Low-resource Language Adaptation with Ensemble of PEFT Approaches

Chin Yuen Kwok*, Sheng Li†, Jia Qi Yip‡* and Eng Siong Chng*
* Nanyang Technological University, Singapore
† National Institute of Information & Communications Technology (NICT), Japan
‡ Alibaba Group, Singapore
E-mail: kwok0062@e.ntu.edu.sg

***Abstract*—Despite recent advances in automatic speech recognition (ASR) performance on common languages, a large fraction of the world's languages remain unsupported. Parameter-efficient fine-tuning (PEFT) methods are used to adapt these models to unseen languages by inserting language-specific modules into the models. To further improve adaptation performance, an ensemble of PEFT models can be formed, where the outputs of the ensemble can be aggregated to create the final prediction, and it has been shown that increasing the diversity of outputs from the ensemble produce can improve results. However, PEFT model ensembles have rarely been studied in the context of ASR despite its advantage of requiring significantly less memory for model storage, and the effect of using diverse PEFT methods to create diversity in PEFT ensemble model outputs remains unexplored. Specifically, it is unclear whether training with different PEFT methods improves diversity more than using the same PEFT method with different random seeds. To verify this, we examine whether a better model ensemble can be formed by combining models adapted by different PEFT methods instead of the same PEFT method. When adapting Whisper to 10 hours of data for each of the 3 unseen languages from Common Voice, results show that our ensemble with diverse PEFT methods consistently outperforms those that use the same PEFT method. Moreover, compared to the common approach of using fully fine-tuned models to form ensembles, our diverse PEFT ensemble can reduce the Word Error Rate from 8.4% to 7.9% while requiring about five times less memory for model storage.**

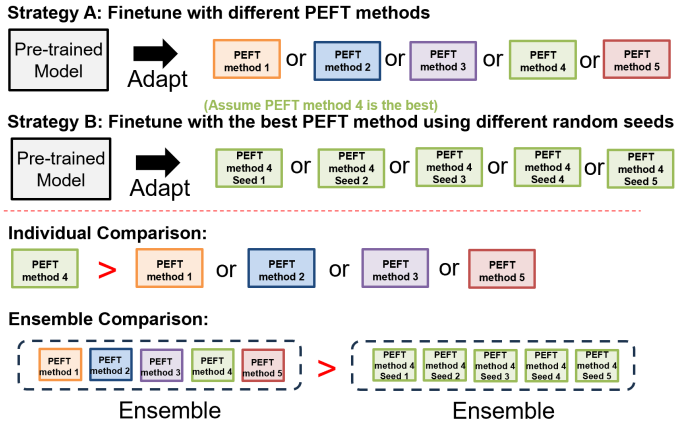

Fig. 1: *Overview of our PEFT model ensemble. Top: A pre-trained model is adapted using different PEFT methods. Bottom: Assume that PEFT method 4 is better than other PEFT methods when the models are used individually. However, an ensemble of models adapted by different PEFT methods produces better results than an ensemble of models adapted by the best PEFT method 4.*

I. Introduction

Automatic Speech Recognition (ASR) is the task of obtaining text transcriptions from a given segment of speech. Due to the availability of training data ASR models primarily only support commonly spoken languages such as English. This poses a significant challenge, as a large portion of the world's linguistic diversity remains unsupported by ASR models. To bridge this gap, researchers have explored techniques for adapting existing models to new languages. However, a major hurdle lies in the fact that most of these new languages are considered low-resource, meaning there's a scarcity of training data available for fine-tuning. This limited data makes it difficult to achieve optimal adaptation performance for these languages.

To efficiently adapt the ASR models to these low-resource languages, parameter-efficient fine-tuning (PEFT) methods [1] have been studied to add language-specific modules to the models, and multiple module designs have been previously explored. These include Sparse Subnetworks [2], which use modules in the form of a binary mask to selectively keep or remove each connection in a model; Low-Rank Modules [3], which are in the form of two low-rank matrices, and are applied to certain groups of model parameters like the linear projection; Prompting [4], which use modules in the form of a learnable vector, and the vector is merged with the model input; and Adapters [5], which use modules in the form of a small neural network, and the network is composed with selected model layers to modify the layers' outputs.

As different PEFT methods have very different designs for the modules they use, previous works [6]–[8] have compared their effectiveness in the audio domain. Among these, a few works have studied PEFT for low-resource ASR. They include [9], which adds adapters and learnable bias into the ASR model, [10], which learns to fuse multiple sets of weights to create binary masks, and [11], which learns to perform fusion on multiple adapters.

To further improve adaptation performance, a PEFT model ensemble can be formed. Compared with the common approach of using fully fine-tuned models to form ensembles

[12]–[14], PEFT model ensembles has the extra benefit of requiring less memory for model storage as most of the model weights are identical among models. However, to the best of our knowledge, PEFT ensembles are rarely studied in the context of ASR and the interaction between the independently useful PEFT models within a model ensemble remains unexplored. Specifically, while previous studies have shown that diverse outputs can improve the performance of model ensembles [15], it is unclear whether training with different PEFT methods improves diversity more than using the same PEFT method with different random seeds. To verify this, we examine whether a better model ensemble can be formed by combining models adapted by diverse PEFT methods instead of the same PEFT method as shown in the bottom part of Figure 1.

Our contributions are four-fold. We show that: 1) An ensemble of models adapted by diverse PEFT methods consistently outperforms ensembles of models adapted by the same PEFT method. 2) When a model ensemble includes two models adapted by the same PEFT method, replacing one with another model adapted by a different PEFT method improves accuracy, even when the other model individually performs worse. 3) Our diverse PEFT ensemble can reduce the Word Error Rate from 8.4% to 7.9% while requiring about five times less memory compared with the common approach of using fully fine-tuned models to form ensembles. 4) PiggyBack (PB) [2] is the overall best PEFT method for low-resource language adaptation and only performs slightly worse than full fine-tuning.

II. Methodology

The following sections discuss five PEFT methods for adapting an ASR model to unseen low-resource languages. The PEFT methods include PB, SA, PA, LoRA, and LGate as shown in Figure 2. In addition, we propose an ensemble method to combine multiple models adapted by different PEFT methods as shown in Figure 1 to improve the adapted language performance.

A. Whisper Model for multilingual ASR

We use Whisper [16] as the base model as it has shown strong multilingual ASR (MASR) capability. The left part of Figure 2 shows an overview of its architecture. Whisper is a speech model trained with half a million hours of multilingual ASR and translation data. Its largest variant consists of 1.5 billion parameters. It supports MASR of 75 languages and also supports language identification (LID). It uses the Transformer [17] attention-based encoder-decoder architecture and decodes in an auto-regressive manner. Whisper is perferred for low-resource language adaptation because while [18] shows that Whisper performs similarly with XLS-R [19] in terms of WER averaged across seen and unseen languages, [20] shows that Whisper surpasses WavLM [21] overall in MASR adaptation and [22] shows that Whisper performs better than wav2vec 2.0 based models [23] and WavLM [21] for LID. In addition, Whisper decoder is preferred over the Large Language Model (LLM) based decoder [24] as it is found that a larger language model size does not benefit low resource languages much [25].

B. PiggyBack (PB)

To adapt an ASR model to new languages, one solution is to learn a Sparse Subnetwork [26]. As shown in Figure 2A, PB [2] learns a language-specific binary mask and derives a language-specific set of model parameters by taking a copy of the original model parameters and zeroing out part of it. More specifically, a trainable real-valued mask is first initialized to a value γ, and it is binarized using a threshold function:

$$y = \begin{cases} 1, & \text{if } x \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where τ is a hyper-parameter and affects the sparsity of the binary mask. The binary mask is then applied to the model to zero out partial model parameters. The gradients obtained through backpropagation of the training loss are used to directly update the real-valued mask weights.

C. Adapter

As an alternative to leaning a binary mask, as shown in Figure 2B, [27] learns a sequential adapter (SA) and adds it after the multi-head attention (MHA) or the feedforward network (FFN) module of a pre-trained network to modify the layer outputs. Specifically, SA consists of a feedforward (FF) layer to down-project the layer outputs into bottleneck features, a ReLU [28] activation function, and another FF layer to up-project the bottleneck features back to the original dimension. Similarly, Figure 2C shows parallel adapter (PA) [5], which adds the adapter in parallel to the MHA or FFN instead of inserting sequentially after them and behaves similarly to prefix-tuning [4], except that prefix-tuning uses Softmax instead of ReLU [28] as the non-linear activation function.

D. Low-Rank Adaptation (LoRA)

LoRA [3] is another approach to learning language-specific modules by adjusting the weight matrices of the FF layers in a model, and its structure is similar to PA. As shown in Figure 2D, LoRA learns a down and up projection layer like PA. Unlike PA, LoRA is placed in parallel to the FF layers in the model instead of the MHA and FFN. Also, LoRA does not have a non-linear activation function between the projection layers.

The forward pass of a FF layer modified with LoRA yields:

$$h = W_0 x + \Delta W x = W_0 x + W_{up} W_{down} x \tag{2}$$

where random Gaussian is used for initializing W_{down} and zero for W_{up}, so $\Delta W = W_{up} W_{down}$ is zero at the beginning of training.

To further adapt the FF layers, as shown in Figure 2E, [29] adds a gating mechanism before and after each FF layer to modulate their input and output. Specifically, each layer's input and output is multiplied with a separate learnable vector of the same dimension. The approach is inspired by previous work [30], which uses the language identity of the input

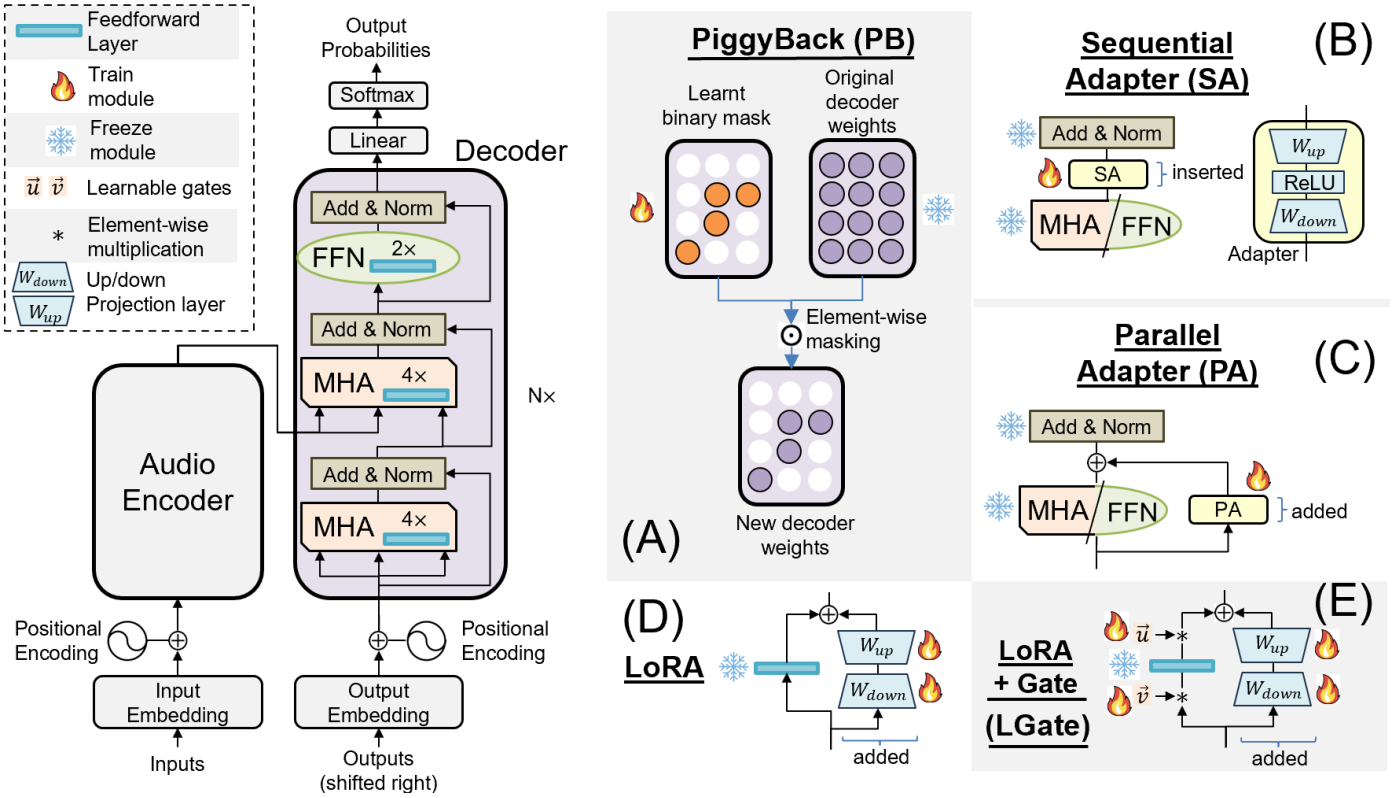

Fig. 2: *Overview of the attention-based encoder-decoder model architecture for ASR (left) and different PEFT methods (right).*

audio to select language-specific gates to modulate internal representations of a multilingual ASR model; and [31], [32], which learns a language code to gate the activity of neurons in an acoustic model. We refer to this method as "LGate".

E. Model Ensemble

To further enhance the adapted language performance, we study the strategy to perform model ensemble, which is a method to combine multiple models, such that the combined system can perform better than any of the individual models. Specifically, we study the strategy of combining multiple models adapted by different PEFT methods instead of models adapted by the same PEFT method to form a model ensemble, as shown in the bottom part of figure 1. As for the model ensemble method, we use ROVER [33]. It is a post-processing system that combines multiple ASR outputs, which uses a "voting" process to reconcile differences in ASR system outputs to obtain a composite ASR output with a lower error rate.

Our method is different from previous works [10], [11] in that we combine models adapted by different PEFT methods, instead of models adapted by the same PEFT method, to form model ensembles. In addition, compared with other approaches which combine models adapted by full fine-tuning to form model ensembles, our approach requires significantly less memory as we only need to store the extra language-specific PEFT modules for each model in the ensemble.

III. Experiments

A. Dataset and Model Details

We implement our methods based on the popular SpeechBrain [34] toolkit and CL-MASR [20].

Following previous works [10], [20], we evaluate our method using a subset of the widely used large-scale CommonVoice dataset[1] [35]. We follow CL-MASR [20] to extract the data subsets for the unseen langauges Esperanto (eo), Interlingua (ia), and Frisian (fy). Each language contains 10 hours of data for training, 1 for validation, and 1 for testing.

For each unseen language, we adapt whisper-small and whisper-large-v2 for two epochs with a train batch size of 4. Specifically, we freeze the Whisper encoder and only adapt the decoder to unseen languages. PEFT methods are applied to add language-specific modules to the decoder, and only the parameters of the modules and the text embedding layer at the decoder's side are updated. We use AdamW [36] as the optimizer and a variant[2] of the ReduceLROnPlateau[3] learning rate (LR) scheduler. Validation is done at an interval of $1/32$ epoch. We sweep through the hyper-parameters to tune them for all methods.

[1]https://commonvoice.mozilla.org/en

[2]https://speechbrain.readthedocs.io/en/latest/_modules/speechbrain/nnet/schedulers.html\#NewBobScheduler

[3]https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

B. Results and Discussion

We show both the results of adapting whisper-small and whisper-large-v2 to new languages Esperanto, Interlingua, and Frisian as shown in Table I. To expand Whisper's language support, a naive solution is to duplicate the model, fully fine-tune (FT) it on new languages, and keep the original copy for old languages. However, this requires double the memory to store both models. To address this, PEFT methods are used to reduce the memory requirement, although the target language WER is compromised as a trade-off. Among the PEFT baseline methods, PB shows the lowest Average Word Error Rate (AWER) on new languages, where it achieves an AWER of 20.9% on whisper-small and 13.7% on whisper-large-v2, which is competent with the result of FT. In addition, our PEFT baseline methods requires more memory than usual as the text embedding layer weights are also stored. We will show later in Section III-C that adapting the embedding layer is necessary to improve the adaptation results for new languages.

For the model ensemble baseline results, three models are adapted by FT using different random seeds to form a model ensemble (ROVER-3a), and it slightly reduces the AWER from 13.7% to 13.4% for whisper-large-v2 compared with single-model FT. Similarly, three models are adapted by PB (ROVER-3b) and it slightly reduces the AWER from 13.7% to 13.5% for whisper-large-v2 compared with single-model PB. The results suggested that combining models that are adapted by the same method to form ensembles only slightly improves the WER. We further test with increasing the model ensemble size from three to five (ROVER-5a), and although AWER is reduced for whisper-small, the AWER for whisper-large-v2 is unaffected. This shows that the improvement made by adding more models to an ensemble may saturate at some point.

Lastly, we adapt 3 models using PB, PA and LoRA respectively to form a model ensemble (ROVER-3M) and adapt 5 models using PB, PA, LORA, SA and LGate to form an ensemble (ROVER-5M). For whisper-small, results show that our method performs worse than FT. We hypothesize this is because applying PEFT methods to smaller models is less effective, as shown by the larger AWER gap between FT and the PEFT baseline methods. For whisper-large-v2, results show that we can reduce AWER from 13.4% to 13.2% if we use 3 models adapted by diverse PEFT methods, instead of 3 models adapted by PB, to form the model ensemble. The AWER can be further reduced from 13.2% to 12.9% as more models adapted by other PEFT methods are added to the ensemble. Finally, our method ROVER-5M reduces WER from 13.4% to 12.9% and outperforms the baseline method ROVER-5a while requiring 5 times less memory.

C. Ablation study

We further perform an ablation study on fine-tuning the text embedding layer at the decoder as shown in Table II. Results show that WER is increased for both whisper-small and whisper-large-v2 if the embedding layer is frozen during FT. This shows that adapting the embedding layer is necessary for low-resource language adaptation.

In addition, we perform an ablation study on combining different models to form model ensembles and show the results in Table III. First, we observe that the performance of a model ensemble heavily depends on the best single-performing model inside the ensemble. This is shown by the results in row 1, which is obtained by combining three models adapted by PA, SA, and LoRA. This model ensemble performs the worst among all other ensembles because it does not include models adapted by PB, which is the best single-performing model as shown in Table I, while the other ensembles do.

Second, results in rows 2-3 show that combining the models adapted by PB, PA, and SA or the models adapted by PB, LoRA, and LGate to form an ensemble have worse performance than other ensembles that have PB-adapted models included. We hypothesize that this is because the models used in rows 2-3 are adapted by less diverse PEFT methods. Specifically, there is less diversity as row 2 uses PA and SA, and row 3 uses LoRA and LGate. These PEFT method pairs are similar in design as shown in Figure 2, so the models adapted by these methods may acquire similar knowledge that complements less with each other. *The results suggested that more diverse PEFT methods are needed to form model ensembles with better accuracy performance.*

The hypothesis that using models with less diverse PEFT methods will lead to inferior results is further supported by the results in row 4 and 7. For row 4, a model ensemble is formed by combining two models adapted by PB and one model adapted by PA. After replacing one of the PB-adapted model to a LoRA-adapted model as shown in row 7, the AWER is reduced from 13.5% to 13.2%. However, LoRA has been shown to perform worse than PB and PA as shown in Table I. *This shows that a model ensemble can benefit from the diversity of PEFT methods even when one of the models in the ensemble is replaced by another model with worse performance.*

D. Increasing model size

We emphasize that simply increasing the model size for adaptation does not guarantee performance improvement. For example, PB has the parameter size fixed by definition, and increasing the bottleneck dimensions of the sequential adapters has diminishing effects on performance improvement [27]. Our paper studies how to obtain SOTA results beyond this limit using speech foundation models, and our diverse PEFT ensemble can reduce the Word Error Rate from 8.4% to 7.9% while requiring five times less memory than the common approach of using fully fine-tuned models to form ensembles.

IV. Conclusion

To conclude, we propose a method to improve low-resource language adaptation performance by combining models adapted by different PEFT methods to form a model ensemble, and ablation study has shown that the performance of a model ensemble can be consistently improved if more diverse PEFT methods are used.

TABLE I: *WER (%) of adapting whisper-small and whisper-large-v2 to three unseen languages: Esperanto (eo), Interlingua (ia), and Frisian (fy). The "None" method refers to not performing adaptation, so Whisper is not trained to transcribe unseen languages.*

Method	**whisper-small**					**whisper-large-v2**				
	Mem	**eo**	**ia**	**fy**	**avg**	Mem	**eo**	**ia**	**fy**	**avg**
FT	2x	17.6	12.3	29.0	19.6	2x	12.2	8.4	20.6	13.7
PEFT baselines										
PB [2]	1.18x	18.9	14.5	29.2	20.9	1.06x	12.0	8.4	20.8	13.7
SA [27]	1.18x	21.6	17.5	34.2	24.5	1.05x	13.9	9.4	23.3	15.5
PA [5]	1.18x	20.8	16.3	33.3	23.5	1.05x	13.3	9.4	22.4	15.0
LoRA [3]	1.18x	21.8	18.8	33.1	24.6	1.05x	14.2	12.6	23.0	16.6
LGate [29]	1.18x	20.9	18.5	31.8	23.7	1.05x	15.0	13.0	21.9	16.6
ensemble baselines										
ROVER-5a	6x	**17.3**	**11.8**	**27.9**	**19.0**	6x	11.7	8.4	20.0	13.4
ROVER-3a	4x	17.7	12.3	28.5	19.5	4x	11.8	8.2	20.1	13.4
ROVER-3b	1.54x	18.2	13.5	29.4	20.4	1.18x	11.7	8.3	20.5	13.5
our PEFT + ensemble methods										
ROVER-3M	1.54x	18.3	13.5	29.8	20.5	1.16x	11.3	8.0	20.3	13.2
ROVER-5M	1.90x	17.9	13.3	28.9	20.0	1.26x	**11.3**	**7.9**	**19.4**	**12.9**

TABLE II: *Ablation study of fine-tuning the text embedding layer at the decoder.*

Method	**eo WER (%)**
FT (*whisper-small*)	17.63
w/o fine-tune embedding layer	18.79
FT (*whisper-large-v2*)	12.15
w/o fine-tune embedding layer	13.43

TABLE III: *Ablation study of combining different models adapted by FT, PB, PA, SA, LoRA, or LGate to form a model ensemble. The last row shows the result of combining two models adapted by FT and one model adapted by PB to form an ensemble.*

Models for Ensemble						**eo,ia,fy**
FT	PB	PA	SA	LoRA	LGate	**AWER (%)**
		✓	✓	✓		14.3
	✓	✓	✓			13.9
	✓			✓	✓	13.8
	2	✓				13.5
	✓	✓			✓	13.5
	✓		✓	✓		13.4
	✓	✓		✓		13.2
2	✓					13.2

REFERENCES

[1] N. Ding, Y. Qin, G. Yang, *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, pp. 220–235, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257316425.

[2] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 67–82.

[3] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[4] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[5] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," *arXiv preprint arXiv:2110.04366*, 2021.

[6] N. M. Selvaraj, X. Guo, A. Kong, B. Shen, and A. Kot, "Adapter incremental continual learning of efficient audio spectrogram transformers," *arXiv preprint arXiv:2302.14314*, 2023.

[7] L.-J. Yang, C.-H. H. Yang, and J.-T. Chien, "Parameter-efficient learning for text-to-speech accent adaptation," *arXiv preprint arXiv:2305.11320*, 2023.

[8] S. Radhakrishnan, C.-H. H. Yang, S. A. Khan, N. A. Kiani, D. Gomez-Cabrero, and J. N. Tegner, "A parameter-efficient learning approach to arabic dialect identification with pre-trained general-purpose speech model," *arXiv preprint arXiv:2305.11244*, 2023.

[9] D. Ng, C. Zhang, R. Zhang, *et al.*, "Adapter-tuning with effective token-dependent representation shift for automatic speech recognition,"

[10] Z. Yu, Y. Zhang, K. Qian, *et al.*, "Master-asr: Achieving multilingual scalability and low-resource adaptation in asr with modular learning," in *International Conference on Machine Learning*, PMLR, 2023, pp. 40 475–40 487.

[11] J. Qi *et al.*, "Parameter-efficient dysarthric speech recognition using adapter fusion and householder transformation," *arXiv preprint arXiv:2306.07090*, 2023.

[12] A. Arunkumar, V. N. Sukhadia, and S. Umesh, "Investigation of ensemble features of self-supervised pretrained models for automatic speech recognition," *arXiv preprint arXiv:2206.05518*, 2022.

[13] A. K. Parikh, L. ten Bosch, and H. van den Heuvel, "Ensembles of hybrid and end-to-end speech recognition.," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds., Torino, Italia: ELRA and ICCL, May 2024, pp. 6199–6205. [Online]. Available: https://aclanthology.org/2024.lrec-main.547.

[14] I. Gitman, V. Lavrukhin, A. Laptev, and B. Ginsburg, "Confidence-based ensembles of end-to-end speech recognition models," *arXiv preprint arXiv:2306.15824*, 2023.

[15] A. K. Parikh, L. ten Bosch, and H. van den Heuvel, "Ensembles of hybrid and end-to-end speech recognition.," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 6199–6205.

[16] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*, PMLR, 2023, pp. 28 492–28 518.

[17] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] A. Rouditchenko, S. Khurana, S. Thomas, *et al.*, "Comparison of multilingual self-supervised and weakly-supervised speech pre-training for adaptation to unseen languages," *arXiv preprint arXiv:2305.12606*, 2023.

[19] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[20] L. Della Libera, P. Mousavi, S. Zaiem, C. Subakan, and M. Ravanelli, "Cl-masr: A continual learning benchmark for multilingual asr," *arXiv preprint arXiv:2310.16931*, 2023.

[21] S. Chen, C. Wang, Z. Chen, *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[22] K. Praveen, B. Radhakrishnan, K. Sabu, A. Pandey, and M. Shaik, "Language identification networks for multilingual everyday recordings," 2023.

[23] A. Babu, C. Wang, A. Tjandra, *et al.*, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," *arXiv preprint arXiv:2111.09296*, 2021.

[24] C. Tang, W. Yu, G. Sun, *et al.*, "Salmonn: Towards generic hearing abilities for large language models," *arXiv preprint arXiv:2310.13289*, 2023.

[25] Z. Liu, J. Spence, and E. Prud'Hommeaux, "Studying the impact of language model size for low-resource asr," in *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, 2023, pp. 77–83.

[26] J. Pfeiffer, S. Ruder, I. Vulić, and E. M. Ponti, "Modular deep learning," *arXiv preprint arXiv:2302.11529*, 2023.

[27] N. Houlsby, A. Giurgiu, S. Jastrzebski, *et al.*, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*, PMLR, 2019, pp. 2790–2799.

[28] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[29] N.-Q. Pham, T.-N. Nguyen, S. Stüker, and A. Waibel, "Efficient weight factorization for multilingual speech recognition," *arXiv preprint arXiv:2105.03010*, 2021.

[30] S. Kim and M. L. Seltzer, "Towards language-universal end-to-end speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 4914–4918.

[31] M. Müller, S. Stüker, and A. Waibel, "Neural language codes for multilingual acoustic models," *arXiv preprint arXiv:1807.01956*, 2018.

[32] M. Müller, S. Stüker, and A. Waibel, "Neural codes to factor language in multilingual speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 8638–8642.

[33] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, IEEE, 1997, pp. 347–354.

[34] M. Ravanelli, T. Parcollet, P. Plantinga, *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[35] R. Ardila, M. Branson, K. Davis, *et al.*, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[36] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.