INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Re-investigating the Efficient Transfer Learning of Speech Foundation Model using Feature Fusion Methods

Zhouyuan Huo, Khe Chai Sim, Dongseong Hwang, Tsendsuren Munkhdalai
Tara Sainath, Pedro Moreno

Google LLC, USA
{zhhuo,khechai}@google.com

Abstract

Speech foundation models, pre-trained on large amounts of unsupervised or supervised audio data, have demonstrated an impressive ability to transfer their learning to specific domains for speech recognition. Parameter-efficient fine-tuning methods offer an efficient paradigm where a small set of parameters are updated to adapt the foundation model to new tasks. However, it is unclear how the intermediate features of the foundation model behave, and how to utilize them in a more efficient way. In this paper, we compare the performance of three speech foundation models for speech recognition. We re-investigate how features from different layers behave and propose a simple and effective feature fusion method for efficient transfer learning. Experimental results demonstrate that the proposed method uses 31.7% fewer trainable encoder parameters, 13.4% less computational memory cost than compared method, and does not compromise quality on the target task.

Index Terms: Speech Recognition, Foundation Model, Transfer Learning

1. Introduction

Foundation models [1], also known as large language models (LLMs), are trained on huge amounts of text or code and can be fine-tuned for a wide range of tasks. They have shown impressive results in natural language processing tasks [2, 3, 4, 5]. In the speech community, self-supervised pre-training of foundation models on large amounts of unlabeled speech has shown promise for improving speech recognition [6, 7]. There are two main categories of self-supervised learning algorithms for speech foundation models: 1) reconstruction-based methods that predict the input feature directly, such as Auto-regressive Predictive Coding [8] and Multi-Predictive Coding [9]; and 2) BERT-style methods that bridge the gap between continuous speech signal and discrete text tokens, such as Wav2vec2.0 [10], HuBERT [11], w2v-BERT [12] and BEST-RQ [13]. After pre-training the speech foundation model, the foundation model can be fine-tuned on the supervised data for the downstream tasks. For example, in the case of speech recognition, the encoder is initialized from a pre-trained foundation model and fine-tuned on the supervised data of the target domain. After the model is fine-tuned, it can be used to recognize speech in the target domain.

A large general-purpose foundation model can be adapted to many downstream tasks, but it is challenging to adapt it to many tasks efficiently with only a small amount of supervised data for each task. Existing works have investigated ways to reduce the number of trainable parameters required for fine-tuning the foundation model. For example, BitFit [14] proposed a sparse-finetuning method where only the bias terms of the model are updated. Houlsby et al. [15] proposed to insert adapter modules between the layers in a frozen pre-trained model. Each adapter module is a small trainable feed-forward neural network. [16, 17] further reduced the number of parameters by exploiting low-rank matrix approximation. These parameter-efficient methods achieve decent performance on the downstream task with a significant reduction in the trainable parameters. However, they still require a lot of computational resources for fine-tuning. It is because that these methods add/update sparse parameters in the intermediate layers of the foundation model, which requires a full backpropagation process from the top to the bottom of the network to compute the gradients of the trainable parameters. Besides, they use the output of the highest layer in the foundation model and do not leverage intermediate features for downstream tasks. Hierarchical Feature Fusion (HFF) [18] proposed a resource-efficient transfer learning method by treating the foundation model as a frozen feature extractor and fused features from multiple intermediate layers of the foundation model using a feature projector. Motivated by the observation that the middle layers encode high-level information while the bottom or top layers encode low-level information, HFF projects and concatenates features from adjacent layers hierarchically. Results in [18] showed that after combining with Adapters [15] at all layers, the HFF can achieve the same performance as fine-tuning the whole model with much fewer trainable encoder parameters and much faster training speed. However, it is redundant to use all feature layers and a manually designed hierarchical structure does not generalize to other tasks.

Pasad et al. [19] analyzed layer-wise features from a speech representation model pre-trained using wav2vec2.0 algorithm [10] and found that the middle layers encode contextual and high-level information while the bottom or top few layers encode lower-level information and local representations. Arunkumar et al. [20] investigated how multiple self-supervised pre-trained models can be used together in ASR. They found that the features from different models are complementary, and that combining them can improve the performance of ASR tasks. Huo et al. [18] found that fusing features from multiple layers of a speech foundation model can benefit the transfer learning. However, fusing the redundant low-level features from both bottom or top layers could hurt adapted model performance and comsuming unnecessary computational resources.

In this paper, we re-investigate the efficient transfer learning of speech foundation model using feature fusion methods. We perform speech recognition transfer learning tasks from three foundation models with different qualities and compare the Word Error Rate (WER) of the adapted model on the target domain. We list our main contributions as following: Firstly, we show that the quality of foundation models plays a more im-

10.21437/Interspeech.2023-2296

portant role for parameter-efficient methods than fine-tuning all parameters. Secondly, we find that dropping top 4 layers of the speech foundation model (Drop4) has no effect on the adapted model and can reduce the computational memory cost significantly. Thirdly, we propose a novel Global Attentional Feature Fusion (GAFF) method which outperforms the compared methods in terms of parameter and training efficiency.

2. Methods

This study focuses on efficient transfer learning of speech foundation models using feature fusion methods. We investigate how foundation model quality affects transfer learning to downstream tasks using parameter-efficient methods (e.g. Adapter and HFF), and propose a novel Global Attentional Feature Fusion (GAFF) method to replace the manually designed hierarchical projector in HFF.

2.1. Speech Foundation Model

We follow the foundation model architecture used in [18, 21], which is a 2-layer convolutional network followed by a 24-layer conformer encoder [22] with hidden dimension 1024 and contains 600M parameters in total. Each conformer layer is a convolution-augmented transformer network, which consists of attention, feed-forward and convolutional modules. The model input is a vector of size 128 logMel features and SpecAugment [23] is also applied to increase model robustness. In the paper, we utilize and adapt three different foundation models: 1) pre-trained on Libri-Light data using W2v-BERT algorithm [12]; 2) pre-trained on unsupervised YouTube data using Best-RQ algorithm [13]; and 3) fine-tune the pre-trained foundation model from 2) using CTC loss [24] on supervised YouTube data. Details of the training data are described in Section 3.1.

2.2. Global Attentional Feature Fusion

To replace the manually designed hierarchical projection in HFF, we propose a global attentional feature fusion method (GAFF) to learn weight of each layer automatically as in Figure 1. After stacking features from n intermediate layers, we get a tensor $X_1 \in \mathcal{R}^{T\times n\times d}$, where T is the sequence length and d is the model dimension. Next, we compress the global sequence information and feature information from X_1 using the average (AVG) operator and squeeze function F_{sq},

$$X_3 := F_{sq}(W, X_2) = \delta(X_2 W) \qquad (1)$$

where δ is the SWISH activation function [25], $W \in \mathcal{R}^{d\times 1}$ and $X_2 = AVG(X_1) \in \mathcal{R}^{1\times n\times d}$ along the sequence dimension T. $X_3 \in \mathcal{R}^{1\times n\times 1}$ is then fed into the excitation function F_{ex} to fully capture the layer-wise dependencies:

$$X_4 := F_{ex}(W_1, W_2, X_3) = \sigma(\delta(X_3 W_1) W_2) \qquad (2)$$

where σ denotes the sigmoid activation, $W_1 \in \mathcal{R}^{n\times \frac{n}{s}}$, and $W_2 \in \mathcal{R}^{\frac{n}{s}\times n}$. s is the scaling factor to fuse the layer-wise information, and we set $s = 2$ across our experiments. Finally, we obtain the re-weighted tensor:

$$X_5 := F_{sc}(X_4, X_1) = X_4 \circ X_1 \qquad (3)$$

after computing the hadamard product of $X_4 \in \mathcal{R}^{1\times n\times 1}$ and $X_1 \in \mathcal{R}^{T\times n\times d}$. Similar to [18], the *Concat&Project* module in Figure 1 concatenates the layer-wise and feature dimensions, and projects it to the model dimension d using a three-layer feed-forward network with hidden dimension d. The GAFF is motivated by the Squeeze-and-Excitation Networks [26], which adaptively recalibrates the channel-wise features by explicitly modelling the inter-channel dependencies for image classification task. The main difference lies in how to squeeze the information at the sequence and feature levels.

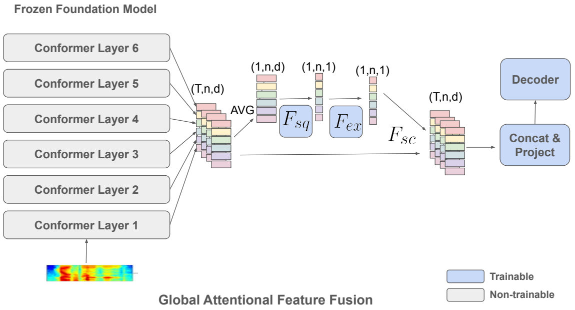

Figure 1: *Global Attentional Feature Fusion.*

3. Experimental Setup

In the paper, we re-investigate the transfer learning capability of the speech foundation model by adapting three foundation models to the target domain using parameter-efficient fine-tuning methods and comparing their speech recognition performance.

3.1. Training Data

There are three different datasets used for pre-training the foundation models as the source domain. **Libri-Light** [27] contains about 60K hours of unannotated speech audio. The other two datasets are collected from YouTube. The unsupervised YouTube data, namely **YT-U**, is a multilingual YouTube dataset segmented using voice activity detection models [28]. This set brings a more diverse speech variations for the foundation model than Libri-Light. The supervised YouTube data, namely **YT-T**, is an English only dataset from videos that have user-uploaded transcripts. These videos are first segmented using a 100M-parameter RNN-T model with a bi-directional LSTM encoder [29]. The non-speech segments identified by the YT teacher model are removed to yield approximately 500K hours of unlabeled audio data. The user provided transcripts, however, are discarded and we generate pseudo-labels using the same YT teacher model. In addition, the target domain Common Voice data contains 1K hours of labeled English audio [30] and is used to fine-tune the encoder and train the RNN-T decoder for the speech recognition task.

3.2. Foundation Model And Task

We pre-train the 600M conformer encoder using three different methods/data to obtain three foundation models. F0: Same as the w2v-BERT XL in [12], we pre-train the encoder on 60K hours of unannotated Libri-Light data with a batch size of 2048 using the Adam optimizer [31] for 400K steps. F1: Following [6], we pre-train the foundation model encoder use BEST-RQ based self-supervised training on YT-U for 800K steps. F2: Starting from F1, we fine-tune the encoder using CTC loss on YT-T for 70K steps. For the downstream speech recognition task, we initialize the encoder using the pre-trained

Table 1: *Comparisons of three speech foundation models on transferring learning to Common Voice (CV) speech recognition task.* ↓ *denotes the smaller the better.* ***# Trainable Encoder Params*** *denotes the number of trainable parameters in the encoder only for the corresponding training method on CV and the whole* 124*M LSTM decoder are trainable for all methods.*

ID	**Pre-training (Loss/Data)** **Unsup**	 **Sup**	**Training Method** **On CV**	**# Trainable** **Encoder Params** ↓	**Computational** **Memory Cost** ↓	**CV Test WER** (%) ↓
B0	-	-	Train All	606 M	11409 MB	15.3
B1	w2v-BERT/Libri-Light	-	Fine-Tune All	606 M	11409 MB	8.2
B2	BEST-RQ/YT-U	-	Fine-Tune All	606 M	11409 MB	8.2
B3	BEST-RQ/YT-U	CTC/YT-T	Fine-Tune All	606 M	11409 MB	8.2
A1	w2v-BERT/Libri-Light	-	Adapter(d=256)	13.3 M	9208 MB	12.7
A2	BEST-RQ/YT-U	-	Adapter(d=256)	13.3 M	9208 MB	9.4
A3	BEST-RQ/YT-U	CTC/YT-T	Adapter(d=256)	13.3 M	9208 MB	8.9
H1	w2v-BERT/Libri-Light	-	HFF	12.3 M	7495 MB	59.4
H2	BEST-RQ/YT-U	-	HFF	12.3 M	7495 MB	11.9
H3	BEST-RQ/YT-U	CTC/YT-T	HFF	12.3 M	7495 MB	11.5

speech foundation model and the output of the encoder is used as input to an RNN-T [6] along with a 6-layer LSTM decoder and dimension 768. We also use exponential moving averaging (EMA) with decay rate 0.9999 for fine-tuning. We update the trainable encoder parameters and LSTM decoder which has 124M trainable parameters on Common Voice data for 100K steps with batch size 256. Specifically, for *Fine-Tune All* or *Train All* in Table 1, all encoder parameters (606M) are trainable. For *Adapter*, the adapter modules inserted in the conformer are trainable and each adapter module is a randomly initialized 2-layer feed-forward network. The total number of trainable trainable encoder parameters is 13.3M with bottleneck dimension 256. For *HFF*, the trainable encoder parameters (12.3M) are feature projectors between the encoder and decoder. If not described explicitly, the parameter efficiency refers to the reduction of the trainable parameters in the encoder only. All experiments are performed on TPUs.

3.3. Evaluation

In this paper, we calculate word error rate (WER) on Common Voice test dataset as the target domain to measure the quality of the adapted model on the downstream speech recognition task. Following [18], we also compare the number of trainable encoder parameters for parameter efficiency and computational memory cost for training efficiency.

4. Results

In this section we present our experimental study on adapting pre-trained foundation models to the target Common Voice domain. The goal of this study is trying to understand the how the quality of foundation models affects the downstream speech recognition task and trying to improve the WER on the target domain with fewer trainable parameters and less computational cost in the adaptation.

4.1. Comparison of Different Foundation Models

In Table 1, we compare the qualities of different foundation models pre-trained on different data and methods by evaluating the WER on Common Voice test data after adaptation. We perform three training methods in transfer learning: Fine-Tune All parameters and two parameter-efficient methods, Adapter [15] and HFF [18].

Table 2: *Comparison of features from different layers of two foundaiton models, F1: BEST-RQ pre-trained on YT-U, F2: BEST-RQ pre-trained on YT-U + CTC fine-tuned on YT-T.*

Feature from Layer	CV Test WER F1	 F2
3	65.2	58.8
7	30.4	29.8
11	15.2	15.1
15	**14.0**	**12.4**
19	31.9	12.5
23	92.6	12.6

By comparing B0 with B1-3 In Table 1, we can observe that a pre-trained encoder can improve the performance of the adapted model on the down-stream speech recognition task when fine-tuning all parameters using the target domain data. However, different pre-trained foundation models (B1, B2, B3) do not show obvious difference on the WER of the adapted model.

On the other hand, the quality of the pre-trained foundation models shows significant effect on the performance of parameter-efficient fine-tuning methods. By Comparing A1 with A2, it is easy to find out that pre-training on more diverse unsupervised data can improve WER of the adapted model greatly from 12.7% to 9.4%. A3 outperforms A2 by absolute 0.5% by pre-training the foundation model further on YT-T data using CTC loss. The same conclusion can be drawn when comparing the results of HFF, especially H1 and H2. It is because that HFF keeps the speech foundation model frozen and use it directly as a feature extractor. The large quality gap between foundation models in H1 and H2 is mainly from the diversity of pre-trained data (Libri-Light vs. YT-U) since the performance of w2v-BERT and BEST-RQ are very close according to the Table 1 in [13].

4.2. Comparison of Features From Different Layers of Two Foundation Models

Knowing that foundation model F2(BEST-RQ pre-trained on YT-U + CTC fine-tuned on YT-T) performs better than F1(BEST-RQ pre-trained on YT-U) in efficient transfer learning,

Table 3: *Combing Drop4, HFF and Adapter to improve parameter and training efficiency in transfer learning to Common Voice (CV) speech recognition task. Drop4 denotes dropping top 4 conformer layers from the pre-trained speech foundation model. ↓ denotes the smaller the better. The (x% ↓) in E2 and E3 represents the relative improvement compared to E1.*

ID	Training Method On CV	# Trainable Encoder Params ↓	Computational Memory Cost ↓	CV Test WER (%) ↓
B3	Fine-Tune All	606 M	11409 MB	8.2
A3	Adapter(d=256)	13.3 M	9208 MB	8.9
H3	HFF	12.3 M	7495 MB	11.5
E1	HFF+Adapter(d=128) at all layers	18.6 M	9178 MB	8.6
E2	Drop4+HFF+Adapter(d=128) at all layers	15.4 M (17.2% ↓)	7922 MB (13.7% ↓)	8.7
E3	Drop4+GAFF+Adapter(d=128) at all layers	12.7 M (31.7% ↓)	7950 MB (13.4% ↓)	8.6

we evaluate and compare the performance of features of their intermediate layers in Table 2. In the training, we update the parameters of RNN-T decoder only and its input is from the corresponding layer specified in the first column of Table 2. Results show that top layers of the foundation model pre-trained using unsupervised loss performed worse compared to the foundation model pre-trained using both unsupervised loss and supervised loss. For F2, even if its features from top layers do not degrade that much as F1, they are not better than the features from middle layers. Our observation is aligned with the results in [19, 18].

4.3. After Pre-Training, How Many Layers to Keep for Down-stream ASR

From the results in Table 2, we observe that features from top few layers cannot perform better than features from middle layers. It is straightforward to ask a question, do we really need these layers for transfer learning of down-stream tasks? In Table 4, we would like to investigate whether it is safe to drop top layers without degrading performance, where the foundation model F2(BEST-RQ pre-trained on YT-U + CTC fine-tuned on YT-T) is used. Results demonstrate that it is safe to drop top 4 layers without compromising the WER on the down-stream tasks for Adapter and HFF, with reduction on training memory consumption by relative 13.7% and 13.2% respectively. It is reasonable to state that as long as the pre-training losses are not compatible with the down-stream task loss (BEST-RQ vs. RNN-T or CTC vs. RNN-T), dropping top few layers would not affect its performance in transfer learning for parameter-efficient fine-tuning methods. This conclusion does not hold for fine-tuning all parameters since all variables can be updated in this case and dropping parameters would hurt performance.

4.4. Combing Drop4, HFF and Adapter for Parameter and Training Efficiency

In this section, we try to combine Drop4(dropping top 4 layers from the pre-trained foundation model), Adapter and HFF together to get efficiency in parameter and training at the same time. Comparing B3, A3, and H3 in Table 3, it is easy to see that Adapter can obtain a decent performance on target domain, while HFF is more training efficient. E1 which combines Adapter and HFF as [18] performs better than either of them alone. By adding Drop4 to E1, E2 can achieve WER 8.7% with 17.2% fewer trainable encoder parameters and 13.7% less computational memory cost than E1.

Table 4: *Comparing the performance of two parameter-efficient fine-tuning methods Adapter and HFF by dropping top few conformer layers from the pre-trained speech foundation model. Memory denotes the computational memory cost and WER is evaluated on the Common Voice test data.*

Layers to	Adapter		HFF	
Keep	Memory	WER	Memory	WER
$0-23$	9109	8.9	7495	11.5
$0-19$	7862	9.0	6503	11.5
$0-15$	6623	9.4	5564	11.6

4.5. Replacing HFF with GAFF

Although a significant improvement on parameter efficiency and training efficiency can be obtained in E2, there is a minor regression on the WER. Besides, it is not optimal to use a manually designed hierarchical structure. In this section, we replace the balanced hierarchical structure in HFF with GAFF proposed in Section 2.2. It is obvious in Table 3 that E3 is more parameter efficient than E2 and achieves the same performance on the target domain. Results show that E3 takes 31.7% fewer trainable encoder parameters and 13.4% less computation memory cost than E1 with the same WER 8.6%.

5. Conclusion

In this paper, we re-investigate the efficient transfer learning of speech foundation model using feature fusion methods. Extensive results show that the quality of foundation models plays a more important role for parameter-efficient methods than fine-tuning all parameters. Besides, we also notice that dropping the top 4 layers of the speech foundation model does not affect the quality of adapted model. After combining with Global Attentional Feature Fusion (GAFF), the new method achieves a better parameter efficiency and training efficiency than the compared methods.

6. References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer." *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[6] Y. Zhang, D. S. Park, W. Han, J. Qin, A. Gulati, J. Shor, A. Jansen, Y. Xu, Y. Huang, S. Wang *et al.*, "Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[7] D. Hwang, A. Misra, Z. Huo, N. Siddhartha, S. Garg, D. Qiu, K. C. Sim, T. Strohman, F. Beaufays, and Y. He, "Large-scale asr domain adaptation using self-and semi-supervised learning," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6627–6631.

[8] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3497–3501.

[9] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6889–6893.

[10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[11] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[12] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 244–250.

[13] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3915–3924.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[15] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[16] R. Karimi Mahabadi, J. Henderson, and S. Ruder, "Compacter: Efficient low-rank hypercomplex adapter layers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1022–1035, 2021.

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[18] Z. Huo, K. C. Sim, B. Li, D. Hwang, T. N. Sainath, and T. Strohman, "Resource-efficient transfer learning from speech foundation model using hierarchical feature fusion," *arXiv preprint arXiv:2211.02712*, 2022.

[19] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 914–921.

[20] 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.

[21] B. Li, D. Hwang, Z. Huo, J. Bai, G. Prakash, T. N. Sainath, K. C. Sim, Y. Zhang, W. Han, T. Strohman *et al.*, "Efficient domain adaptation for speech foundation models," *arXiv preprint arXiv:2302.01496*, 2023.

[22] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[23] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[24] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[25] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[27] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, "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)*. IEEE, 2020, pp. 7669–7673.

[28] H. Purwins, B. Li *et al.*, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[29] C.-C. Chiu, A. Narayanan *et al.*, "RNN-T models fail to generalize to out-of-domain audio: Causes and solutions," in *Proc. SLT*. IEEE, 2021.

[30] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[31] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015.