Learnable Layer Selection and Model Fusion for Speech Self-Supervised Learning Models

Sheng-Chieh Chiu[1,2], *Chia-Hua Wu*[2], *Jih-Kang Hsieh*[1,2], *Yu Tsao*[2], *Hsin-Min Wang*[2]

[1]National Tsing Hua University, Taiwan
[2]Academia Sinica, Taiwan

maxwu@iis.sinica.edu.tw

Abstract

In this paper, we investigate methods for fusing feature representations derived from multiple speech self-supervised learning (SSL) models, along with techniques to determine the optimal layer within each model. We evaluate five fusing strategies, finding that temporal interleaved concatenation is the most robust and effective for the SUPERB ASR task. Additionally, we demonstrate that Gumbel layer selection can automatically select the most appropriate SSL layer with better performance than the commonly used weighted sum method. Furthermore, dimension-wise Gumbel layer selection shows promise in adaptive combination of layers of a single SSL model. Finally, we show that joint SSL model fusion and dimension-wise Gumbel layer selection further enhances effectiveness.

Index Terms: ASR, self-supervised learning, feature fusion, layer selection, SUPERB

1. Introduction

Self-supervised learning (SSL) has gained huge attention in the speech community as a popular method. Speech SSL models, pre-trained on vast amounts of unlabeled speech data, demonstrate remarkable results when fine-tuned for downstream tasks with limited transcribed speech [1]. To date, a variety of successful speech SSL models have emerged, including Wav2vec 2.0 [2], HuBERT [3], WavLM [4], and Data2vec [5], among others.

These SSL models are trained with different training objectives under different conditions, model architectures, and modalities. To more fairly evaluate the performance and generalization ability of these SSL models in different downstream tasks, attention has been paid to initiatives such as SUPERB (Speech Processing Universal Performance Benchmark) [6]. Recent extensions of SUPERB include SUPERB-SG [7] and ML-SUPERB [8] to include more tasks. In the experiments of SUPERB [6], it was found that the representation of the last layer of various SSL models did not always perform best in various downstream tasks, but the weighted sum representation of all layers showed good potential. Many related studies have found that each layer of these SSL models contains information useful for different tasks [4, 5, 9, 10, 11, 12]. For example, representations of the last layers are beneficial for ASR, while representations of the earlier layers are more effective for speaker verification.

Recent advances also involve integrating speech SSL models into end-to-end (E2E) frameworks, serving as front-end feature extractors [13, 14, 15, 6, 16, 17]. These SSL models replace traditional spectral features (SF), such as log Mel-filterbanks (FBANK) [18, 19, 20]. Through a completely unsupervised training process, these SSL models autonomously acquire their own feature extraction modules. In light of the success of these SSL models, methods for combining their representations have attracted attention, including the combination of traditional features and SSL features [21, 22, 23] and the fusion of multiple SSL models [24, 25, 26].

Most previous studies utilize either the last layer or the weighted sum of all layers as features. To more effectively utilize specific layers for downstream tasks, we introduce the Gumbel-Softmax (GS) trick [27] to explore ways to automatically select SSL model layers for input into downstream ASR models. We propose two selection methods, **Gumbel layer selection** and **dimension-wise Gumbel layer selection**, aiming to automatically select appropriate layers for downstream tasks during training.

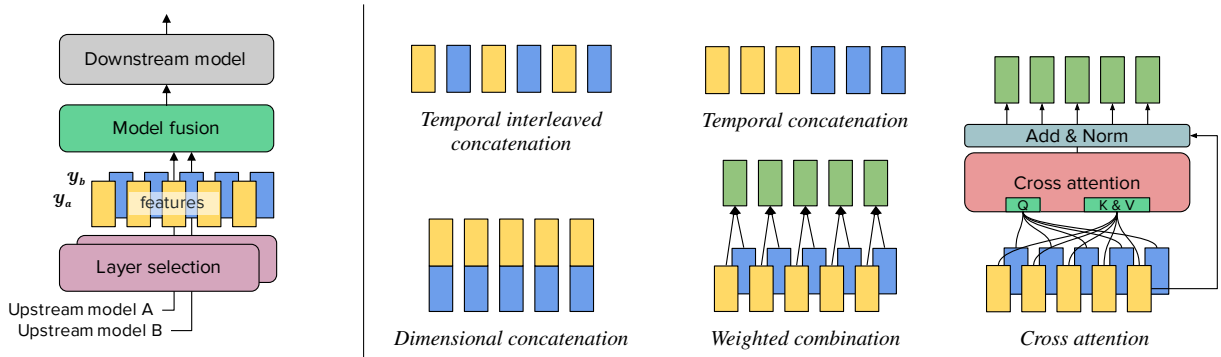
Tang et al. proposed an effective method to fuse features of multiple SSL models. Their research shows that fusing features from SSL models has a greater impact on speech recognition performance than probabilistically combining the predictions of individual downstream models [24]. Inspired by this, in this work, we also study multiple fusion methods for HuBERT, WavLM+, and Data2vec on the SUPERB ASR task, including temporal interleaved concatenation, temporal concatenation, dimensional concatenation, weighted combination, and cross attention.

Overall, in this paper, we discuss both aspects of layer selection and model fusion and combine their advantages to enhance overall performance. Our contributions can be summarized as follows:

- We propose an automatic layer selection method called **dimension-wise Gumbel layer selection**, which can effectively leverage specific layers to fine-tune downstream tasks. This method achieves lower word error rates compared to using the weighted sum of all layers as features.
- We investigate SSL model fusion in E2E ASR and find that **temporal interleaved concatenation** consistently outperforms other fusion methods. This finding highlights the importance of fusion techniques in improving ASR performance.
- We provide guidance on effectively integrating multiple SSL models. Our results showcase the potential of temporal interleaved concatenation-based model fusion and Gumbel layer selection in enhancing ASR performance using multiple SSL encoders. These findings are expected to generalize to other speech tasks.

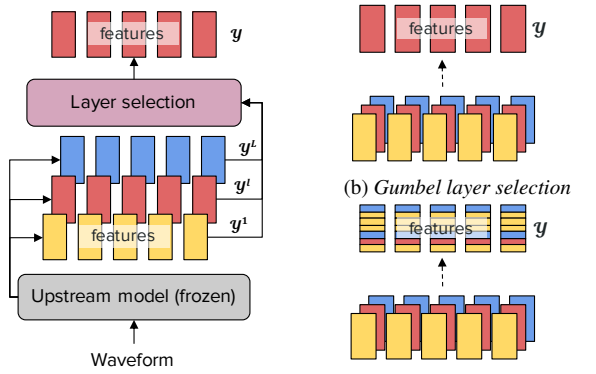
2. Methodology

The structure of the proposed model is shown in Figure 1a. It consists of layer selection modules for each upstream SSL model and a model fusion module.



(a) *Overall model structure* (b) *Model fusion methods*

Figure 1: *(a) The proposed model structure. (b) The five model fusion methods.*



(a) *Flowchart of layer selection.* (b) *Gumbel layer selection* (c) *Dimension-wise Gumbel layer selection*

Figure 2: *Illustration of layer selection: (a) the flowchart of layer selection, (b) Gumbel layer selection, and (c) Dimension-wise Gumbel layer selection.*

2.1. Model fusion

Since upstream SSL model fusion can bring significant performance improvements to downstream tasks, in this study, we delve deeper into model fusion and discuss the following five methods (Figure 1b):

Temporal concatenation: The two feature sequences $\boldsymbol{y_a}$ and $\boldsymbol{y_b}$ are concatenated along time and input to the downstream model.

Dimensional concatenation: The two feature vectors at each time point are concatenated along the feature dimension.

Cross attention: Assuming $\boldsymbol{y_a}$ outperforms $\boldsymbol{y_b}$, cross attention is used to capture the residual information between them. This information is then added to $\boldsymbol{y_a}$, followed by normalization of the resulting features, to obtain an enhanced representation $\boldsymbol{y}$. The process is formulated as

$$\boldsymbol{y} = norm(\boldsymbol{y_a} + attention(\boldsymbol{Q_b}, \boldsymbol{K_a}, \boldsymbol{V_a})). \quad (1)$$

Weighted combination: $\boldsymbol{y_a}$ and $\boldsymbol{y_b}$ are combined with a trainable weight λ using $\boldsymbol{y} = \lambda \boldsymbol{y_a} + (1-\lambda)\boldsymbol{y_b}$, where $0 < \lambda < 1$.

Temporal interleaved concatenation: $\boldsymbol{y_a}$ and $\boldsymbol{y_b}$ are arranged alternately in sequence.

2.2. Layer selection

Many studies have shown that feature representations at different layers of upstream SSL models excel in different downstream tasks. Therefore, automated strategies for selecting the best layer from an upstream SSL model are worth exploring. In this study, we compare the following selection strategies:

Weighted sum: This is the default feature selection strategy provided by SUPERB. The final feature $\boldsymbol{y}$ is derived from an SSL model via

$$\boldsymbol{y} = \sum_{l}^{L} p^l \boldsymbol{y^l}, \quad (2)$$

where L is the number of layers, $\boldsymbol{y^l}$ is the feature vector sequence of the l-th layer, and the weight p^l is derived via the softmax function of trainable parameters $\{w^l\}_{l=1}^{L}$ via

$$p^l = \underset{l}{softmax}\{w^l\}. \quad (3)$$

Highest weight: The layer with the highest weight based on Eq. (3) is selected.

Best performing layer: The layer with the best downstream task performance is selected from the first three highly weighted layers based on Eq. (3).

Gumbel layer selection: As shown in Figure 2b, the Gumbel-Softmax trick is used to select the best layer from the SSL model during downstream model training. The probability distribution across layers is determined by the learnable weights and temperature τ as

$$p^l(\tau) = \underset{l}{softmax}\left\{\frac{w^l}{\tau}\right\}. \quad (4)$$

When $\tau \to 0$, the probability will converge to one hot distribution, and the best feature $\boldsymbol{y}$ is

$$\boldsymbol{y} = \sum_{l} gumbel\text{-}max_l(p^l(\tau))\boldsymbol{y^l}. \quad (5)$$

Dimension-wise Gumbel layer selection: Similar to Gumbel layer selection, but instead of selecting a certain layer as the best feature, dimension-wise Gumbel layer selection selects the best layer for each dimension, as shown in Figure 2c. Specifically, the Gumbel-Softmax trick is applied simultaneously in each dimension to select the best layer separately.

Table 1: *Comparison of different model fusion methods (in WER (%)).*

Fusion methods	WavLM+&HuBERT		Data2vec&WavLM+	
	Weighted-sum	Best performing	Weighted-sum	Best performing
Temporal concatenation	5.38	5.79	4.82	4.56
Dimensional concatenation	5.64	5.37	5.04	4.50
Cross attention	5.54	5.34	5.07	4.77
Weighted combination	5.66	5.39	5.04	4.52
Temporal interleaved concatenation	**5.17**	**5.26**	**4.68**	**4.36**

3. Experiments and results

In this section, we detail the experimental setup and results.

3.1. Experimental setup

All our experiments were performed following the recipe for the ASR downstream task in S3PRL[1]. We selected HuBERT Base, WavLM Base+[2] and Data2vec Base for testing, focusing on the LibriSpeech ASR task, trained on train-clean-100h and evaluated in word error rate (WER) on the test-clean dataset [28]. For simplicity, hereafter these three SSL models will be referred to as HuBERT, WavLM+ and Data2vec, respectively. We followed the constrained track in SUPERB, where SSL upstream models are frozen during downstream model training and only the feature selection and downstream model are learnable.

We trained the downstream ASR models for 200k steps with a learning rate of 0.0001 using the SUPERB's default settings and without using a language model. For Gumbel layer selection and dimension-wise Gumbel layer selection, the models marked with "Anneal" indicate that the temperature in Eqs. (4) and (5) during training starts at 1.0, decays linearly to 0.1 in the first 1k steps, then decays exponentially to 0.0001 in the subsequent 10k steps, and remains constant until the end of training. All fusion experiments were performed on an Nvidia RTX 3080, with an average cost per experiment of 80 hours, and all single-model experiments were performed on an Nvidia RTX 2080 Ti, with an average cost per experiment of 60 hours.

3.2. Comparison of different model fusion methods

In this experiment, we compare the five fusion methods on two groups of SSL models with different model similarities, one group combines two similar models (WavLM+&HuBERT) and the other group combines two relatively dis-similar models (Data2vec&WavLM+). For each SSL model, the weighted sum of all layers or the best performing layer (see Section 2.2) is used in model fusion. In the temporal concatenation and temporal interleaved concatenation methods, for each group, we first used the features of the first model and then used the features of the second model. In the cross attention method, for each group, the features of the first model were used as the key and value, and the features of the second model were used as the query.

From Table 1, several observations can be drawn. First, it is evident that temporal interleaved concatenation consistently outperforms other fusion methods in all cases. Second, under the weighted sum condition, only temporal concatenation and temporal interleaved concatenation always perform better than the individual models in both groups (see Table 2 for the performance of individual SSL models). Third, when using the best performing layer, all fusion methods fail to reduce WER in the WavLM+&HuBERT group, while most fusion methods can reduce WER in the Data2vec&WavLM+ group. The reason may be that WavLM+ and HuBERT have similar characteristics, and their best-performing layers may not be complementary, so their fusion cannot improve downstream ASR tasks. Fourth, no matter which fusion method is used, the fusion of Data2vec and WavLM+ always performs better than the fusion of WavLM+ and HuBERT. Overall, our best results for fusing Data2vec and WavLM+ are 4.68% (weighted sum) and 4.36% (best-performing layer). These results are comparable to or better than the 4.62% achieved by an equal-weighted combination of Data2vec and HuBERT reported in [24].

Table 2: *Comparison of different layer selection methods on individual SSL models (in WER (%)). Lx in parentheses indicates the selected feature layer. Gumbel and Dim.-wise Gumbel denote Gumbel layer selection and dimension-wise Gumbel layer selection, respectively.*

Selection methods	HuBERT	WavLM+	Data2vec
Weighted sum	6.40	5.58	5.01[3]
Highest weight	5.93 (L10)	7.34 (L9)	4.60 (L9)
Best performing	5.93 (L10)	5.20 (L10)	4.60 (L9)
Gumbel	6.20 (L10)	5.65 (L10)	4.53 (L9)
+ Anneal	5.89 (L10)	5.15 (L10)	4.58 (L9)
Dim.-wise Gumbel	**5.76**	5.32	**4.51**
+ Anneal	6.03	**5.00**	4.53

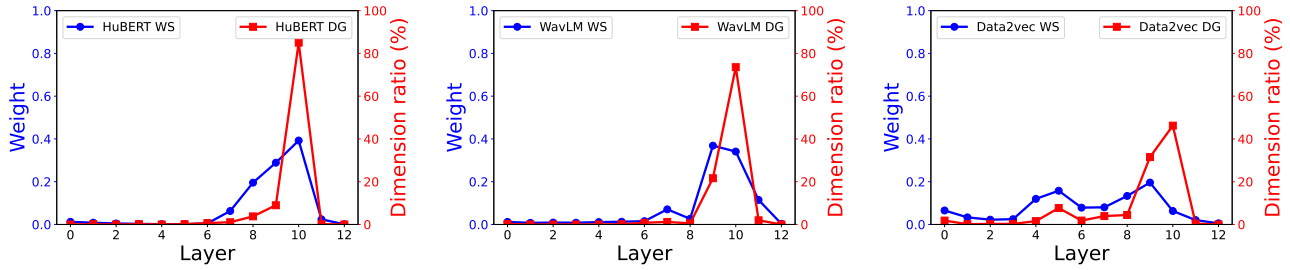
3.3. Comparison of different layer selection methods on individual SSL models

In this experiment, we compare different layer selection methods on three SSL models. As shown in Table 2, for all three SSL models, the best performing layer consistently outperforms the weighted sum feature. But the best performing layer can only be determined after an exhausted comparison of ASR models trained using different layers of the SSL model, which is obviously impractical. Although it is possible to obtain feature representations from multiple layers of an SSL model by selecting the highest-weighted layer directly based on the common weighted sum method, the highest-weighted layer is not necessarily the layer with the best performance. In contrast, our proposed Gumbel layer selection method can select the correct best layer, which not only outperforms the weighted sum feature but also performs comparably to or even better than the best performing layer. Furthermore, upon integrating the annealing mechanism, Gumbel layer selection consistently outperforms

[1] https://github.com/s3prl/s3prl

[2] WavLM Base+ is WavLM Base trained with 94k hours of speech.

[3] 5.01% is slightly higher than 4.94% reported in [24]. This inconsistency is common and acceptable in the S3PRL community.



(a) *HuBERT* (b) *WavLM+* (c) *Data2vec*

Figure 3: *The weight distribution of the weighted sum method (WS) and the dimension ratio of the dimension-wise Gumbel layer selection method (DG). Since the dimension-wise Gumbel layer selection method will select the best layer for each feature dimension, for the dimension ratio, 80% means that a particular layer is selected for 80% of the dimensions.*

the best performing layer for all three SSL models. We found that annealing quickly converges to the best-performing layer in approximately the first 15k training steps.

Although Gumbel layer selection performs well in our experiments, there is still a risk of selecting the wrong layer and degrading downstream task performance. Therefore, we further propose dimension-wise Gumbel layer selection, which automatically selects the best layer for each dimension to form new features. From Table 2, it is clear that dimension-wise Gumbel layer selection always outperforms Gumbel layer selection with or without using the annealing mechanism. In our experiments, we found that for Gumbel layer selection, when the layer selection decision changes, the downstream model faces drastic input switching, and annealing helps stabilize the layer selection earlier, allowing the downstream model to better adapt to the upstream model output. In contrast, for dimension-wise Gumbel layer selection, changes in the layer selection do not greatly affect the output. This may explain why the benefit from annealing is less pronounced compared to Gumbel layer selection.

Figure 3 shows the weight distribution of the weighted sum method and the dimension ratio of the dimension-wise Gumbel layer selection method. It can be found that both methods have high values corresponding to the best performing layers. However, the distribution of dimension-wise Gumbel layer selection is relatively concentrated. In the Data2vec graph, the weighted sum method has two obvious peaks, respectively at layer 5 and layer 9. But for dimension-wise Gumbel layer selection, the peak value of layer 5 is less obvious. For further analysis, we conducted additional ASR experiments on the 5-th layer of Data2vec, and found that the performance was quite poor[4]. This confirms the reliability of dimension-wise Gumbel layer selection. The weighted sum method may be too biased toward layer 5, reducing the weight of the best performing layer, resulting in performance degradation.

3.4. Performance of joint layer selection and model fusion

We aim to combine the best model fusion method in Section 3.2 with the most effective layer selection method in Section 3.3. According to the results in Table 1, the most effective fusion method is temporal interleaved concatenation. Therefore, in Table 3, we present the experimental results obtained using various layer selection methods before temporal interleaved concatenation.

[4]Training the ASR model using layer 9 of Data2vec resulted in a WER of 4.6%, while using layer 5 resulted in a WER of 8.44%.

For the WavLM+&HuBERT group, the combination of temporal interleaved concatenation-based model fusion and weighted sum-based layer selection achieves the best performance. Although the combination of temporal interleaved concatenation and Gumbel layer selection with annealing and the combination of temporal interleaved concatenation and dimension-wise Gumbel layer selection outperform the combination of temporal interleaved concatenation and the best performing layer, they still perform slightly worse than the combination of temporal interleaved concatenation and weighted sum.

For the Data2vec&WavLM+ group, both Gumbel layer selection with annealing and dimension-wise Gumbel layer selection outperform the weighted sum method, respectively achieving superior WERs of 4.56% and 4.40%, which are lower than the lowest ever WER of 4.62% reported in [24].

Table 3: *Performance of joint layer selection and model fusion.*

Selection methods	**WavLM+ & HuBERT**	**Data2vec & WavLM+**
Weighted sum	**5.17**	4.68
Best performing	5.26 (L10 & L10)	4.36 (L9 & L10)
Gumbel + Anneal	5.25 (L10 & L10)	4.56 (L9 & L10)
Dim.-wise Gumbel	5.20	**4.40**

4. Conclusions

In this paper, we have proposed temporal interleaved concatenation for fusing two speech SSL models for downstream ASR tasks. Experimental results show that temporal interleaved concatenation outperforms other model fusion methods. In addition, we have also proposed Gumbel layer selection and dimension-wise Gumbel layer selection for automatically selecting the best layer from an SSL model. Experimental results show that both Gumbel layer selection and dimension-wise Gumbel layer selection outperform the commonly used weighted sum method. Furthermore, we have also demonstrated the effectiveness of combining temporal interleaved concatenation and Gumbel layer selection (or dimension-wise Gumbel layer selection). Our method achieves the lowest ever word error rate on the SUPERB ASR task.

5. Acknowledgements

This work was partially supported by the National Science and Technology Council of Taiwan under grant numbers: MOST 110-2221-E-001-015-MY3 and NSTC 112-2221-E-001-009-MY3.

6. References

[1] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe, T. N. Sainath, and S. Watanabe, "Self-supervised speech representation learning: A review," *IEEE J.Sel.Top.Sig.Proc.*, vol. 16, no. 6, pp. 1179–1210, 2022.

[2] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.

[3] 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 Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[4] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Top. Sig. Proc.*, vol. 16, no. 6, pp. 1505–1518, 2022.

[5] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proc. ICML*, 2022.

[6] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K.-t. Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee, "SUPERB: Speech processing Universal PERformance Benchmark," in *Proc. Interspeech*, 2021.

[7] H.-S. Tsai, H.-J. Chang, W.-C. Huang, Z. Huang, K. Lakhotia, S.-w. Yang, S. Dong, A. Liu, C.-I. Lai, J. Shi, X. Chang, P. Hall, H.-J. Chen, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee, "SUPERB-SG: Enhanced speech processing Universal PERformance Benchmark for semantic and generative capabilities," in *Proc. ACL*, 2022.

[8] J. Shi, D. Berrebbi, W. Chen, H.-L. Chung, E.-P. Hu, W. P. Huang, X. Chang, S.-W. Li, A. Mohamed, H.-y. Lee, and S. Watanabe, "ML-SUPERB: Multilingual speech Universal PERformance Benchmark," in *Proc. Interspeech*, 2023.

[9] Y.-A. Chung, Y. Belinkov, and J. Glass, "Similarity analysis of self-supervised speech representations," in *Proc. ICASSP*, 2021.

[10] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *Proc. ASRU*, 2021.

[11] D. Ma, N. Ryant, and M. Liberman, "Probing acoustic representations for phonetic properties," in *Proc. ICASSP*, 2021.

[12] A. Pasad, B. Shi, and K. Livescu, "Comparative layer-wise analysis of self-supervised speech models," in *Proc. ICASSP*, 2023.

[13] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, 2019.

[14] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," arXiv preprint arXiv:1910.05453, 2020.

[15] X. Chang, T. Maekaku, P. Guo, J. Shi, Y.-J. Lu, A. S. Subramanian, T. Wang, S.-w. Yang, Y. Tsao, H.-y. Lee, and S. Watanabe, "An exploration of self-supervised pretrained representations for end-to-end speech recognition," in *Proc. ASRU*, 2021.

[16] K. D. N, P. Wang, and B. Bozza, "Using large self-supervised models for low-resource speech recognition," in *Proc. Interspeech*, 2021.

[17] J. Zhao and W.-Q. Zhang, "Improving automatic speech recognition performance for low-resource languages with self-supervised models," *IEEE J. Sel. Top. Sig. Proc.*, vol. 16, no. 6, pp. 1227–1241, 2022.

[18] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, "Applying wav2vec2.0 to speech recognition in various low-resource languages," arXiv preprint arXiv:2012.12121, 2021.

[19] A. Wu, C. Wang, J. Pino, and J. Gu, "Self-supervised representations improve end-to-end speech translation," in *Proc. Interspeech*, 2020.

[20] A. T. Liu, S.-W. Li, and H.-y. Lee, "TERA: Self-supervised learning of transformer encoder representation for speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2351–2366, 2021.

[21] S.-J. Chen, W. Xia, and J. H. L. Hansen, "Scenario aware speech recognition: Advancements for Apollo fearless steps & CHiME-4 corpora," in *Proc. ASRU*, 2021.

[22] P. Vieting, C. Lüscher, W. Michel, R. Schlüter, and H. Ney, "On architectures and training for raw waveform feature extraction in ASR," in *Proc. ASRU*, 2021.

[23] D. Berrebbi, J. Shi, B. Yan, O. Lopez-Francisco, J. D. Amith, and S. Watanabe, "Combining spectral and self-supervised features for low resource speech recognition and translation," in *Proc. Interspeech*, 2022.

[24] C. Tang, Y. Wang, X. Chen, and W.-Q. Zhang, "Exploring effective fusion algorithms for speech based self-supervised learning models," in *Proc. NCMMSC*, 2022.

[25] S.-J. Chen, J. Xie, and J. H. Hansen, "FeaRLESS: Feature refinement loss for ensembling self-supervised learning features in robust end-to-end speech recognition," in *Proc. Interspeech*, 2022.

[26] T. Srivastava, J. Shi, W. Chen, and S. Watanabe, "EFFUSE: Efficient self-supervised feature fusion for E2E ASR in multilingual and low resource scenarios," arXiv preprint arXiv:2310.03938, 2023.

[27] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," in *Proc. ICLR*, 2016.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.