



Enhancing Target-speaker Automatic Speech Recognition Using Multiple Speaker Embedding Extractors with Virtual Speaker Embedding

Ju-Seok Seong, Jeong-Hwan Choi, Ye-Rin Jeoung, Ilseok Kim, Joon-Hyuk Chang

Department of Electronic Engineering, Hanyang University, Seoul, Republic of Korea

as2835510@hanyang.ac.kr, brent1104@hanyang.ac.kr, jyr0328@hanyang.ac.kr,
maybe911@hanyang.ac.kr, jchang@hanyang.ac.kr

Abstract

Target-speaker automatic speech recognition (TS-ASR) utilizes speaker embeddings to identify a target speaker in multi-talker environments. While high-performance speaker embedding extractors provide discriminative embeddings, their computational demands limit practical deployment. In this study, we present two novel methods that effectively utilize lightweight extractors to enhance TS-ASR performance. First, we propose a multiple embeddings modulation that effectively transfers comprehensive speaker information to the ASR module, thereby improving overall performance and robustness against embedding variations. Second, we present a virtual speaker embedding augmentation technique that synthesizes embeddings of unseen speakers, reducing dependence on specific extractors while enhancing independent contributions from each extractor. Experimental results on the Libri2Mix dataset demonstrate that our proposed methods achieve significant WER reductions compared to the baseline model.

Index Terms: Target-speaker automatic speech recognition, speaker embedding, lightweight speaker embedding extractor, virtual speaker embedding

1. Introduction

Automatic speech recognition (ASR) under single-talker conditions has shown significant improvements in performance [1, 2, 3]. However, ASR still remains a challenging task in multi-talker environments since its performance is severely degraded by interfering speech from other speakers [4, 5]. To address this issue, target-speaker ASR (TS-ASR) technology [6, 7, 8] has emerged as a promising solution that aims to recognize the target speaker’s speech while minimizing the impact of interfering speakers in multi-talker environments. This technology shows great potential for enhancing the user experience across various applications, particularly in crowded environments and voice-controlled devices such as smart cars, mobile devices, and smart home devices.

The most common approaches to the TS-ASR can be categorized as cascade-based [6] and end-to-end (E2E)-based [7, 8] methods. Cascade-based methods, which have been widely adopted due to individual module performance and intuitive roles, typically consist of a target speaker extraction (TSE) module as the front-end and an ASR module as the back-end. However, these approaches have significant limitations. Achieving optimal system performance is challenging because minimizing the loss of the TSE module does not directly correlate with ASR performance improvements, and the TSE module requires additional labels during training. E2E-based methods have recently gained attention as they demonstrate superior performance. These methods address the limitations by eliminat-

Table 1: Evaluation results for speaker recognition models: Speaker verification (SV) performance on VoxCeleb1-O trial set, reported as equal error rate (EER, %) and minimum Detection Cost Function (minDCF). TS-ASR performance on dev-mix-clean (dev) and test-mix-clean (test) subsets of Libri2Mix, reported as word error rate (WER, %). The ↓ symbol indicates that a lower value represents a better metric.

Model	#Params	SV		TS-ASR [†]	
		EER ↓	minDCF ↓	dev ↓	test ↓
ResNet293 [9, 10]	28.6 M	0.51	0.0588	12.67	12.14
ECAPA2 [11]	27.1 M	0.43	0.0408	12.82	11.83
Thin ResNet-34 [12]	1.4 M	2.23	0.2848	13.61	13.47
Fast ResNet-34 [12]	1.4 M	2.27	0.2979	13.45	14.06
ReDimNet-B0 [13]	1.0 M	1.16	0.1013	13.17	13.20

[†]TS-ASR model specifications are detailed in 3.1.

ing the need for additional labels through direct optimization of target speaker transcription prediction, resulting in enhanced robustness in multi-speaker scenarios.

TS-ASR systems typically utilize speaker embeddings that are capable of discriminating among different speakers [7, 8]. In this approach, the quality of speaker embedding significantly impacts system performance to transcribe the speech of the target speaker. Table 1 presents an analysis of both speaker verification (SV) and TS-ASR performance across different speaker extractor models. We evaluate two categories of extractors with two large-scale models containing over 25M parameters and three lightweight models with less than 2M parameters. The experimental results demonstrate that larger models not only exhibit superior SV performance compared to lightweight models but also achieve enhanced TS-ASR performance. However, larger-scale models require substantial computational resources to enroll new target speakers, making their deployment in real-world applications challenging.

In this paper, we analyze the impact of speaker embeddings in TS-ASR systems and propose methods to enhance their utilization during training. Furthermore, we introduce novel approaches that maximize performance using lightweight speaker embedding extractor models, offering an efficient solution for real-world applications.

2. Proposed methods

We propose multiple speaker embeddings target-speaker ASR (MUSE-TSASR), a novel framework that incorporates two efficient methods to enhance performance using lightweight speaker embedding extractors. First, we introduce a multiple speaker embeddings (MUSE) modulation layer that effec-

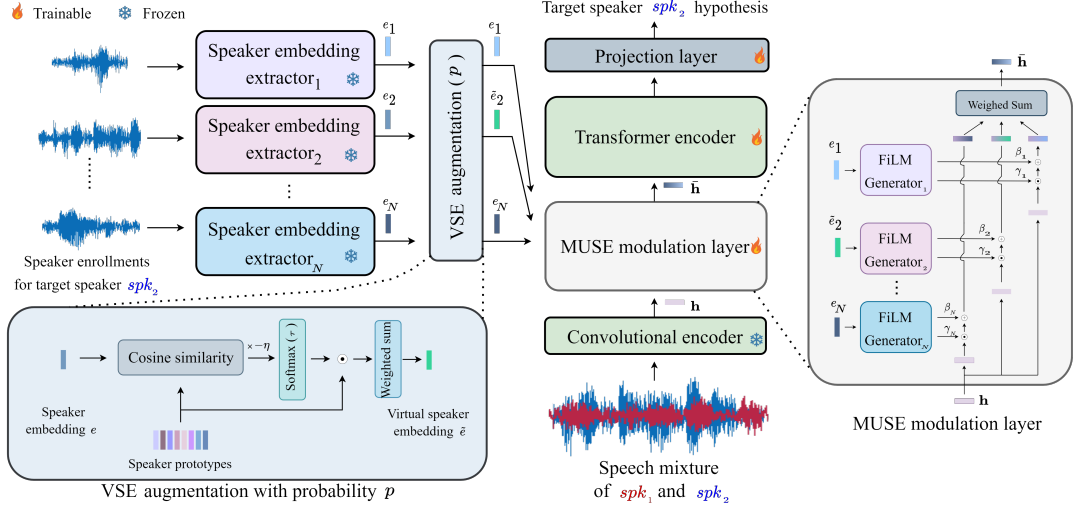


Figure 1: Illustration of the proposed methods. As shown in the bottom center of the figure, the input speech consists of overlapped utterances from both the target speaker spk_2 and an interference speaker spk_1 . As illustrated on the upper left side of the figure, multiple lightweight speaker embedding extractors extract speaker embeddings that are then fed into the MUSE modulation layer together with the convolutional encoder output $\mathbf{h}$. Each speaker embedding undergoes virtual speaker embedding augmentation with probability p , shown here, applied only to e_2 . Then, the model transcribes the target speaker’s speech while ignoring speech from other speakers. The dot symbol denotes the Hadamard product. This figure is best viewed in color.

tively integrates information from various speaker embedding extractors. Based on the standard Transformer architecture [14], which consists of a convolutional encoder and a transformer encoder, we incorporate the MUSE modulation layer between these two encoders, as illustrated in the middle and right of Fig 1. In this layer, we employ the Feature-wise Linear Modulation (FiLM) method [15] to utilize each speaker embedding as conditioning information. Second, as depicted in the left side of Fig 1, we present a virtual speaker embedding (VSE) augmentation technique that synthesizes embeddings of unseen speakers. This approach mitigates over-confidence in each extractor while maximizing the utilization of distinctive features captured by each extractor, thereby improving overall performance and enhancing cross-embedding robustness.

2.1. Feature-wise linear modulation

FiLM [8, 16] performs a straightforward and feature-wise affine transformation that adaptively modulates the intermediate features of a neural network based on conditioning input. The model learns two mapping functions $f, h : \mathbb{R}^d \rightarrow \mathbb{R}^k$ that transform the speaker embedding $e \in \mathbb{R}^d$ as follows:

$$\gamma = f(e) \quad \beta = h(e), \quad (1)$$

where $\gamma \in \mathbb{R}^k$ and $\beta \in \mathbb{R}^k$ modulate the representation $\mathbf{h} \in \mathbb{R}^k$ as follows:

$$\begin{aligned} \bar{\mathbf{h}} &= \text{FiLM}(\mathbf{h} \mid \gamma, \beta) \\ &= \gamma \odot \mathbf{h} + \beta, \end{aligned} \quad (2)$$

where $\odot$ is the element-wise (Hadamard) product. Both γ and β are the learnable scaling and shifting parameters, respectively.

2.2. Multiple speaker embeddings modulation

The proposed architecture leverages N distinct speaker embedding extractors, where each extractor generates a distinct

speaker embedding with different dimensions. Specifically, for the n -th extractor, the embedding $e_n \in \mathbb{R}^{d_n}$ has dimension d_n , where d_n may vary across extractors. These embeddings jointly serve as conditioning information for the target speaker. We extend Eq. (1) by implementing specialized functions $f_n, h_n : \mathbb{R}^{d_n} \rightarrow \mathbb{R}^k$ for each n -th embedding, defined as follows:

$$\gamma_n = f_n(e_n) \quad \beta_n = h_n(e_n). \quad (4)$$

The resulting $\gamma_n \in \mathbb{R}^k$ and $\beta_n \in \mathbb{R}^k$ are then utilized to modulate the representation $\mathbf{h}$ through the following equations:

$$\bar{\mathbf{h}} = \sum_{n=1}^N \text{FiLM}(\mathbf{h} \mid \gamma_n, \beta_n), \quad (5)$$

$$= \sum_{n=1}^N \gamma_n \odot \mathbf{h} + \sum_{n=1}^N \beta_n. \quad (6)$$

This MUSE modulation layer enables the model to simultaneously condition its representations on multiple speaker embeddings, capturing diverse speaker characteristics and enhancing the overall system performance.

To enhance system performance, we expand the embedding space diversity by utilizing distinct enrollment utterances for each extractor, as depicted on the left side of Fig. 1. This methodology significantly increases the potential combinations of mixed speech inputs and speaker embeddings during the training phase, leading to improved model robustness and generalization capabilities. During evaluation, all extractors utilize the same enrollment utterance to extract embeddings.

2.3. Virtual speaker embedding augmentation

To mitigate dependency on specific speaker embedding extractors, we propose a VSE augmentation method. This augmentation utilizes speaker prototypes from the training data. As illustrated on the lower left side of Fig 1, the virtual embedding is

synthesized through a weighted combination of the speaker prototypes. Let $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$ denote the set of N speakers in the training dataset. For each speaker s_i , we compute a speaker prototype $p_i \in \mathbb{R}^d$ by averaging all embeddings associated with s_i . The virtual speaker embedding generation process consists of three main steps.

1. **Dissimilarity Computation:** Given an input speaker embedding $e \in \mathbb{R}^d$, we compute the cosine similarity with all prototypes:

$$\text{sim}(e, p_j) = \frac{e \cdot p_j}{|e||p_j|}, \quad \forall j \in 1, \dots, N. \quad (7)$$

Then, we apply a negative scaling factor $-\eta$ to these similarities to calculate dissimilarity scores.

2. **Probability Weighting:** The negatively scaled similarities are then transformed into probability weights through a temperature-controlled softmax function:

$$\alpha_j = \frac{\exp(-\eta \text{sim}(e, p_j)/\tau)}{\sum_{k=1}^N \exp(-\eta \text{sim}(e, p_k)/\tau)}, \quad (8)$$

where the temperature parameter τ controls the concentration of the distribution.

3. **Virtual Embedding Synthesis:** Finally, we synthesize the virtual speaker embedding $\tilde{e} \in \mathbb{R}^d$ as a weighted sum of prototypes as follows:

$$\tilde{e} = \sum_{j=1}^N \alpha_j p_j. \quad (9)$$

During training, we apply VSE augmentation probabilistically, where the virtual embedding $\tilde{e}$ is synthesized with probability p , while retaining the original embedding e with a probability $1 - p$.

3. Experimental setups

3.1. Model architecture and implementation details

We employed the WavLM Base+ model [17] as our upstream model, which consists of a CNN encoder followed by a Transformer encoder. The CNN encoder consists of seven temporal convolution blocks that down-sample the input by a factor of 320. The subsequent Transformer encoder comprises 12 Transformer blocks, each of which has 768-dimensional hidden states, 8 self-attention heads, and a 3,072-dimensional feed-forward layer. In addition, gated relative position bias [17] was used in the self-attention layers. The WavLM Base+ was pre-trained on 94k hours of large-scale data, including LibriLight [18], GigaSpeech [19], and VoxPopuli [20]. These diverse datasets encompass various backgrounds, contents, and many speakers. Furthermore, the training process incorporated simulated noisy and overlapped speech, making the model particularly suitable for multi-talker downstream tasks [17, 21].

For the speaker embedding extractors, we employed ResNet-293 [9, 10], ECAPA2 [11], Thin ResNet-34 [12], Fast ResNet-34 [12], and ReDimNet-B0 [13]. We utilized their pre-trained models from their public repositories [9, 10, 11, 12, 13] and fine-tuned using Large Margin fine-tuning (LM-FT) [22] on the LibriSpeech [23] train-clean-100 subset, which consists of 100 hours of speech data from 251 speakers. The LM-FT was conducted for a total of 7 epochs with an additive angular margin softmax [24] with a margin set to 0.4 and the

Table 2: *Experimental evaluation and comparative analysis of our proposed approaches against state-of-the-art methods on the Libri2Mix dataset, using both dev-mix-clean (dev) and test-mix-clean (test) subsets. We exclude results with the JSM module as described in [8] for the assumption that the interference speaker identity is unknown during inference.*

Training Method	WER (%)	
	dev	test
Whisper-m + LoRA [28]	-	11.98
WavLM base + CLN [8]	11.04	12.32
MUSE-TSASR (group 1)	11.52	11.19
MUSE-TSASR (group 2)	10.95	10.53
without VSE augmentation	12.31	12.06
without MUSE modulation	13.17	13.20

length of the duration set to 6 seconds. The embedding dimensions d_n of each model are 256, 192, 512, 512, and 192, respectively. Functions f_n and h_n were implemented as single linear layers that project these embeddings into a 768-dimensional representation. Both learnable parameters γ_n and β_n are 768 dimensional vectors. In contrast to our approach, the baseline TS-ASR model utilized a single extractor model and was constructed using Eqs. (1)–(3). Due to space constraints, detailed specifications of each speaker extractor model can be found in the original publications [9, 10, 11, 12, 13].

During the downstream task, the CNN encoder and speaker embedding extractors were frozen. The model was optimized using character-level CTC loss with Adam optimizer [25] using a peak learning rate of $3e-5$. During training, we applied VSE augmentation with probability $p = 0.1$ and $\eta = 1$. The experiments were conducted using the Fairseq toolkit [26] without incorporating an external language model.

3.2. Dataset

For the TS-ASR downstream task, we utilized the LibriMix [27] dataset, which contains multi-speaker overlapped speech generated from the LibriSpeech corpus. The model was fine-tuned on the train-100-mix-clean subset of the Libri2Mix, which is generated under 2-speakers, 16 kHz, and max conditions based on the LibriSpeech train-clean-100 dataset. It consists of 13,900 utterances (251 speakers, 58 hours). We selected three utterances to enroll for each speaker from the train-clean-100 dataset.

For evaluation, we used both dev-mix-clean (dev) and test-mix-clean (test) subsets. Each subset consists of 3,000 utterances from 40 speakers, comprising approximately 11 hours of simulated speech data. For a fair comparison analysis in Table 2, we utilized the publicly available enrollment dataset¹ as in [8, 28]. We used this enrollment dataset for all experiments presented in the subsequent tables. For each speaker, the enrollment dataset contains a single 15-second utterance collected from the LibriLight project, excluding the LibriSpeech portion.

¹https://github.com/HuangZiliAndy/SSL_for_multitalker

Table 3: Performance comparison of different configurations. WER(%) results for TS-ASR with MUSE layer incorporating VSE augmentation method evaluated across different temperature values.

Training method	WER (%)	
	dev	test
TS-ASR with MUSE layer	12.31	12.06
with VSE ($\tau = 10.0$)	12.17	11.54
with VSE ($\tau = 5.0$)	11.54	11.32
with VSE ($\tau = 1.0$)	11.12	10.97
with VSE ($\tau = 0.1$)	10.95	10.53
with VSE ($\tau = 0.01$)	11.02	10.72

Table 4: WER(%) comparison of models with and without VSE training technique. #Num indicates the number of speaker embedding extractors used during the evaluation. All models were trained with group 2 configuration.

Training method	#Num	WER (%)	
		dev	test
MUSE-TSASR (group 2)	2	12.12 $\pm$ 0.25	11.87 $\pm$ 0.16
without VSE augmentation	2	17.66 $\pm$ 0.93	16.38 $\pm$ 0.27
MUSE-TSASR (group 2)	1	14.92 $\pm$ 0.08	15.70 $\pm$ 0.19
without VSE augmentation	1	24.44 $\pm$ 0.73	22.18 $\pm$ 0.42

4. Results

4.1. Performance evaluation and analysis

Table 2 provides a comparative analysis of our proposed MUSE-TSASR against current state-of-the-art methods [8, 28]. For the implementation of MUSE-TSASR, we investigated two variant groups of lightweight embedding extractors. The first group, designated as group 1, comprised Thin ResNet-34, Fast ResNet-34, and ReDimNet-B0, with a total parameter count of 3.4M. The second configuration, designated as group 2, utilized three ReDimNet-B0 models with a total of 3M parameters. With the temperature parameter set to $\tau = 0.1$, our experimental results demonstrated that while both groups achieved competitive performance, the speaker embedding extractor from group 2 yielded superior results. Specifically, compared to the system that used only ReDimNet-B0 shown in Table 1, we achieved performance improvements of 16.85% and 20.22% on the dev and test sets, respectively. Our ablation study, which sequentially removed each proposed method, revealed substantial performance degradation, thereby demonstrating the significant contribution of our proposed methods to the overall system performance.

4.2. Ablation study with VSE augmentation

Table 3 presents the experimental results obtained by varying the temperature parameter τ in Eq. (8). We conducted experiments with values ranging from 0.01 to 10, where decreasing τ makes distribution sharper and includes a higher dissimilarity prototype with the embedding. Our experiments demonstrated optimal performance at $\tau = 0.1$, which we subsequently maintained as the fixed parameter for all experiments. We experimented with an alternative approach of masking embeddings with zero values for each extractor. However, this method demonstrated limited effectiveness, yielding WER of 12.11% and 11.73% on dev and test sets, respectively.

In Table 4, we evaluated the model performance based on varying numbers of speaker embedding extractors during the

Table 5: Performance evaluation of TS-ASR and MUSE-TSASR models using addition (Add) and concatenation (Concat) conditioning methods. Information on the utilized speaker embedding extractors is provided in parentheses.

Method	Condition	WER (%)	
		dev	test
TS-ASR (ReDimNet-B0)	Add	14.14	14.28
MUSE-TSASR (group 2)	Add	12.30	10.98
TS-ASR (ReDimNet-B0)	Concat	14.42	13.58
MUSE-TSASR (group 2)	Concat	11.84	10.88

evaluation, while three extractors were used during training. We conducted a comprehensive evaluation across all possible extractor combinations and reported both the mean performance and standard deviation for these configurations. Our analysis revealed that models trained without the VSE technique showed significant performance degradation when the number of extractors was reduced. In contrast, the MUSE-TSASR model trained with the VSE technique showed more robust performance even when utilizing only a single extractor. The results conclusively showed that the VSE technique enables robust model training that is less dependent on specific extractors.

4.3. Ablation study with different conditioning methods

In Table 5, we evaluated our proposed method using both addition and concatenation conditioning approaches. For the addition-based method, we utilized linear layers to match the dimension of representation $\mathbf{h}$ for each e_n . In the concatenation-based approach, we concatenated all e_n along the feature axis and utilized a single linear layer to match the dimensions with $\mathbf{h}$. Our proposed MUSE-TSASR (group 2) demonstrated superior performance compared to the baseline TS-ASR model in both conditioning methods. Specifically, with the addition-based method, MUSE-TSASR achieved WER reductions of 13.01% and 23.11% on the dev and test sets, respectively. Similarly, with the concatenation-based method, we observed WER reductions of 17.89% and 19.88% on the dev and test sets. These results consistently demonstrated that our MUSE-TSASR training method significantly enhances performance regardless of the conditioning method used.

5. Conclusion

In this paper, we analyzed the impact of speaker embeddings in TS-ASR systems and proposed methods to enhance their utilization during training. We introduced two effective methods for improving TS-ASR performance. First, we proposed a modulation method that jointly conditions hidden representation using multiple lightweight extractor models. Second, we introduced a virtual speaker embedding augmentation technique that enhances independent contributions from each extractor. Notably, our proposed methods achieved competitive performance while utilizing significantly fewer parameters through lightweight speaker extractors. Furthermore, experimental results demonstrated that our proposed methods consistently achieved significant improvements across various conditioning methods.

6. Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2025-00557944)

7. References

- [1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4960–4964.
- [2] J.-S. Seong, J.-H. Choi, J. Kyung, Y.-R. Jeoung, and J.-H. Chang, “Noise-Aware Target Extension with Self-Distillation for Robust Speech Recognition,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023.
- [3] D. Amodei *et al.*, “Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 173–182.
- [4] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, “Recognizing Overlapped Speech in Meetings: A Multichannel Separation Approach Using Neural Networks,” in *Proc. INTERSPEECH*, 2018, pp. 3038–3042.
- [5] N. Kanda *et al.*, “Auxiliary Interference Speaker Loss for Target-Speaker Speech Recognition,” in *Proc. INTERSPEECH*, 2019, pp. 236–240.
- [6] K. Žmolíková *et al.*, “SpeakerBeam: Speaker Aware Neural Network for Target Speaker Extraction in Speech Mixtures,” *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 4, pp. 800–814, 2019.
- [7] T. Moriya, H. Sato, T. Ochiai, M. Delcroix, and T. Shinozaki, “Streaming Target-Speaker ASR with Neural Transducer,” in *Proc. INTERSPEECH*, 2022, pp. 2673–2677.
- [8] Z. Huang, D. Raj, P. García, and S. Khudanpur, “Adapting Self-Supervised Models to Multi-Talker Speech Recognition Using Speaker Embeddings,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023.
- [9] H. Wang *et al.*, “Wespeaker: A Research and Production Oriented Speaker Embedding Learning Toolkit,” in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2023.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [11] J. Thienpondt and K. Demuynck, “ECAPA2: A Hybrid Neural Network Architecture and Training Strategy for Robust Speaker Embeddings,” in *Proc. IEEE Workshop Autom. Speech Recognit. Underst.*, 2023.
- [12] J. S. Chung *et al.*, “In Defence of Metric Learning for Speaker Recognition,” in *Proc. INTERSPEECH*, 2020, pp. 2977–2981.
- [13] I. Yakovlev *et al.*, “Reshape Dimensions Network for Speaker Recognition,” in *Proc. INTERSPEECH*, 2024, pp. 3235–3239.
- [14] A. Vaswani *et al.*, “Attention is All you Need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
- [15] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, “FiLM: Visual Reasoning with a General Conditioning Layer,” in *Proc. AAAI Conf. AI*, vol. 32, no. 1, 2018.
- [16] J. Pilault, A. Elhattami, and C. Pal, “Conditionally Adaptive Multi-Task Learning: Improving Transfer Learning in NLP Using Fewer Parameters & Less Data,” in *Proc. Int. Conf. Learn. Represent.*, 2021.
- [17] S. Chen *et al.*, “WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing,” *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [18] J. Kahn *et al.*, “Libri-Light: A Benchmark for ASR with Limited or No Supervision,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 7669–7673.
- [19] G. Chen *et al.*, “GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio,” in *Proc. INTERSPEECH*, 2021, pp. 3670–3674.
- [20] C. Wang *et al.*, “VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation,” in *Proc. Assoc. Comput. Linguist.*, 2021, pp. 993–1003.
- [21] S. Yang *et al.*, “SUPERB: Speech Processing Universal Performance Benchmark,” in *Proc. INTERSPEECH*, 2021, pp. 1194–1198.
- [22] J. Thienpondt, B. Desplanques, and K. Demuynck, “The Idlab Voxsrc-20 Submission: Large Margin Fine-Tuning and Quality-Aware Score Calibration in DNN Based Speaker Verification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5814–5818.
- [23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 5206–5210.
- [24] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690–4699.
- [25] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proc. Int. Conf. Learn. Represent.*, 2015.
- [26] M. Ott *et al.*, “Fairseq: A Fast, Extensible Toolkit for Sequence Modeling,” in *Proc. North Amer. chapter Assoc. Comput. Linguistics*, 2019, pp. 48–53.
- [27] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, “LibriMix: An Open-Source Dataset for Generalizable Speech Separation,” *arXiv:2005.11262*, 2020.
- [28] H. Ma, Z. Peng, M. Shao, J. Li, and J. Liu, “Extending Whisper with Prompt Tuning to Target-Speaker ASR,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 12 516–12 520.