SPEECH SEPARATION WITH LARGE-SCALE SELF-SUPERVISED LEARNING

Zhuo Chen, Naoyuki Kanda, Jian Wu, Yu Wu, Xiaofei Wang, Takuya Yoshioka,
Jinyu Li, Sunit Sivasankaran, Sefik Emre Eskimez

Microsoft, One Microsoft Way, Redmond, USA

ABSTRACT

Self-supervised learning (SSL) methods such as WavLM have shown promising speech separation (SS) results in small-scale simulation-based experiments. In this work, we extend the exploration of the SSL-based SS by massively scaling up both the pre-training data (more than 300K hours) and fine-tuning data (10K hours). We also investigate various techniques to efficiently integrate the pre-trained model with the SS network under a limited computation budget, including a low frame rate SSL model training setup and a fine-tuning scheme using only the part of the pre-trained model. Compared with a supervised baseline and the WavLM-based SS model using feature embeddings obtained with the previously released 94K hours trained WavLM, our proposed model obtains 15.9% and 11.2% of relative word error rate (WER) reductions, respectively, for a simulated far-field speech mixture test set. For conversation transcription on real meeting recordings using continuous speech separation, the proposed model achieves 6.8% and 10.6% of relative WER reductions over the purely supervised baseline on AMI and ICSI evaluation sets, respectively, while reducing the computational cost by 38%.

***Index Terms*—** WavLM, speech separation, conversation transcription, self-supervised learning, multi-speaker

1. INTRODUCTION

Speech separation (SS) has long been studied as an effective front-end method to handle overlapping speech and quick speaker transitions in conversations. Recently, there have been tremendous improvements in the SS model architecture [1, 2, 3, 4], enabling their successful applications to conversation transcription [5, 6, 7, 8].

Most SS models are trained by using overlapping speech signals that are artificially generated from clean speech data rather than using real noisy recordings. It is because the SS training requires clean signals as the training targets. Such a simulation-based training scheme faces two limitations. Firstly, although an unlimited amount of overlapped speech data can be generated by simulation, the available clean speech data are limited because they should not include background noise and reverberation to be used as the training targets. This results in inherently limited speech variations even after the simulation [9]. Secondly, the simulated far-field speech mixtures often exhibit a systematic mismatch with real recordings, leading to potentially sub-optimum separation performance on real data [10].

There were several attempts to leverage real noisy data for training SS models. In [9, 11], a combination-invariant loss function was proposed for training an SS model only from noisy audio. On the other hand, a series of studies on self-supervised learning (SSL) [12, 13, 14] showed that the SS quality could be significantly improved by leveraging the SSL pre-trained models. Particularly, WavLM [12] showed prominent improvement in SS, achieving the state-of-the-art performance in the LibriCSS benchmark [7]. However, the previous investigations had several limitations. Firstly, the amount of the pre-training data was limited only to 94K hours, which can be massively extended by using real-world recordings. Secondly, no consideration was made for the inference-time computational cost, where a naive application of the SSL pre-trained models often incurs an unacceptable increase in the inference cost. Thirdly, the past evaluation of SSL-based SS was only limited to simulation data, and the effectiveness for real recordings was unexplored.

In this work, we explore the SSL-based SS and its deployment in conversation transcription systems further with the dual objective of improving the SS performance and reducing the system's computation complexity. Our exploration is conducted based on WavLM [12] as a backbone of the SSL model. We first examine the impact of the pre-training data size by drastically increasing the pre-training data quantity for more than 300K hours. We also investigate various practical techniques to leverage the pre-trained model under the limited inference computation budget, including a low frame rate SSL training setup and a fine-tuning scheme using only the part of the pre-trained model. The SS network is integrated into a continuous SS (CSS)-based conversation transcription system and evaluated by using both simulation and real recordings from the AMI [15] and ICSI [16] meeting corpora.

2. RELATED WORKS

2.1. WavLM

WavLM is a self-supervised speech model that achieves the state-of-the-art performance in multiple downstream tasks [17, 12, 18]. WavLM follows the learning scheme of masked speech prediction (MSP). In this scheme, a pseudo label is first estimated for each frame of unlabeled audio. During training, a random mask is applied to each input audio, and the network is optimized to predict the pseudo label of the masked region, as shown in Fig. 1.

WavLM distinguishes itself from other MSP-based SSL models by introducing an "utterance mixing" procedure, where a training speech sample is randomly mixed with another one. This data augmentation forces the network to ignore interfering speakers and predict the labels from the main speakers. In this way, speaker discrimination is promoted in WavLM representations, making it beneficial for downstream tasks requiring speaker distinction, such as speaker verification, speaker diarization, and SS. The pseudo label is obtained by clustering unsupervised representations from the unlabeled data. Following [18], we adopt an online speaker clustering tokenizer in WavLM training.

2.2. CSS

CSS is introduced as a front-end to handle overlapping speech in long-form multi-talker recordings [19, 20, 21]. In the CSS framework, an input long-form audio is segmented into a set of over-

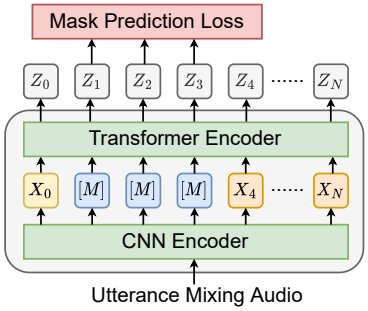

Fig. 1. WavLM network. X denotes the input sequence. M,Z are the masked frames and outputs. In training, the encoder inputs the full masked sequence and MSP loss is calculated on masked region

lapping chunks. Each chunk contains frames representing the history, current and future frames, denoted by T_h, T_c, and T_f, respectively. A local SS network processes each chunk to estimate N short overlap-free signals. N long-form overlap-free signals are then obtained by stitching the short overlap-free signals (i.e., the SS outputs) from each chunk. See [7] for more details.

3. METHOD

3.1. WavLM-based SS network

We train an SS network with WavLM by following the training scheme in [12]. Specifically, we first train WavLM by using unlabeled speech data. We then fine-tune the conformer-based SS network [7] by feeding a concatenation of an audio spectrogram and the WavLM embedding as an input. The WavLM embedding is computed by the weighted average of outputs from all transformer layers of WavLM, where the weights are learned during the fine-tuning. We either update or freeze the parameters of WavLM during the fine-tuning. The SS network is optimized by utterance-level permutation invariant training loss [22] with $N = 2$, where each output branch estimates a magnitude mask of each speaker. Fine-tuning is performed on the artificially mixed dataset, whose detail will be introduced in Sec. 4.2. During the inference, the output for each speaker is obtained through a speech masking [23].

3.2. Data scale up

Three data sets with different data scales are explored for WavLM training. We denote them by small (S), medium (M), and large (L) based on their data sizes. Data set S is the same data set used in the original WavLM paper [12] and consists of 94k hours of speech. The combination of S and the 220K-hour data from [24], totaling 314K hours, serves as data set M. Data set L is even larger than M. While we are not disclosing further details of this data set due to company's confidentiality policy, we believe that the experimental results with S and M should provide concrete insights regarding the impact of the data sale.

The training data, except the data S, were collected from an in-house dataset from a variety of data sources. Because most of the data more or less contain noise, reverberation, and distortion, they are not appropriate for the overlapped speech simulation for supervised SS training. In our preliminary experiments, we observed that a supervised SS model trained with the above-mentioned data showed significantly worse results than an SS model trained with our fine-tuning data generated from the clean signals (detailed later).

3.3. Inference cost reduction

To reduce the inference cost of the WavLM-based SS, we investigate three aspects of model configurations, namely, balancing the model sizes for WavLM and SS networks, low frame rate WavLM, and a partial usage of WavLM.

Table 1. Model configurations for SSL and SS networks.

SSL model	Params (M)	nLayers	nHeads	nDim	nFFdim
WavLM Small	53	12	12	384	1536
WavLM Base	90	12	12	768	3072
WavLM Large	300	24	16	1024	4096
SS model	**Params (M)**	**nLayers**	**nHeads**	**nDim**	**nFFdim**
SS-9.5	9.5	8	4	256	1024
SS-26	26	16	4	256	1024
SS-59	59	18	8	512	1024
SS-79	79	24	8	512	1024
SS-92	92	28	8	512	1024

3.3.1. Balancing model sizes for WavLM and SS networks

The model size configurations for the SSL pre-trained network and the SS network under a limited computation budget has not been sufficiently explored for SS. Generally speaking, a large SSL model offers a better generalization ability for downstream tasks, allowing a tiny SS network to be used. On the other hand, under the same computation budget, it can be the case where assigning a larger capacity to the SS network by using a small SSL model results in better SS performance.

Thus, we examine various combinations of WavLM and SS networks with different sizes as shown in Table 1. WavLM consists of a transformer [25] with relative position bias [26]. We examine three WavLM model configurations: Small, Base, and Large. The SS network consists of a conformer [7] by following [23]. We test five model configurations for the SS network as shown in Table 1. The model size varies from 9.5 million to 92 million parameters.

3.3.2. Low frame rate WavLM

In the WavLM-based SS framework, different frame rates can be used for the pre-trained and SS models. It is often beneficial to use a high frame rate (or equivalently, a small frame shift) for SS modeling [27] as it helps the SS model capture the fine-grained details of the speech. However, this results in an increase in the computational cost. Meanwhile, the symbolic representation prediction, as used by WavLM, can be modeled by using a relatively lower frame rate (or a larger frame shift) because the prediction targets do not change too frequently.

To reduce the computational cost needed for the WavLM embedding extraction, we examine the lower frame rate modeling, i.e., using a larger frame shift for the WavLM pre-trained models. In this work, a frame shift $f^{\text{ss}} = 10$ ms is used for the SS networks, while WavLM employs $f^{\text{wl}} = 20$ ms as the default frame shift. We also explore 30-ms and 40-ms frame shifts for WavLM to examine the possibility of further reducing the inference cost. During fine-tuning, to align the time sequence between the WavLM embedding and spectrogram, the WavLM embedding is duplicated by $f^{\text{wl}}/f^{\text{ss}}$ times before concatenation.

3.3.3. Higher WavLM layer removal

It is known that the contribution of each layer of an SSL model is different for each downstream task. Specifically, it is shown that the lower layers are more beneficial for speaker-related tasks such as SS and speaker diarization, while higher layers are essential for semantic tasks such as speech recognition [12].

Based on this observation, we explore fine-tuning the SS network with the WavLM embeddings that are computed by using only lower layers of the pre-trained models. This can significantly reduce

Table 2. Utterance-wise evaluation for data size and model configuration variation. During the fine-tuning, WavLM was frozen.

ID	SSL		SS	RTF	WER (%)	
	Model	Data			Far-mix	Clean-mix
B1	-	-	SS-59	$\times$ 0.21	22.7	22.7
B2	-	-	SS-79	$\times$ 0.27	23.2	23.8
B3	-	-	SS-92	$\times$ 0.32	23.1	23.6
P1	WavLM Large	S	SS-59	$\times$ 0.55	21.5	22.8
P2	WavLM Large	M	SS-59	$\times$ 0.55	20.6	18.2
P3	WavLM Large	L	SS-59	$\times$ 0.55	19.1	17.5
P4	WavLM Large	L	SS-26	$\times$ 0.47	19.2	20.0
P5	WavLM Base	L	SS-26	$\times$ 0.25	20.4	19.2
P6	WavLM Small	L	SS-26	$\times$ 0.20	20.2	20.2

Table 3. Utterance-wise evaluation for frame shift variation and partial layer fine-tuning (FT). WavLM was pre-trained by using the data L, and frozen during the fine-tuning.

ID	SSL			SS	RTF	WER (%)	
	Model	$f^{\text{wl}}(ms)$	FT-layers			Far-mix	Clean-mix
P3		20			$\times$ 0.55	19.1	17.5
L1	WavLM-Large	30	24	SS-59	$\times$ 0.46	21.9	24.8
L2		40			$\times$ 0.38	22.8	25.7
P4			24		$\times$ 0.47	19.2	20.0
S1			16		$\times$ 0.38	19.1	18.7
S2	WavLM-Large	20	12	SS-26	$\times$ 0.35	19.9	18.4
S3			8		$\times$ 0.31	19.7	18.6
S4			4		$\times$ 0.27	21.0	21.3

the inference cost. We examine various configurations, which will be presented in Section 5.

4. EXPERIMENT SETTINGS

4.1. Pre-training configuration

Various WavLM models as shown in Table 1 were trained using the data S, M or L (Section 3.2) by following the training recipe in [12]. Specifically, we optimized the network based on the masked speech prediction loss [28] while adding the noise or interference speaker speech to the input audio [12]. All WavLM models were trained on 32 NVIDIA A100 GPU cards. Adam optimizer with a peak learning rate of $5e^{-5}$ and a polynomial decay of 0.01 was used. The large and base model had the same batch size and training steps as in [12]. We maximized the batch size for the small model to fill the GPU memory and set the training steps to 600K.

4.2. Fine-tuning configuration

The SS network was fine-tuned by using an artificially generated overlapping speech. We used the procedures in [23] to generate training samples, where four types of mixing patterns are simulated: partially overlapped speech, fully overlapped speech, sequential two-speaker speech, and single-speaker. The source data for simulation is selected from public datasets including LibriSpeech [29], common voice [30], VCTK [31] and a Microsoft in-house corpus. For each training mixture, two clean utterances from different speakers are sampled and convolved with two artificially simulated room impulse responses using image method [32]. The reverberated utterances were then mixed with a random offset and a random source energy ratio between $-5 \sim 5$ dB. A background noise sampled from DNS challenge noise dataset [33] was then added to the reverberated mixture audio, with an SNR ranging from 10 dB $\sim$ 30 dB. As a result, 10K hours of overlapped speech was created as the fine-tuning data. A development set of 20 hours of speech was also generated from the same procedure, which was used for model selection.

Table 4. Utterance-wise evaluation for WavLM-based SS without and with additional fine-tuning without freezing WavLM. S3 model was used for comparison.

ID	Additional fine-tuning by unfreezing WavLM?	RTF	WER (%)	
			Far-mix	Clean-mix
S3	no	$\times$ 0.31	19.7	18.6
S3'	yes	$\times$ 0.31	19.3	18.0

We fine-tuned various SS networks listed in Table 1. The window size and shift for short-time Fourier transform (STFT) were set to 32 ms and 10 ms, respectively. The SS networks were optimized by the three-stage training as follows. We first shortly warmed up the SS network by training the model with artificially mixed data from VoxCeleb dataset [34, 35]. In this warm-up stage, we used the AdamW optimizer with a linear decay learning rate schedule, where the peak learning rate was set to $1e^{-4}$. We then further fine-tuned the model by using the 10K-hour data described above. During this stage, the WavLM parameters are frozen. The AdamW optimizer with a learning rate of $1e^{-3}$ and a linear decay is used for SS network training. Finally, we optionally continued fine-tuning with 10 times lower learning rate by unfreezing the WavLM network.

4.3. Evaluation scheme

Two evaluation schemes were used for assessing the proposed WavLM-based SS models, namely utterance-wise evaluation (Sec. 4.3.1) and continuous evaluation4.3.2.

4.3.1. Utterance-wise evaluation

The utterance-wise evaluation was performed on simulation-based test sets where each test sample was made by mixing two utterances. In this evaluation, an SS model was applied for each test sample without chunking the input, as is done for the CSS.

The SS models were evaluated based on the word error rate (WER) and real-time factor (RTF). For WER calculation, the SS model was first applied for each test sample to generate separated speech signals, and the ASR was then applied for each separated speech signal. We computed the WER for all permutations between the hypotheses and references, and selected the best WER among them. We used the latency-controlled long short-term memory-based hybrid ASR system from [6]. RTF was measured to assess the inference cost. For each model, we measured the computation time T_r for 2.4-sec audio and RTF was defined as $T_r/2.4$. The measurement was performed using one core and one thread from an INTEL i7-6700 3.4GHz CPU, averaging over 100 independent runs.

Two test sets named "far-mix" and "close-mix" were created for this evaluation. The far-mix test set was created by shifting and adding two far-field single-speaker utterances from real in-house meeting recordings, including the speech from 188 speakers. The overlap ratio was controlled to be 40%. The total word count for the "far-mix" set was 164,025. On the other hand, the close-mix test was similarly created by shifting and adding two close-microphone utterances from real in-house recordings, including the speech from 172 speakers. The total word count for the "close-mix" set was 47,206.

4.3.2. Continuous evaluation

The continuous evaluation was performed for real meeting recordings based on the CSS framework. We first applied the CSS with $T_h = 0.7$ sec, $T_c = 1.6$ sec and $T_f = 0.1$ sec, respectively, by using the WavLM-based SS as a local SS network. A single speaker merger was also applied in the CSS process to alleviate the distortion introduced for single speaker region [7], where the two outputting

Table 5. Utterance-wise evaluation for various models having similar computation cost as SS-59. Models are sorted in the order of RTF. The data L was used for SSL pre-training.

ID	SSL		SS	RTF	WER (%)	
	Model	FT-layers			Far-mix	Clean-mix
B1	-	-	SS-59	$\times$ 0.21	22.7	22.7
S3'	WavLM Large	8	SS-26	$\times$ 0.31	19.3	18.0
S4'	WavLM Base	12	SS-26	$\times$ 0.25	19.6	18.2
S5'	WavLM Base	4	SS-26	$\times$ 0.21	20.9	20.0
S6'	WavLM Small	12	SS-26	$\times$ 0.20	20.0	18.8
S7'	WavLM Small	8	SS-26	$\times$ 0.19	20.9	20.0
S8'	WavLM Base	12	SS-9.5	$\times$ 0.19	20.1	21.0
S9'	WavLM Small	8	SS-9.5	$\times$ 0.13	20.6	18.5

Table 6. Continuous evaluation for real meeting recordings. The data L was used for SSL pre-training.

ID	SSL		SS	RTF	AMI WER (%)		ICSI WER (%)	
	Model	FT-layers			dev	eval	dev	eval
B1	-	-	SS-59	$\times$ 0.21	21.6	25.0	23.2	20.7
S3'	WavLM Large	8	SS-26	$\times$ 0.31	19.1	22.6	17.8	16.5
S4'	WavLM Base	12	SS-26	$\times$ 0.25	19.4	22.9	18.6	17.2
S8'	WavLM Base	12	SS-9.5	$\times$ 0.19	19.5	22.9	18.0	17.0
S9'	WavLM Small	8	SS-9.5	$\times$ 0.13	19.6	23.3	18.3	18.5

channels are merged in detected single speaker regions. We then applied the same ASR used for the utterance-wise evaluation for the CSS-separated speech signals. We computed speaker-agnostic WER by using the multiple dimension Levenshtein edit distance calculation implemented in ASCLITE toolkit [36].

We used the recordings in the AMI [15] and ICSI [16] meeting corpora for this evaluation. For the AMI corpus, the first channel of the microphone array signals is used in the experiment. For the ICSI corpus, we used the recorded signal from the microphone with index "D2". Before sending it to the SS network, a causal logarithmic-loop-based automatic gain control is applied on both corpora. The evaluation was conducted based on the utterance-group segmentation [37], where a long-form recording is segmented at silence positions or non-overlapping utterance boundaries.

5. RESULTS AND DISCUSSIONS

5.1. Utterance-wise evaluation on simulated test sets

5.1.1. Impacts of data and model sizes

Table 2 shows the results of utterance-wise evaluation for various data and model configurations. We made the following observations from the results:

- All three models without WavLM (B1, B2, B3) showed similar WERs, indicating the saturation in their performances for our fine-tuning data.
- Significant and consistent WER reductions were observed by incorporating WavLM. For example, SS-59 trained with WavLM Large (P3) achieved 15.9% and 22.9% relative WER reductions over the best model without WavLM (B1) for far-mix and close-mix test sets, respectively.
- Increasing the amount of the pre-training data clearly benefited the ASR accuracy. The model pre-trained with the data L (P3) showed 11.2% and 23.2% WER reductions over the model pre-trained with the data S (P1) for far-mix and close-mix test sets, respectively. Generally, we observed a large improvement in the close-mix test set compared to the far-mix test set. It would be caused because our pre-training data mainly contained close microphone recordings.
- Increasing the model size of WavLM showed a clear advantage on the WERs over WavLM Base or Small, but with a cost of an increase of RTF (P4, P5, P6). On the other hand, increasing the model size of the SS network showed relatively less improvement on the WER. For example, the model with SS-26 (P4) showed only 0.1 points worse WER compared to the model with SS-59 (P3) for the far-mix test set.

Finally, it should be highlighted that SS-26 with WavLM-small (P6) achieved 11.0% and 11.0% WER reductions over the model without WavLM (B1) for far-mix and clean-mix test sets, respectively, while having a slightly smaller RTF.

5.1.2. Inference cost reduction

The utterance-wise evaluation results for using the low frame rate WavLM are shown in the upper half of Table 3. We observe that the low frame rate WavLM (L1, L2) showed significantly worse WERs compared to the WavLM with $f^{\mathrm{wl}} = 20$ msec (P3). This result suggests that preserving fine-grained details in audio signals is crucial not only for the SS task, but also for the SSL models.

We then evaluated the WavLM-based SS in which only a lower part of WavLM layers was used. The results are shown in Table 3. Unlike the low frame rate modeling, the usage of the lower part of WavLM showed a promising trade-off between RTF and WERs. For example, by using only the bottom 8 layers of WavLM (S3), the computation was reduced by 34% while the WER was increased only relative 2.5%. Interestingly, there was no WER improvement from the model using 16 layers (S1) to the model using all 24 layers (P4), suggesting that the representation in the higher layer is not beneficial for the SS task.

Table 4 shows the impact of additionally fine-tuning the WavLM-based SS by unfreezing the WavLM parameters. As shown in the table, we observed noticeable improvements for both far-mix and clean-mix. Given this result, we used the additional fine-tuning for all the remaining experiments.

In table 5, we further explored combinations of SSL model and SS networks. We collected model configurations that yielded similar inference costs as the baseline SS-59. From the table, we observed that all WavLM-based systems demonstrated considerable WER improvements over the pure-supervised baseline (B1) under a similar or even smaller RTF. Particularly, SS-9.5 trained with 8 layers of WavLM Small (S9') enjoyed 38% less RTF (0.21 RTF to 0.13 RTF), while achieving 9.3% and 18.5% of relative WER reduction over the B1 system for far-mix and clean-mix test sets, respectively.

5.2. Continuous evaluation on real meeting recordings

Table 6 shows the continuous evaluation for real meeting recordings. We selected representative configurations obtained in the previous experiment. Here, all WavLM-based SS models again yielded significant WER reduction in all test sets compared to the purely supervised baseline (B1). It should be highlighted that the S9' model achieved 6.8% and 10.6% WER reduction over the B1 model for the AMI and ICSI evaluation set, respectively.

6. CONCLUSIONS

In this work, we conducted a comprehensive investigation of WavLM-based SS by massively scaling up both the pre-training data and fine-tuning data, and evaluated the model from both the WER and RTF perspectives. The proposed model finally achieved 6.8% and 10.6% of relative WER reductions over the purely supervised baseline on AMI and ICSI evaluation sets, respectively, while reducing the computational cost by 38%.

7. REFERENCES

[1] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. ICASSP*, 2020, pp. 46–50.

[2] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016, pp. 31–35.

[3] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi *et al.*, "Attention is all you need in speech separation," in *Proc. ICASSP*, 2021, pp. 21–25.

[4] Z.-Q. Wang, S. Cornell, S. Choi *et al.*, "TF-GridNet: Making time-frequency domain models great again for monaural speaker separation," *arXiv preprint arXiv:2209.03952*, 2022.

[5] T. von Neumann, K. Kinoshita, M. Delcroix *et al.*, "All-neural online source separation, counting, and diarization for meeting analysis," in *Proc. ICASSP*, 2019, pp. 91–95.

[6] T. Yoshioka, I. Abramovski, C. Aksoylar, Z. Chen *et al.*, "Advances in online audio-visual meeting transcription," in *Proc. ASRU*, 2019, pp. 276–283.

[7] S. Chen, Y. Wu, Z. Chen *et al.*, "Continuous speech separation with Conformer," in *Proc. ICASSP*, 2021, pp. 5749–5753.

[8] A. Sivaraman, S. Wisdom *et al.*, "Adapting speech separation to real-world meetings using mixture invariant training," in *Proc. ICASSP*, 2022, pp. 686–690.

[9] S. Wisdom, E. Tzinis *et al.*, "Unsupervised sound separation using mixture invariant training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3846–3857, 2020.

[10] X. Wang, D. Wang, N. Kanda, S. E. Eskimez *et al.*, "Leveraging real conversational data for multi-channel continuous speech separation," *Proc. Interspeech*, 2022.

[11] E. Tzinis, Y. Adi *et al.*, "RemixIT: Continual self-training of speech enhancement models via bootstrapped remixing," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[12] S. Chen, C. Wang, Z. Chen, Y. Wu *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[13] H.-S. Tsai, H.-J. Chang *et al.*, "SUPERB-SG: Enhanced speech processing universal performance benchmark for semantic and generative capabilities," *Proc. ACL*, 2022.

[14] Z. Huang, S. Watanabe, S.-w. Yang, P. García *et al.*, "Investigating self-supervised learning for speech enhancement and separation," in *Proc. ICASSP*, 2022, pp. 6837–6841.

[15] J. Carletta, S. Ashby *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[16] A. Janin, D. Baron, J. Edwards, D. Ellis *et al.*, "The ICSI meeting corpus," in *Proc. ICASSP*, vol. 1, 2003, pp. I–I.

[17] Z. Chen, S. Chen, Y. Wu *et al.*, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *Proc. ICASSP*, 2022, pp. 6147–6151.

[18] S. Chen, Y. Wu, C. Wang, S. Liu *et al.*, "Why does self-supervised learning for speech recognition benefit speaker recognition?" *Proc. Interspeech*, 2022.

[19] T. Yoshioka, H. Erdogan, Z. Chen *et al.*, "Multi-microphone neural speech separation for far-field multi-talker speech recognition," in *Proc. ICASSP*, 2018, pp. 5739–5743.

[20] T. Yoshioka, Z. Chen, C. Liu, X. Xiao *et al.*, "Low-latency speaker-independent continuous speech separation," in *Proc. ICASSP*, 2019, pp. 6980–6984.

[21] T. Yoshioka, H. Erdogan *et al.*, "Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks," in *Proc. Interspeech*, 2018, pp. 3038–3042.

[22] D. Yu, M. Kolbæk, Z.-H. Tan *et al.*, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017, pp. 241–245.

[23] J. Wu, Z. Chen, S. Chen, Y. Wu *et al.*, "Investigation of practical aspects of single channel speech separation for ASR," *Proc. Interspeech*, 2021.

[24] C. Wang, Y. Wu, S. Liu, J. Li *et al.*, "UniSpeech at scale: An empirical study of pre-training method on large-scale speech recognition dataset," *arXiv preprint arXiv:2107.05233*, 2021.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[26] Z. Chi, S. Huang, L. Dong *et al.*, "XLM-E: Cross-lingual language model pre-training via electra," in *Proc. ACL*, 2022, pp. 6170–6182.

[27] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM trans. on ASLP*, vol. 27, no. 8, pp. 1256–1266, 2019.

[28] W.-N. Hsu, B. Bolte *et al.*, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. on ASLP*, 2021.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[30] R. Ardila, M. Branson, K. Davis, M. Kohler *et al.*, "Common Voice: A massively-multilingual speech corpus," in *Proc. LREC*, 2020, pp. 4218–4222.

[31] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *University of Edinburgh. CSTR*, 2017.

[32] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *JASA*, vol. 65, pp. 943–950, 1979.

[33] C. K. Reddy, V. Gopal *et al.*, "The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," *Proc. Interspeech*, 2020.

[34] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," *Proc. Interspeech*, pp. 2616–2620, 2017.

[35] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," *Proc. Interspeech*, pp. 1086–1090, 2018.

[36] J. G. Fiscus, J. Ajot *et al.*, "Multiple dimension Levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech," in *Proc. LREC*, 2006, pp. 803–808.

[37] N. Kanda, G. Ye *et al.*, "Large-scale pre-training of end-to-end multi-talker ASR for meeting transcription with single distant microphone," *Proc. Interspeech*, pp. 3430–3434, 2021.