INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Investigating Pre-trained Audio Encoders in the Low-Resource Condition

Hao Yang, Jinming Zhao, Gholamreza Haffari, Ehsan Shareghi

Department of Data Science & AI, Monash University

first.last@monash.edu

Abstract

Pre-trained speech encoders have been central to pushing state-of-the-art results across various speech understanding and generation tasks. Nonetheless, the capabilities of these encoders in low-resource settings are yet to be thoroughly explored. To address this, we conduct a comprehensive set of experiments using a representative set of 3 state-of-the-art encoders (Wav2vec2, WavLM, Whisper) in the low-resource setting across 7 speech understanding and generation tasks. We provide various quantitative and qualitative analyses on task performance, convergence speed, and representational properties of the encoders. We observe a connection between the pre-training protocols of these encoders and the way in which they capture information in their internal layers. In particular, we observe the Whisper encoder exhibits the greatest low-resource capabilities on content-driven tasks in terms of performance and convergence speed.[1]

Index Terms: speech encoders, low-resource setting, speech understanding

1. Introduction

In recent years, the advancement in various speech tasks has largely been driven by encoder models that are typically pre-trained on large-scale datasets via self-supervised learning [1, 2]. As the prominent examples, the Wav2vec2 [3] model leverages a speech quantiser module to simulate token prediction of BERT [4], while HuBERT [5] adopts a clustering method to produce discrete labels for each feature vector to imitate masked language model loss. Similar to HuBERT, WavLM [6] proposes a denoising masked speech modelling, which masks segments of speech signals to predict the pseudo-label at the output.

While it is expected that these pre-trained encoders produce universal speech features effective for a broad range of downstream tasks, in practice pre-trained models still require large amounts of fine-tuning labelled data to produce state-of-the-art performance, or to converge. This could be attributed to their inefficiency in utilising the representation space [7], as well as the difference between the objectives for pre-training and fine-tuning steps [8]. For instance, the pre-training objective is typically designed in the absence of any textual or content cue (i.e., to predict masked speech segments), while the downstream tasks (i.e., automatic speech recognition and speech translation) often require a mapping between speech and text. An exception in this space is the Whisper encoder-decoder model [8], which leverages weak supervision through large scale crawled data of (audio, transcript) pairs from the internet, and is pre-trained by learning the mapping between speech and decoder outputs (i.e., in transcription or and translation).

To better understand the interplay between pre-training protocols of speech encoders, the amount of fine-tuning data, and speech task types, we conduct a comprehensive study in this work. We evaluate a set of three very recent speech models (Wav2vec2, WavLM, and Whisper) and assess their performance on 7 downstream tasks (covering content, speaker and semantic types) in the low-resource setting. Through extensive experiments in the low-resource setting, we found that Whisper significantly outperforms Wav2vec2 and WavLM by a large margin on content-related (content, semantics) tasks, and shows performance degradation when speaker information is required for a downstream task. To investigate how this behaviour is connected with Whisper's pre-training and representational properties, we examine layer-wise information of Whisper and the other baselines. Additionally, through qualitative and quantitative analyses, we highlight how Whisper's superior performance could be attributed to the properties of its representational space. We hope our study to provide insights for a more effective use of pre-trained speech encoders in the resource-constrained setting.

2. Related work

We provide a brief overview of some of the well-known pre-trained speech models. The Wav2vec [9] model proposed two multi-layer convolutional neural networks stacked on top of each other to map raw audio to a representation instead of traditional acoustic feature extraction. Subsequently, Wav2vec 2.0 [3] attached Transformer layer [10] to the feature extractor layer and utilised InfoNCE loss [11] and quantiser modules to predict masked spans of the representation at the output. Similar to BERT [4] in the text domain, HuBERT [5] adopts the clustering method to produce discrete labels for each input feature, with the purpose of imitating masked language model loss. WavLM [6] is proposed with denoising masked speech modeling, which randomly transforms the input audio and masks 50% of speech signals to predict the labels corresponding to the masked positions. Additionally, it follows the idea proposed by HuBERT [5], converting continuous signals into discrete labels through a clustering method, and models the discrete labels as targets. WavLM achieves state-of-art results on several downstream tasks from the SUPERB benchmark [12]. As an exception to the above models, Whisper [8] is pre-trained under weak supervision through crawled audio-transcript pairs from the internet. Transcription and translation are set as pre-training targets, pre-training the model via learning to map the input audio to its transcript as the output.

Hsu et al. [13] highlighted the benefits of pre-training on several domains. The pre-trained audio representations have been investigated from different aspects [14, 15, 16, 17, 18,

[1] https://github.com/YangHao97/investigateAudioEncoders

10.21437/Interspeech.2023-343

Table 1: *Main results and the number of updates* 🕐 *required for fine-tuning in low-resource scenarios. The encoders' details are as follows: W2V2: 317M/24 layers, WavLM: 317M/24 layers, Whisper* Base*: 21M/6 layers, Whisper* Small*: 88M/12 layers, Whisper* Medium*: 307M/24 layers. The* ***bold*** *font and underlined numbers denote the fastest convergence speed and performance, respectively.*

		SD		SF		IC		KS		ASR		SID		ST	
Tr.	Model	DER ↓	🕐	F1 ↑	🕐	Acc ↑	🕐	Acc ↑	🕐	WER ↓	🕐	Acc ↑	🕐	BLEU ↑	🕐
1%	W2V2	10.23	6.7k	57.88	42k	12.54	6.8k	85.17	9.2k	99.99	50k	9.74	4.5k	0.17	**3k**
	WavLM	6.38	0.4k	75.56	98k	26.02	3.25k	93.57	10k	17.84	8.4k	12.69	11k	0.69	14k
	Whisper-Base	7.24	**0.2k**	70.47	84k	67.04	**2.75k**	96.79	1k	26.43	16k	2.66	**2k**	0.87	30k
	Whisper-Small	5.37	1.2k	74.45	**38k**	57.63	4.25k	96.62	**0.5k**	20.27	10k	3.35	3.5k	0.94	28k
	Whisper-Medium	5.23	0.4k	77.76	48k	73.74	3k	96.72	0.75k	17.56	**5k**	3.97	4.5k	0.98	24k
5%	W2V2	9.20	4k	78.29	**58k**	53.07	16k	94.25	20k	14.70	100k	41.90	47k	0.20	**4k**
	WavLM	5.16	1.8k	86.50	92k	91.30	5k	95.91	**1k**	7.90	**50k**	55.52	27k	4.19	12k
	Whisper-Base	6.84	1.6k	82.80	94k	95.39	3k	97.44	**1k**	16.18	100k	11.63	15k	3.41	20k
	Whisper-Small	4.89	**1k**	85.83	70k	95.78	**2.5k**	97.73	**1k**	11.76	90k	13.47	16k	3.84	26k
	Whisper-Medium	4.59	2.4k	87.60	62k	98.23	**2.5k**	97.95	**1k**	9.75	84k	17.94	**13k**	4.22	30k
10%	W2V2	8.21	6k	80.74	90k	77.91	45k	95.85	15.5k	5.96	90k	56.09	78k	7.21	25k
	WavLM	4.76	1k	88.84	**80k**	94.38	**2.5k**	96.82	**0.5k**	5.99	98k	79.51	61k	6.99	**22k**
	Whisper-Base	5.89	**0.2k**	85.15	86k	96.92	3k	97.24	3k	13.41	100k	19.48	**13k**	5.19	28k
	Whisper-textscSmall	4.69	0.6k	87.90	98k	96.44	**2.5k**	97.63	2k	9.47	86k	23.04	**13k**	6.09	23k
	Whisper-Medium	4.38	0.4k	89.80	96k	98.78	7.5k	97.96	1k	7.74	**74k**	30.05	**13k**	6.48	29k

19], indicating they can generalise to wide range of corpora. However, Yang et al. [7] demonstrated that the Wav2vec2 speech encoder under-utilises the representation space, and proposed a self-supervision approach to improve the representation isotropy, leading to faster convergence during downstream task training. Yi et al. [20] applied Wav2vec2 in low-resource conditions for multilingual speech recognition and verified the potential for transfer-ability of monolingual Wav2vec2 to other languages. The Whisper encoder-decoder model has exhibited its capabilities in zero-shot settings [8] by achieving state-of-art performance on various tasks, from multilingual ASR, and translation, to Language Identification, and Long-form Transcription.

3. Experiments

In this section, we first describe our experimental settings (§3.1). Next, we report the results on 7 downstream tasks in low-resource scenarios (§3.2). Lastly, we provide an analysis of Whisper encoders on the quantitative and qualitative properties compared to two other widely used encoders, Wav2vec2 and WavLM (§3.3).

3.1. Experimental Settings

Tasks and Dataset. We conducted experiments on various tasks from SUPERB and SUPERB-SG benchmarks:[2] Automatic Speech Recognition (ASR), Speaker Diarisation (SD), Intent Classification (IC), Slot Filling (SF), Keyword Spotting (KS), Speaker Identification (SID), and Speech Translation (ST). For evaluation, we use word error rate (WER), diarisation error rate (DER), accuracy (ACC), slot-type F1 score, accuracy (ACC), accuracy (ACC), and BLEU score, respectively. To simulate training in the low-resource setting, for a given task we randomly sample 1%, 5% and 10% from the corresponding training set. The statistics of these data splits are reported in Table 2.

Models. We use three versions of Whisper encoders[3], including base.en, small.en and medium.en[4], denoted as Base, Small and Medium. Our baseline models are Wav2vec 2.0 Large[5] (W2V2) [3] and WavLM Large[6] (WavLM) [6]. We report the maximum number for training updates in Table 2. We use the SUPERB evaluation pipeline by *freezing* the encoders for downstream tasks while attaching a benchmark-specified lightweight prediction head for each task, unless mentioned otherwise. We adopt the identical training configuration (e.g., batch size, optimizer) for all models based on SUPERB hyperparameter settings. Experiments were done on 1xRTX 6000 GPU with 48GB Memory.

Table 2: *Training tasks' types and splits, and the corresponding training data sizes / cap on training updates.*

Task	Type	1%	5%	10%
SD	speaker	0.14k / 20k	0.70k / 20k	1.39k / 50k
SID	speaker	1.38k / 20k	6.92k / 50k	13.8k / 100k
SF	semantics	1.05k / 100k	5.24k / 100k	10.5k / 100k
IC	semantics	0.23k / 20k	1.16k / 20k	2.32k / 50k
KS	content	0.51k / 20k	2.56k / 50k	5.11k / 50k
ASR	content	0.28k / 50k	1.43k / 100k	2.86k / 200k
ST	semantics	2.88k / 32k	14.4k / 32k	28.8k / 32k

3.2. Main Results

We report results in Table 1. Overall, Whisper variants outperform W2V2 and WavLM on the majority of tasks in various data conditions with fewer updates except for SID. We summarise the findings for each task as follows:

- **IC** Various Whisper models exhibit a significantly better performance in all settings. Even with 1% of fine-tuning data, Base surpasses WavLM by 150% with a faster convergence rate. As the size of training data increases, Whisper on average converges 10× faster than W2V2.
- **SF** As data resources become more scarce, the benefits of

[2] https://superbbenchmark.org

[3] For brevity, we drop encoder when Whisper is mentioned.

[4] https://github.com/openai/whisper

[5] https://huggingface.co/facebook/wav2vec2-large-lv60

[6] https://huggingface.co/microsoft/wavlm-large

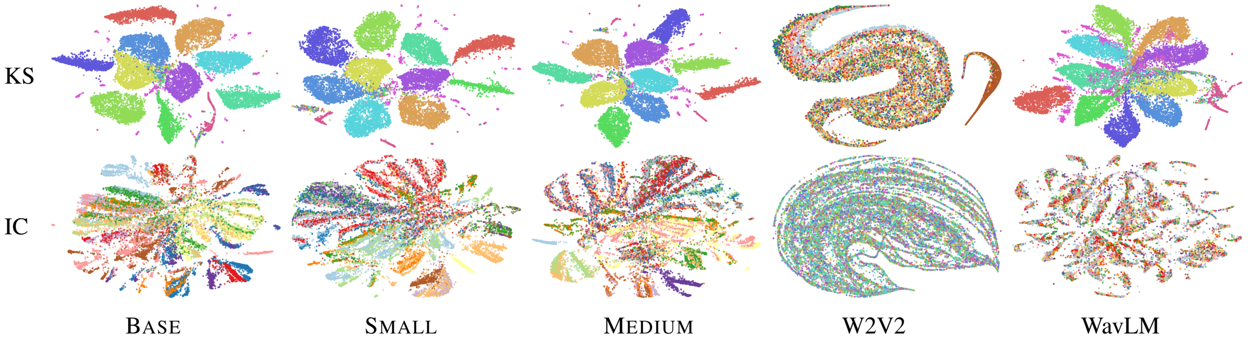

Figure 1: *t-SNE visualisation of the representation spaces produced by the encoders on KS (top) and IC (bottom) tasks training set (prior to fine-tuning) with colours indicating class labels.*

using Whisper become more eminent in training speed and performance. Notably, MEDIUM outperforms the baselines by a large margin with 50% less number of updates.

- **SD** MEDIUM significantly outperforms the baselines in all settings. BASE and SMALL (despite being 75% and 90% smaller) converge much faster with better or comparable performance compared to W2V2 and WavLM. Note that although SD is regarded as a speaker task, content information is still required as models tend to distinguish speaker timestamps by content, not just speaker features.
- **KS** Whisper dramatically boosts the performance with higher convergence speed. BASE and MEDIUM, fine-tuned on 1% and 5% of task data, surpass W2V2 (96.66) and WavLM (97.86) models that are fine-tuned on 100% of task data.[7]
- **ASR** Whisper models achieve robust performance at 1%, i.e., the extremely low data condition, with a few thousand updates. MEDIUM outperforms WavLM even though the latter was pre-trained on Librispeech and the former was not, whereas W2V2 has difficulties to converge. W2V2 and WavLM gradually pick up as the amount of training instances increases.
- **SID** Whisper models perform poorly on this task. Our hypothesis is that Whisper pre-training places emphasis on capturing content (via mapping audios to text) rather than speaker information and speech features that are important in SID (a speaker task). In contrast, W2V2 and WavLM, which are pre-trained only on speeches, are better positioned to tackle this task. We will unpack this hypothesis later.
- **ST** Whisper models do not perform well on translation, even with the increase in training corpus size. This could be due to replacing the internal Whisper decoder (i.e., used during the pre-training phase) with SUPERB's decoder. Nonetheless, Whisper still achieves the best performance at 1% and 5% compared to the baselines.

3.3. Analysis and Discussion

In this section, we start with a qualitative comparison of the pre-trained representations produced by Whisper variants, W2V2 and WavLM. We then measure the utilisation of the representation space through isotropy, and finish by investigating the information captured at different layers of these encoders.

t-SNE. We create t-SNE visualisations of the training data of KS and IC with the vanilla encoders, as shown in Figure 1. On KS, the embedding space of Whisper exhibits a better clustering of speech representations compared to W2V2 and WavLM, facilitating a much faster fine-tuning convergence and better task performance. This is less eminent on IC, although representations of Whisper are still better clustered than the baselines. This also explains why the performance of Whisper at 1% on IC is less remarkable compared with the KS task (but still far exceeds the baselines). Furthermore, the relatively tangled embedding spaces are partially due to the IC task having more classes than KS (31 vs. 12), which increases the overall task difficulty. We also produced the visualisation on SID, and observed a much worse clustering pattern compared with IC, explaining the weaker performance of Whisper on this task.

Isotropy. The geometry of representations generated by pre-trained Transformer have been shown to suffer from the anisotropy problem [21]. Ideally the representations should be uniformly distributed in a spherical space (isotropic), but in practice they only occupy narrow regions of the embedding space (anisotropic) [22]. The average isotropy scores of W2V2, WavLM, and Whisper (average of the three versions) on 7 downstream task datasets are 1e-300, 1e-14, and 1e-2, respectively. While being several orders of magnitude better than the other baselines, the Whisper's isotropy even approaches that of a text embedding space (i.e., compared with 1e-1 of the MirrorBert text encoder [23]). This indicates that Whisper largely mitigates the anisotropic problem that the other baselines face.

Task Fine-tuning. We adopt different strategies to generate speech representations. For the baselines, the weighted-sum of hidden states of each layer is considered as the feature for downstream task heads. For Whisper models, we considered only the last-layer output from the encoder[8] as speech representation for SUPERB downstream fine-tuning. We refer to this as the *Vanilla* configuration, which is our default in the previous sections. In Table 3, we compare the performance of the *Vanilla* setting with that of the *Weighted-sum* and *Fine-tuned* Whisper.[9] We observe that across most tasks the *Vanilla* version works best, while for the SID task, the *Weighted-sum* represen-

[7] The numbers are obtained from SUPERB Leaderboard at the time of writing this paper.

[8] The pre-training process of Whisper involves an encoder-decoder where the encoder provides the last-layer output to its decoder, but for the other baselines (which are encoder-only models), more emphasis is placed on the connection between the intermediate encoder layers, suggesting a higher gain for them to be achieved from aggregating information across several layers for downstream tasks.

[9] We freeze Whisper for *Weighted-sum* and unfreeze for *Fine-tuned*.

Table 3: *Task performance of Whisper models, under different settings of task fine-tuning: Vanilla (V) denotes the Whisper encoder is frozen and its last-layer output is used as the task feature. Weighted-sum (W) is similar to Vanilla except for the construction of task features for which the weighted-sum of each layer's hidden state is taken. Fine-tuned Whisper (F) is also similar to Vanilla, except the encoder is no longer frozen and is fine-tuned together with the downstream task. -: The Whisper* MEDIUM *could not be fine-tuned due to the limitation of GPU memory.*

		KS↑			IC↑			ASR↓			SID↑		
Tr.	Model	V	W	F	V	W	F	V	W	F	V	W	F
1%	BASE	**96.79**	94.87	93.12	**67.04**	34.06	41.92	**26.43**	31.97	62.00	2.66	**6.35**	5.61
	SMALL	96.62	95.81	**97.31**	57.63	33.54	**72.24**	**20.27**	25.71	44.68	3.35	**8.71**	7.32
	MEDIUM	**96.72**	95.52	-	**73.74**	34.19	-	**17.56**	24.56	-	3.97	**12.48**	-
5%	BASE	**97.44**	97.31	95.52	95.39	90.46	**96.28**	**16.18**	17.37	30.51	11.63	25.74	**25.90**
	SMALL	**97.73**	97.31	97.05	95.78	90.46	**95.86**	**11.76**	12.85	41.61	13.47	**36.90**	34.07
	MEDIUM	**97.95**	97.57	-	**98.23**	89.48	-	**9.75**	11.12	-	17.94	**45.22**	-
10%	BASE	**97.24**	**97.24**	95.88	96.92	90.43	**97.47**	**13.41**	13.59	25.11	19.48	41.20	**45.78**
	SMALL	**97.63**	97.53	97.37	96.44	93.33	**98.10**	**9.47**	10.09	38.03	23.04	**55.02**	53.55
	MEDIUM	**97.96**	97.54	-	**98.78**	95.28	-	**7.74**	8.52	-	30.05	**65.51**	-

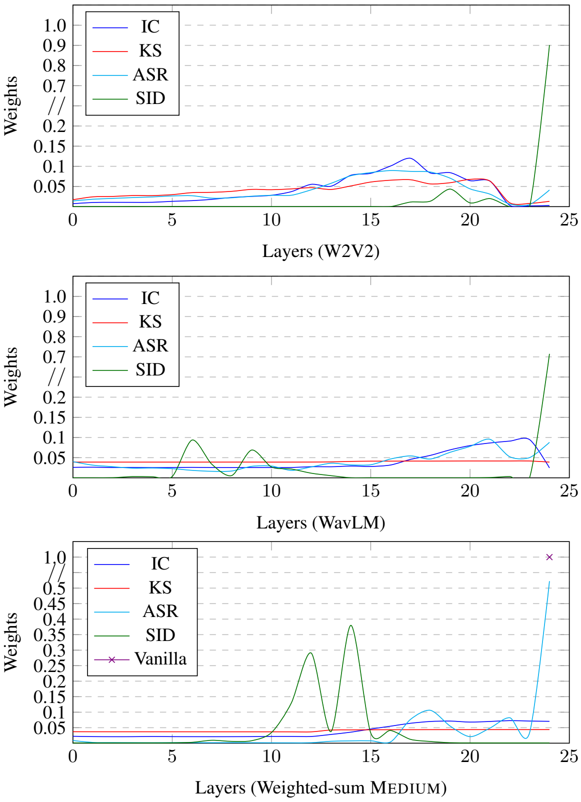

Figure 2: *The weight coefficients distribution of layers. The x-axis denotes different layers; the y-axis denotes the weight coefficients. Vanilla: the weight of vanilla Whisper on tasks.*

tations is a better choice suggesting that speaker information is retained in the intermediate layers of Whisper. In general, the *Fine-tuned* Whisper that uses the last-layer output as features underperforms the *Vanilla* variant with the frozen encoder, in the extreme low-data conditions (i.e., 1%). We speculate this occurs as the captured knowledge stored in Whisper will be disrupted after fine tuning the model on the small task data. As the size of training corpus increases, the trend continues for KS and ASR. For IC and SID, depending on the size of Whisper, *Fine-tuning* may surpass *Vanilla*.

Weight Coefficients Distribution. We visualise the distribution of the weight coefficients (i.e., signifying the contribution of the corresponding layer in task fine-tuning) attached to each layer of Transformer and learned during the fine-tuning step in Figure 2. The numbers are based on fine-tuning in the 10% training size. For Wav2vec2, WavLM and Whisper MEDIUM encoders on 4 tasks we observe different patterns of layer contribution layers. The Figure also reveal that the speech features are distributed in various layers of the encoders. An interesting pattern is SID which places more emphasis on the last layers of W2V2 and WavLM, but shifts that to the intermediate layers of Whisper, indicating a stronger presencce of speaker features in its intermediate layers. As expected we observe that for Whisper and ASR task, most of the importance is placed on the final layers of the encoder. This also verifies why the *Vanilla* configuration (which uses the last-layer as the task feature) is better at content-related tasks compared with a speaker task like SID.

Summary. We highlighted the quality of speech representations generated by Whisper. Compared to W2V2 and WavLM, the Whisper BASE and SMALL have notably much fewer parameters, less than 100M, which leads to faster training convergence and inference. The representations achieve state-of-art performance on several downstream tasks. Regarding specific tasks (refer to Table 3), we observed that the *Vanilla* Whisper for content tasks (ASR and KS) performs well in the very low-resource scenario. On IC, we observed that Whisper benefits more from fine-tuning with increasing number of training instances and model size. *Vanilla* did not do well overall on SID whereas the *Weighted-sum* improved the results substantially.

4. Conclusion

In this paper, we evaluated the performance of three widely used pre-trained speech encoders in the low-resource setting on 7 diverse speech tasks from the SUPERB and SUPERB-SG benchmarks. We analysed the generated speech representations, for their qualitative and quantitative properties. Additionally, we looked at the internal contribution of layers from these encoders in various downstream task settings. Our findings highlighted the superior capabilities of the recent Whisper model's encoder for most of the semantic-content tasks and its performance degradation on speaker-focused task. We established a connection between the pre-training protocol of these models and their representational properties, and their downstream task performance.

5. 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 *et al.*, "Self-supervised speech representation learning: A review," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[2] S. Liu, A. Mallol-Ragolta, E. Parada-Cabaleiro, K. Qian, X. Jing, A. Kathan, B. Hu, and B. W. Schuller, "Audio self-supervised learning: A survey," *Patterns*, vol. 3, no. 12, p. 100616, 2022. [Online]. Available: https://doi.org/10.1016/j.patter.2022.100616

[3] 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.

[4] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[5] 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.

[6] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *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.

[7] H. Yang, J. Zhao, G. Haffari, and E. Shareghi, "Self-supervised rewiring of pre-trained speech encoders: Towards faster fine-tuning with less labels in speech processing," in *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 1952–1959. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.141

[8] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[9] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018. [Online]. Available: https://arxiv.org/pdf/1807.03748

[12] S. Yang, P. Chi, Y. Chuang, C. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G. Lin, T. Huang, W. Tseng, K. Lee, D. Liu, Z. Huang, S. Dong, S. Li, S. Watanabe, A. Mohamed, and H. Lee, "SUPERB: speech processing universal performance benchmark," in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, and P. Motlícek, Eds. ISCA, 2021, pp. 1194–1198. [Online]. Available: https://doi.org/10.21437/Interspeech.2021-1775

[13] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, and M. Auli, "Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training," in *Proc. Interspeech 2021*, 2021, pp. 721–725.

[14] 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 *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 228–235.

[15] J. Shah, Y. K. Singla, C. Chen, and R. R. Shah, "What all do audio transformer models hear? probing acoustic representations for language delivery and its structure," *arXiv preprint arXiv:2101.00387*, 2021.

[16] 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.

[17] Y.-A. Chung, Y. Belinkov, and J. Glass, "Similarity analysis of self-supervised speech representations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 3040–3044.

[18] A. Pasad, B. Shi, and K. Livescu, "Comparative layer-wise analysis of self-supervised speech models," *arXiv preprint arXiv:2211.03929*, 2022.

[19] J. Zhao, H. Yang, G. Haffari, and E. Shareghi, "M-adapter: Modality adaptation for end-to-end speech-to-text translation," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022, pp. 111–115. [Online]. Available: https://doi.org/10.21437/Interspeech.2022-592

[20] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, "Transfer ability of monolingual wav2vec2.0 for low-resource speech recognition," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–6.

[21] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "A latent variable model approach to pmi-based word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 385–399, 2016.

[22] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 9119–9130.

[23] F. Liu, I. Vulić, A. Korhonen, and N. Collier, "Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 1442–1459.