borg**[†]
[†]Department of Artificial Intelligence, University of Malta, Malta
[§]University of Groningen, The Netherlands
{ahnaf.samin.22, andrea.demarco, claudia.borg}@um.edu.mt, s.nayak@rug.nl

Abstract

Recent years have witnessed the adoption of parameter-efficient adapters in pre-trained language models for natural language processing. Yet, their application in speech processing remains less studied. In this work, we explore the adapters for low-resource speech recognition, introducing a novel technique - ConvAdapt into pre-trained speech models. We investigate various aspects such as data requirements, transfer learning within adapters, and scaling of feed-forward layers in adapters. Our findings reveal that bottleneck adapters offer competitiveness with full fine-tuning with at least 10 hours of data, but they are not as effective in few-shot learning scenarios. Notably, ConvAdapt demonstrates improved performance in such cases. In addition, transfer learning in adapters shows promise, necessitating research in related languages. Furthermore, employing larger speech models for adapter-tuning surpasses fine-tuning with ample data, potentially due to reduced overfitting than fine-tuning.

1 Introduction

Automatic speech recognition (ASR) advancements have favored high-resource languages due to abundant data and computing power. However, over 7000 languages are low-resource or zero-resourced, raising concerns of extinction (Dunbar et al., 2021). Large pre-trained self-supervised speech models like Wav2vec 2.0 show promise in enhancing ASR for low-resource languages through fine-tuning with smaller datasets (Baevski et al., 2020). Fine-tuning such large and even multi-lingual models with a low-resource language data usually works well in practice but has its own limitations. It involves updating most of the model parameters which is inefficient, resource-intensive and storage-demanding. Moreover, it poses challenges in dealing with multiple tasks/languages, causing catastrophic forgetting and complex decision-making for choosing the task sequence (Pfeiffer et al., 2021).

Bottleneck adapters, initially introduced in computer vision (Rebuffi et al., 2017), consist of two-layer feed-forward networks inserted into large pre-trained models (Houlsby et al., 2019). This technique selectively updates adapter parameters while keeping the rest of the model frozen, effectively reducing trainable parameters. It facilitates task-specific adapter integration into pre-trained models, avoiding the need for full re-training and mitigating catastrophic forgetting (Pfeiffer et al., 2021).

While adapters have been well studied in natural language processing (NLP) literature (Houlsby et al., 2019), investigating adapters in the speech signal processing domain is relatively new. Bottleneck adapters are implemented for ASR with the Wav2vec 2.0 English base model (Thomas et al., 2022; Yue et al., 2024), MMS (Pratap et al., 2023), and Google Universal Speech Model (Zhang et al., 2023). These studies indicate that adapters perform on par with fine-tuning while being parameter efficient. Few studies explore bottleneck adapters for specialized tasks like multi-domain ASR modeling with Transformers (Lee et al., 2021), personalized speech recognition in a multi-turn dialog setting with Transducers (RNN-T) (Chang et al., 2023), atypical and accented speech recognition with RNN-T and Transformer Transducers (Tomanek et al., 2021). The latter one utilized residual connections within adapters. Different adapter-based approaches are compared for several speech processing tasks with three of the state-of-the-art pre-trained models (Chen et al., 2023a). The selection of different neural layers to insert the adapters is performed with a two-stage algorithm (Huang et al., 2023). While the prior works rely on bottleneck adapters, CHAPTER technique based on convolutional neural network (CNN) adapters are employed in HuBERT feature extractor on emotion and speaker tasks (Chen et al., 2023b). To the best

Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025), pages 100–107

of our knowledge, no work leveraged convolutional nets as adapters by incorporating them into the contextual Transformer layers in the speech processing domain. Also, given the limited work on ASR modeling using adapters, there exist substantial research gaps that necessitate a comprehensive study of this method. It still remains an open question what the training data size must be for adapter-tuning to perform on par with complete fine-tuning for the low-resource ASR task since the prior work investigates the adapter-tuning for speech processing with only high-resource languages such as English, omitting the suitability of the approach for low-resource languages. Furthermore, the possibility of scaling the adapter modules or pre-training the adapters with a source language are not explored in the literature.

This study aims to address the aforementioned research gaps by conducting a comprehensive investigation of adapters for ASR, with a particular focus on the low-resource aspect. Through this research, we aim to reduce computational complexity while simultaneously maintaining ASR performance, ensuring the representation and preservation of low-resource languages in the field of speech technology. The contribution of this work is four-fold:

- Exploring the adapter-tuning approach for ASR across various resource-constrained scenarios, spanning from low-resource to medium/high-resource conditions in three diverse languages: English (West Germanic), Bengali (Indo-Aryan), and Maltese (Semitic). To this end, we propose a simple yet effective technique ConvAdapt for extreme low-resource parameter-efficient ASR. Notably, no prior research has explored the data requirements for adapter-based low-resource ASR, to the best of our knowledge.

- Leveraging the potential of multilingual, pre-trained self-supervised speech models, we incorporate adapters into state-of-the-art models, namely XLS-R (Babu et al., 2021) and MMS (Pratap et al., 2023). Additionally, we investigate whether employing a larger pre-trained model with a higher number of parameters enhances the performance of the adapter-tuning approach for ASR. Adapter performance for varied sizes of multi-lingual pre-trained models is not studied in the literature, to our knowledge.

- Exploring pre-training adapters on a source language and subsequently fine-tune them for the target language, enabling transfer learning within adapters for the first time.

- While bottleneck adapters with a two-layer feed-forward network are common in adapter architectures (Houlsby et al., 2019; Thomas et al., 2022), this study extends the adapter module by adding more fully connected layers and assesses their influence on performance across the three languages.

2 Integrating Adapters into Wav2vec 2.0

Figure 1 presents the architecture of the adapter-based Wav2vec 2.0 model. The core structure of Wav2vec 2.0 remains unchanged (Baevski et al., 2020), while each Transformer block includes two adapter modules. The process starts with raw input signal passing through a feature encoder, then entering the contextual network (Transformer). Each Transformer block consists of sub-modules like Multi-Head Self-Attention (MHSA) and feed-forward layer. Adapter modules are inserted after the MHSA and the feed-forward layer. There are two residual connections in each Transformer block. The model can contain N transformer blocks, with N being either 24 or 48, depending on the specific Wav2vec 2.0 model. A linear classifier (classification head) is added at the end of the network. During adapter-tuning, only adapter modules, normalization layers, and the head are trained while keeping the pre-trained backbone frozen, substantially reducing trainable parameters.

The bottleneck adapter architecture (FFAdapter), depicted on the upper right side of Figure 1, consists of two fully-connected feed-forward networks. The first layer acts as a down-sampler, projecting the Transformer model dimension to a lower inner dimension through down-projection. A GELU activation is added after that. The second layer functions as an up-sampler, projecting back to the original dimension. Both layers maintain an inner dimension of 256. A residual connection adds the second layer's output with the initial adapter input, processed through layer normalization to yield the final output.

Let the Transformer model representation be d_m and the representation from the second FC layer is f_m. Both d_m and the f_m have the same dimension of m. The output of the Add & Norm layer is

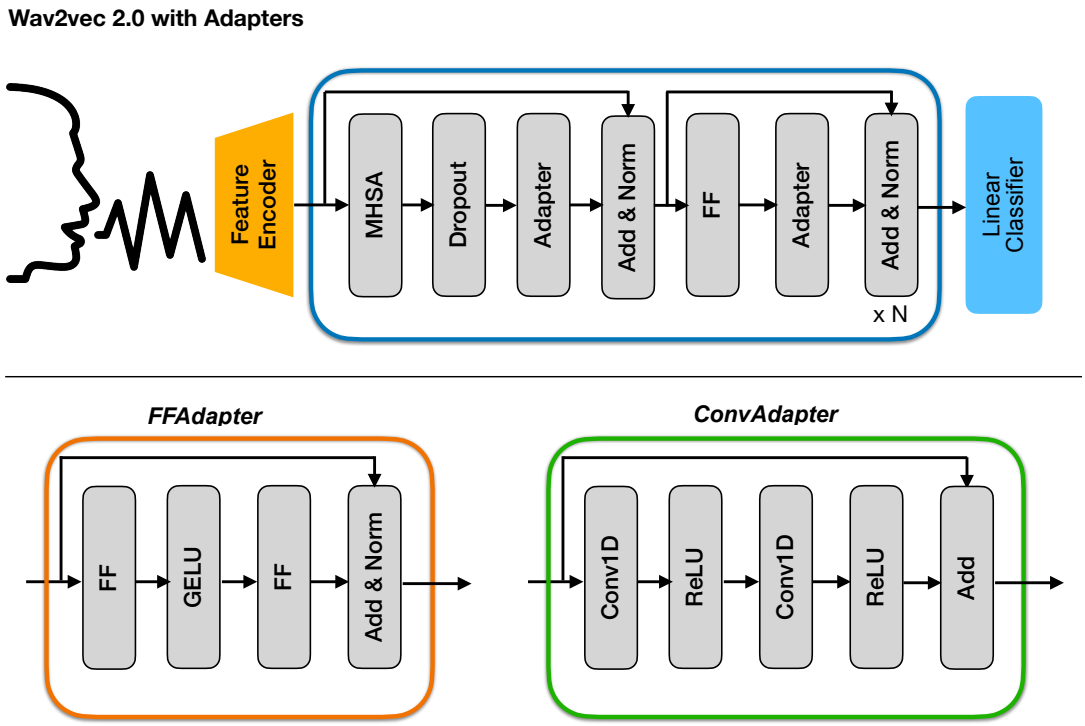

Figure 1: The upper figure depicts the Wav2vec 2.0 architecture with adapter modules. The bottom left figure shows the bottleneck adapter (FFAdapter), while the ConvAdapter is displayed on the bottom right. MHSA and FF represent multi-head self-attention and feed-forward layers, respectively.

computed as,

$$AdapterOutput = LayerNorm(d_m + f_m) \quad (1)$$

We propose the ConvAdapt technique by replacing the bottleneck adapters with CNN-based adapters while keeping them at the same position in the Transformer layers (See Figure 1). The latent representation from the MHSA/FF of the transformer is fed as input to the ConvAdapters after rearranging the tensors to avoid dimension mismatch. We employ two 1-dimensional convolutional layers, each followed by a rectified linear unit (ReLU). There are 1280 input channels and 1280 output channels, and both kernel size and stride are set to 1. The same padding is used. Finally, we rearrange the resultant tensors again to obtain the original dimension and add this to the adapter input through a residual connection.

3 Experiments

Datasets: We conduct the experiments on three languages: Bengali, Maltese, and English. The LibriSpeech corpus is used for English (Panayotov et al., 2015), the SUBAK.KO corpus for Bengali (Kibria et al., 2022), and a combination of datasets including CommonVoice, MASRI-HEADSET, MEP, Tube, MERLIN, and Parliament for Maltese (Ardila et al., 2020; Mena et al., 2020). Various subsets of data are created, ranging from 10 minutes to 50 hours, for analyzing data requirements. Besides the 10 minute, 1 hour and 10 hour subsets from LibriLight (Kahn et al., 2020), we add an additional 50 hour subset from the standard 100-hour English LibriSpeech subset. For Bengali, we create the subsets by randomly selecting samples from the 200-hour SUBAK.KO train set. We follow a similar random sampling approach for Maltese. To ensure standardized benchmarking, we utilize the LibriSpeech development (dev) and test sets, both containing *clean* and *other* subparts. The SUBAK.KO dataset includes standard dev and test sets with 20 hours of data each. Similarly, we utilize the standard Maltese dev and test sets, containing 1.5 hours and 2.3 hours of speech, respectively. The dataset details are summarized in Table 1.

Experimental setup: We choose two large pretrained cross-lingual models, namely XLS-R and MMS (Pratap et al., 2023; Babu et al., 2021). XLS-R has three variants with 0.3 billion (B), 1B, and 2B trainable parameters. We utilize MMS containing

Lang	Language Group	Datasets	train set length	dev set clean	dev set other	test set clean	test set other
BN	Indo-Aryan	SUBAK.KO (Kibria et al., 2022)	200.3	20.5	-	20.3	-
MT	Semitic	Common Voice (Ardila et al., 2020) MASRI-HEADSET (Mena et al., 2020) MEP Tube MERLIN Parlament	52.5	2.3	-	1.5	-
EN	West Germanic	LibriSpeech (Panayotov et al., 2015)	960.9	5.4	5.3	5.4	5.1

Table 1: The datasets are split into train, dev and test sets. BN, MT, and EN refer to Bengali, Maltese, and English, respectively. For English, each of dev and test sets has clean and other (noisy) versions.

1B parameters. Both fully fine-tuned and adapter-based ASR models are trained with a batch size of 4, accumulating gradients for two steps, max 150K steps, early stopping patience for 10K steps, and seed 100. The learning rates of 3e-5, 5e-5, and 5e-4 are used for complete fine-tuning, bottleneck adapter-tuning, and ConvAdapt, respectively. We use greedy search decoding leveraging connectionist temporal classification (CTC) to obtain the output characters (Graves et al., 2006).

Results: The comparison between fine-tuning and adapter-tuning reveals their distinct advantages depending on the dataset size (See Table 2). In extremely low-resource scenarios, like those with just 10 minutes or 1 hour of training data, fine-tuning significantly outperforms bottleneck adapter-tuning (FFAdapter) across languages and model sizes. This situation can be seen as few-shot learning due to the extremely limited labeled speech. However, in moderately low-resource conditions (at least 10 hours), bottleneck adapter-tuning performs competitive to fine-tuning while significantly reducing trainable parameters. We argue that, with less data, the fully connected feed-forward adapters cannot be properly trained and the subsequent modules in Transformer rely upon the output representations from adapter. For this reason, bottleneck adapters are not suitable for few-shot learning. To counter this issue, our proposed technique ConvAdapt is able to outperform bottleneck adapters in extremely low-resource cases while still under-performing than full fine-tuning. We hypothesize that due to sparse connectivity and weight sharing in convolutional nets as opposed to full connectivity in FF nets, ConvAdapter achieves superior performance than bottleneck adapters in few-shot scenarios with less data. As training data increases, however, the benefit of ConvAdapt over bottleneck adapters diminishes because fully connected weights in bottleneck adapters can be learned with sufficient amount of data.

From Table 2, it is evident that the fully fine-tuned XLS-R model with 2B parameters yields comparatively high WERs across different languages and training dataset sizes. The XLS-R 2B performance is consistently surpassed by smaller capacity fully fine-tuned XLS-R models (0.3B, 1B) and the MMS 1B model in all cases. Investigating further, we refer to (Babu et al., 2021), which explores fine-tuned XLS-R models on LibriSpeech. Though the authors argue that higher-capacity models could mitigate interference issue of pre-trained models and yield lower WERs, this remains unverified for the XLS-R 2B model with no results presented for this model.

In our work, we find that employing the bottleneck adapter-tuning approach enables the XLS-R with 2B parameters to achieve the lowest WERs across several dataset sizes except for the extremely low-resource ones e.g. 10 min or 1 hour. This finding is noteworthy since XLS-R 2B with bottleneck adapters not only improves performance but also reduces trainable parameters remarkably from 2B to 64M (almost 31 times reduction). We argue that adapters can function as regularizers in large pre-trained speech models by mitigating overfitting and

Train set size	Model	Adapter Type	# Params in adapters	Maltese		Bengali		English			
				FT	Adapter	FT	Adapter	FT clean	FT other	Adapter clean	Adapter other
10 min	XLS-R 0.3B	FF	26M	63.6	98.9	70.2	93.7	39.3	48.4	100.0	100.0
	XLS-R 1B	FF	64M	70.5	90.3	70.4	89.9	**36.1**	45.7	98.3	100.0
	XLS-R 2B	FF	64M	62.9	93.5	69.9	87.6	39.4	49.0	91.8	96.2
	MMS 1B	FF	64M	**60.5**	89.0	**64.2**	100.0	36.5	**43.4**	100.0	100.0
	XLS-R 1B	Conv	192M	70.5	76.7	70.4	76.5	**36.1**	45.7	43.3	54.2
1 hour	XLS-R 0.3B	FF	26M	**43.1**	65.4	46.2	63.4	16.7	25.5	86.0	92.2
	XLS-R 1B	FF	64M	48.1	63.4	**44.3**	56.9	**15.5**	24.9	38.2	53.3
	XLS-R 2B	FF	64M	43.9	98.2	47.5	66.2	17.9	27.6	24.2	36.3
	MMS 1B	FF	64M	43.5	61.9	**44.3**	58.5	16.1	**23.9**	34.9	49.4
	XLS-R 1B	Conv	192M	48.1	46.3	**44.3**	49.3	**15.5**	24.9	16.8	26.8
10 hours	XLS-R 0.3B	FF	26M	27.8	34.8	20.1	26.9	8.7	17.2	10.7	21.6
	XLS-R 1B	FF	64M	28.6	29.4	19.8	21.0	8.3	17.4	9.3	18.3
	XLS-R 2B	FF	64M	31.1	31.0	20.2	25.7	10.1	20.1	7.6	15.7
	MMS 1B	FF	64M	35.0	36.1	**18.8**	28.4	8.7	16.7	9.2	18.1
	XLS-R 1B	Conv	192M	28.6	**27.7**	19.8	24.2	8.3	17.4	**7.3**	**13.9**
20 hours	XLS-R 0.3B	FF	26M	26.0	28.2	15.2	17.4	7.1	16.3	8.0	19.8
	XLS-R 1B	FF	64M	26.2	26.5	18.6	25.0	7.1	18.2	6.8	16.7
	XLS-R 2B	FF	64M	28.2	**25.6**	**13.7**	15.8	7.4	18.2	**6.0**	14.9
	MMS 1B	FF	64M	26.5	30.2	13.9	18.0	7.5	16.5	7.5	16.2
	XLS-R 1B	Conv	192M	26.2	26.5	18.6	15.4	7.1	18.2	6.2	**14.7**
50 hours	XLS-R 0.3B	FF	26M	24.5	26.2	12.4	14.6	5.8	14.1	6.4	16.2
	XLS-R 1B	FF	64M	**21.1**	24.9	**10.9**	12.9	6.0	16.2	**5.1**	12.7
	XLS-R 2B	FF	64M	24.4	23.9	19.8	11.3	6.4	17.3	5.3	12.9
	MMS 1B	FF	64M	21.5	29.9	12.1	13.2	6.0	14.8	5.5	**12.6**
	XLS-R 1B	Conv	192M	**21.1**	25.2	**10.9**	12.6	6.0	16.2	**5.1**	12.8

Table 2: **Evaluation of full fine-tuning (FT) and adapter-tuning (bottleneck with FF and ConvAdapt approaches) with XLS-R and MMS models for low-resource ASR in terms of WERs (%).** Varied trainable parameters (0.3B, 1B, 2B) in pre-trained ASR models are explored. Maltese, Bengali, and English (LibriSpeech) are chosen, representing diverse language groups. Five training subsets ranging from 10 min to 50 hours are derived from corresponding datasets. Test set results are provided, and CTC-based greedy decoding is employed.

Language	ISO	XLS-R pre-training data
English	en	69493 hours
Maltese	mt	9120 hours
Bengali	bn	100 hours

Table 3: Number of hours of English, Maltese, and Bengali untranscribed speech data used for pre-training XLS-R (Babu et al., 2021).

effectively harnessing their potential.

Table 2 highlights that adapter-tuning provides the most benefits for English ASR, while Bengali ASR with adapters exhibits higher WERs across all dataset sizes. Notably, XLS-R underwent pre-training with 69,493 hours for English, 9,120 hours for Maltese, and only 100 hours for Bengali as shown in Table 3 (Babu et al., 2021). The subpar performance of adapter-tuning in Bengali might be attributed to its insufficient representation in the pre-trained XLS-R model (See Table 1). However, with increased Bengali labeled data (200 hours) for adapter-tuning, performance substantially improves over full fine-tuning (See Table 4).

For a mid-resource case (200 hours and 360 hours of training data), Table 4 illustrates that

Train dataset	Approach	dev set		test set	
		clean	other	clean	other
BN - 200 hrs	fine-tuning	18.8	-	16.3	-
	adapter-tuning	**8.1**	-	**6.9**	-
EN - 360 hrs	fine-tuning	6.4	17.9	5.8	15.7
	adapter-tuning	**3.5**	**9.4**	**3.7**	**9.4**

Table 4: Evaluation of fine-tuning and bottleneck adapter-tuning with XLS-R 2B for moderately large amount of data of 200 hours for Bengali (BN) and 360 hours for English (EN). Results are reported in terms of WERs (%).

Language	Transfer learning in adapters	dev set		test set	
		clean	other	clean	other
BN	No	**11.6**	-	**11.3**	-
	EN $\rightarrow$ BN	14.8	-	13.8	-
	MT $\rightarrow$ BN	14.5	-	13.6	-
MT	No	15.4	-	23.9	-
	BN $\rightarrow$ MT	**14.0**	-	**22.7**	-
	EN $\rightarrow$ MT	15.3	-	24.1	-
EN	No	5.0	12.7	5.3	12.9
	BN $\rightarrow$ EN	4.9	12.8	4.9	12.9
	MT $\rightarrow$ EN	**4.7**	**12.6**	**4.7**	**12.6**

Table 5: Bottleneck adapters in XLS-R 2B are pre-trained with a 50-hour source language dataset, then fine-tuned with an equivalent-sized target language dataset. The classification head dedicated to the source language is removed. WERs (%) on dev and test sets are reported. "Source language" $\rightarrow$ "target language" signifies knowledge transfer.

bottleneck adapter-tuning achieves notably lower WERs than fine-tuning for both Bengali and English, indicating its suitability for developing ASR models with a moderate to large amount of data. We underscore the significance of this finding for the speech processing community.

Pre-training the adapters with a source language shows a slight performance improvement for Maltese and English ASR, yet the Bengali ASR performance deteriorates when adapters are pre-trained with a source language (See Table 5). We hypothesize that initializing adapters with weights from a closely related source language could be advantageous.

The standard bottleneck adapter, widely used in computer vision and NLP, contains two FF layers. We investigate the impact of increasing the number of FF layers in each adapter block (See Figure 2), with an inner dimension of 256 and GELU activation after each FF layer. Our results show that using 6 FF layers in the adapter architecture yields optimal performance across all three languages. It is worth noting that increasing FF layers in adapters elevates the number of trainable parameters, such as from 64M for 2 FF layers to 102M for 6 FF layers, although not significantly.

4 Conclusion and Future Scope

This study presents a comprehensive analysis of parameter-efficient adapters for large pre-trained speech models. We find that bottleneck adapters are not suitable for few-shot learning, however, they perform competitive to full fine-tuning when at least 10 hours of data are available. Our proposed ConvAdapt technique in Transformer layers is simple yet effective to deal with extremely low-resource cases. In mid-to-high resource scenarios, bottleneck adapter-tuning surpasses the widely used full fine-tuning technique, signifying its considerable impact in the field. Leveraging higher-capacity models like XLS-R 2B significantly improves adapter-based tuning, countering the overfitting challenge posed by large pre-trained models during full fine-tuning. Impressively, adapters

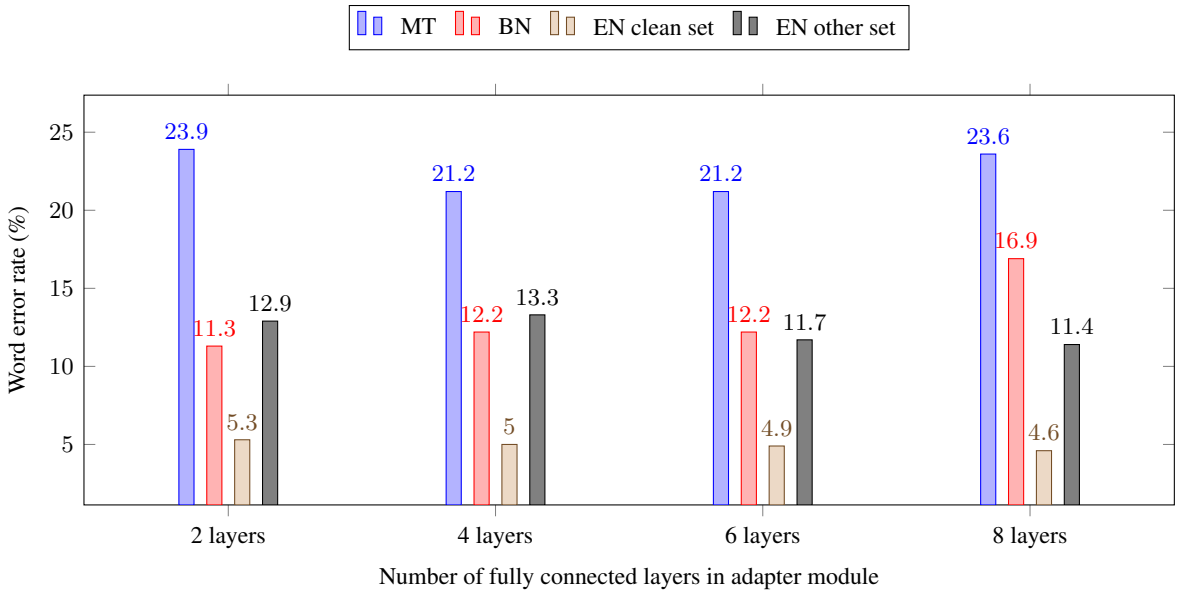

Figure 2: Impact of increasing the number of FC layers in each bottleneck adapter, inserted into XLS-R 2B.

achieve strong performance with merely 2.96% of total trainable parameters. The approach proves better for languages with ample pre-training data. Moreover, transfer learning within adapters benefits Maltese and English, but not Bengali potentially due to the lower amount of Bengali data used in pre-training. Scaling adapters with six feed-forward layers is optimal for all three languages.

We believe that our intriguing findings on adapter-tuning showing remarkable potential for both low-resource and mid/high-resource ASR would encourage more research into this direction. Future work includes exploring transfer learning in adapters with closely-related languages and performing multiple tasks using a single encoder.

5 Limitations

While this work provides novel findings applying adapters for ASR, there exist some limitations. In our experiments with pre-training adapters on the source language and then finetuning on the target language, we use three languages (Bengali, Maltese, and English) that derive from distinct language groups. However, using closely-related language pairs, more performance gain is expected as observed in different studies on transfer learning (Baevski et al., 2020). Due to the limited scope, we restrict this experiment to only the selected three languages in this paper and leave it for future studies.

Acknowledgments

Work supported by the Language and Communication Technologies program of the Erasmus+ project of the European Commission.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al. 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *Interspeech*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Feng-Ju Chang, Thejaswi Muniyappa, Kanthashree Mysore Sathyendra, Kai Wei, Grant P. Strimel, and Ross McGowan. 2023. Dialog act guided contextual adapter for personalized speech recognition. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Zih-Ching Chen, Chin-Lun Fu, Chih-Ying Liu, Shang-Wen Daniel Li, and Hung-yi Lee. 2023a. Exploring

efficient-tuning methods in self-supervised speech models. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 1120–1127. IEEE.

Zih-Ching Chen, Yu-Shun Sung, and Hung-yi Lee. 2023b. Chapter: Exploiting convolutional neural network adapters for self-supervised speech models. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 1–5. IEEE.

Ewan Dunbar, Mathieu Bernard, Nicolas Hamilakis, Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Eugene Kharitonov, and Emmanuel Dupoux. 2021. The zero resource speech challenge 2021: Spoken language modelling. In *Interspeech 2021-Conference of the International Speech Communication Association*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

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.

Junwei Huang, Karthik Ganesan, Soumi Maiti, Young Min Kim, Xuankai Chang, Paul Liang, and Shinji Watanabe. 2023. Findadaptnet: Find and insert adapters by learned layer importance. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. IEEE.

Shafkat Kibria, Ahnaf Mozib Samin, M Humayon Kobir, M Shahidur Rahman, M Reza Selim, and M Zafar Iqbal. 2022. Bangladeshi bangla speech corpus for automatic speech recognition research. *Speech Communication*, 136:84–97.

Taewoo Lee, Min-Joong Lee, Tae Gyoon Kang, Seokyeoung Jung, Minseok Kwon, Yeona Hong, Jungin Lee, Kyoung-Gu Woo, Ho-Gyeong Kim, Jiseung Jeong, et al. 2021. Adaptable multi-domain language model for transformer asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7358–7362. IEEE.

Carlos Daniel Hernandez Mena, Albert Gatt, Andrea DeMarco, Claudia Borg, Lonneke van der Plas, Amanda Muscat, and Ian Padovani. 2020. Masri-headset: A maltese corpus for speech recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6381–6388.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *16th Conference of the European Chapter of the Associationfor Computational Linguistics, EACL 2021*, pages 487–503. Association for Computational Linguistics (ACL).

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2023. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30.

Bethan Thomas, Samuel Kessler, and Salah Karout. 2022. Efficient adapter transfer of self-supervised speech models for automatic speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7102–7106. IEEE.

Katrin Tomanek, Vicky Zayats, Dirk Padfield, Kara Vaillancourt, and Fadi Biadsy. 2021. Residual adapters for parameter-efficient asr adaptation to atypical and accented speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6751–6760.

Xianghu Yue, Xiaoxue Gao, Xinyuan Qian, and Haizhou Li. 2024. Adapting pre-trained self-supervised learning model for speech recognition with light-weight adapters. *Electronics*, 13(1):190.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*.