Attention-based multi-channel speaker verification with ad-hoc microphone arrays

Chengdong Liang, Junqi Chen, Shanzheng Guan and Xiao-Lei Zhang
CIAIC, School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
E-mail: {liangchengdong, jqchen, gshanzheng}@mail.nwpu.edu.cn, xiaolei.zhang@nwpu.edu.cn

***Abstract*—Recently, ad-hoc microphone array has been widely studied. Unlike traditional microphone array settings, the spatial arrangement and number of microphones of ad-hoc microphone arrays are not known in advance, which hinders the adaptation of traditional speaker verification technologies to ad-hoc microphone arrays. To overcome this weakness, in this paper, we propose attention-based multi-channel speaker verification with ad-hoc microphone arrays. Specifically, we add an inter-channel processing layer and a global fusion layer after the pooling layer of a single-channel speaker verification system. The inter-channel processing layer applies a so-called residual self-attention along the channel dimension for allocating weights to different microphones. The global fusion layer integrates all channels in a way that is independent to the number of the input channels. We further replace the softmax operator in the residual self-attention with sparsemax, which forces the channel weights of very noisy channels to zero. Experimental results with ad-hoc microphone arrays of over 30 channels demonstrate the effectiveness of the proposed methods. For example, the multi-channel speaker verification with sparsemax achieves an equal error rate (EER) of over 20% lower than *oracle one-best* system on semi-real data sets, and over 30% lower on simulation data sets, in test scenarios with both matched and mismatched channel numbers.**

I. Introduction

In the past decades, the performance of automatic speaker verification (ASV) has been improved significantly, such as the i-vector based methods [1] and deep neural network (DNN) based methods [2]. However, such advances are mainly achieved on closed-talking scenarios with less interference. With the fast development of smart devices, such as smart speakers and various voice-enabled IoT gadgets, the need for far-field speech interaction will continue to grow. Recognizing who is speaking is essential to such smart devices for providing customized services. Far-field speech processing tasks including ASV remain challenging yet due to attenuated speech signals, noise interference, as well as room reverberations.

In order to solve the above challenging problem, various methods have been proposed at different stages of ASV systems. DNN based denoising methods [3] for single-channel speech enhancement [4], [5] and multi-channel speech enhancement [6]–[8] were explored for ASV systems under complex environments [9]. Linear prediction inverse modulation transfer function [10] and weighted prediction error [11] methods were used for dereverberation. Warped minimum variance distortionless response (MVDR) cepstral coefficients [12], power-normalized cepstral coefficients (PNCC) [13] and DNN bottleneck features [14] have been applied to ASV system for suppressing the adverse impacts of reverberation and noise.

The above research mainly focuses on single channel front-ends or multi-channel front-ends on single device. A microphone array with a known geometry is an important way to improve the performance of ASV. However, because speech quality degrades significantly when the distance between the speaker and microphone array enlarges, the performance of ASV is upper-bounded physically no matter how many microphones are added to the array [15]. Compared with the fixed microphone array, an ad-hoc microphone array consists of a set of microphone nodes randomly placed in an acoustic environment [16]. It provides more flexibility, and allows users to use their own mobile devices to virtually form a microphone array system. Recently, ad-hoc microphone arrays have been widely studied. [17] proposed deep ad-hoc beamforming based on speaker extraction for speech separation. In [18], a neural network architecture was proposed to capture both the inter-channel and temporal correlations from the multi-channel input of ad-hoc microphone arrays for speech separation. In [19], the authors provided a set of baseline systems that are trained with the far-field speaker verification data in the transfer learning manner, and then performed the system fusion at the score level.

In this paper, we propose an attention-based multi-channel ASV with ad-hoc microphone arrays. Specifically, we first propose an inter-channel processing layer based on residual self-attention and a global fusion layer. Compared with single-channel ASV models, we add the inter-channel processing layer and the global fusion layer after the pooling layer. The softmax function in the residual self-attention of the inter-channel processing layer transforms the similarity matrix to attention weights. Then, we further replace the softmax function with a so-called sparsemax function, which can force the channel weights of the noisy channels that do not contribute to the performance improvement to zero. We first train the single-channel ASV with clean speech data, then train the multi-channel ASV with ad-hoc data. This training strategy is motivated by the following two points: i) when we take some very noisy channels into training, the ASV system may not to be trained successfully; ii) the data of all channels is very large.

We conducted an extensive experiment on Librispeech simulated with ad-hoc microphone arrays and semi-real Libri-adhoc40 [20] corpora. Experimental result on the simulated

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:45:48 UTC from IEEE Xplore. Restrictions apply.

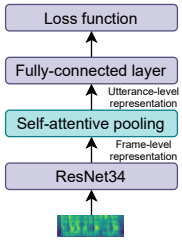

Fig. 1. Singel-channel speaker verification.

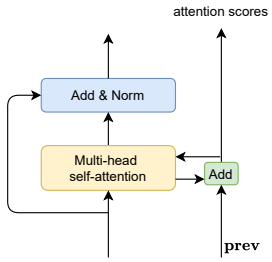

Fig. 2. The residual self-attention layer.

dataset shows that the proposed multi-channel ASV with sparsemax achieves a relative EER reduction of 36.2% over the *oracle one-best* system, and 6.2% over the multi-channel ASV with softmax on the matched 20-channel test scenario. Experimental result on the semi-real dataset shows that the proposed methods perform well in all test scenarios, and the model with sparsemax is slight better than the model with softmax.

II. Proposed System

In this section, we first introduce the single-channle ASV model. Then, we introduce the proposed attention-based multi-channle ASV. Finally, we introduce cross-channel multi-head residual self-attention with sparsemax.

A. Single-channel speaker verification system

As shown in Figure 1, the network structure of the single-channel ASV is the same as in [21]. The network architecture contains three main components: a front-end residual convolution neural network (ResNet) [22], a self-attentive pooling (SAP) [23] layer and a fully-connected layer. The front-end ResNet transforms the raw feature into a high-level abstract representation. The subsequent SAP layer outputs a single utterance-level representation. A fully-connected layer then further processes the utterance-level representation to be a more abstract utterance-level speaker embedding. All the components are jointly optimized in an end-to-end manner with a unified loss function. We adopt the angular prototypical loss [21]. It constructs training batches in the same way as that in the original prototypical loss [24].

B. Multi-channel speaker verification system

As shown in Figure 3, compared with the aforementioned single-channel system, the proposed multi-channel ASV system adds a novel inter-channel processing layer based residual self-attention and a global fusion layer after the SAP layer.

1) Inter-channel processing layer: The inter-channel processing layer collects the utterance-level representations from all channels, and outputs channel-reweighted utterance-level representations. As shown in Figure 4(a), the weights are calculated by a multi-head residual self-attention mechanism along the channel dimension, which is able to extract cross-channel information for the weight calculation. As shown in Figure 2, a residual self-attention layer takes the raw attention scores from previous layer as additive residual scores of the current attention function [25]. The detailed calculation process is as follows:

We stacked multiple inter-channel processing layers. For each layer, let $\mathbf{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_C]$ denote the input, where $\boldsymbol{x}_c \in \mathbb{R}^d$ is an utterance-level feature of the c-th channel. while C represents the number of channels and d denotes the dimension of the feature. We assume that the number of the attention heads of the self-attention is h. For each attention head, the input features $\mathbf{X}$ are transformed into query, key, and value embedding subspaces of dimension E respectively as follows:

$$\mathbf{Q}^i = \mathbf{X}\mathbf{W}_Q^i, \mathbf{K}^i = \mathbf{X}\mathbf{W}_K^i, \mathbf{V}^i = \mathbf{X}\mathbf{W}_V^i \tag{1}$$

where $d_k = E/h$; the matrices $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denote the query, key, and value embeddings respectively, all of which are in $\mathbb{R}^{C\times d_k}$; $\mathbf{W}_k \in \mathbb{R}^{d\times d_k}$ ($k \in \{K, Q, V\}$) are model parameters; and the superscript i denotes the i-th attention head. Within each head, the cross-channel similarity matrix is obtained from the multiplication of the query and key matrices. A softmax function is applied to each column of the cross-channel similarity matrix to obtain an attention matrix $\mathbf{A}^i \in \mathbb{R}^{C\times C}$:

$$\mathbf{A}^i = \text{softmax}\left(\frac{\mathbf{Q}^i \cdot \left(\mathbf{K}^i\right)^\top}{\sqrt{d_k}} + \mathbf{prev}\right), \tag{2}$$

where $\mathbf{prev}$ is the attention scores from the previous inter-channel processing layer. Finally, the new attention scores $\frac{\mathbf{Q}^i\cdot\left(\mathbf{K}^i\right)^\top}{\sqrt{d_k}} + \mathbf{prev}$ are sent to the upper layer. The value matrix $\mathbf{V}^i$ is multiplied by the attention matrix as

$$\mathbf{H}^i = \mathbf{A}^i \cdot \mathbf{V}^i, \tag{3}$$

where $\mathbf{H}^i \in \mathbb{R}^{C\times d_k}$ is the output of the i-th attention head, which is concatenated across the subspaces as:

$$\mathbf{Z} = \text{Concat}\left[\mathbf{H}^1, \mathbf{H}^2, \ldots, \mathbf{H}^h\right]\mathbf{W}^O \tag{4}$$

where $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ is the weight matrix of the linear projection layer.

The position-wise feed-forward network (FFN) with ReLU activation is applied to $\mathbf{Z}$ to generate the output of the current inter-channel processing layer. A residual connection is applied between the input and output of the cross-channel

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:45:48 UTC from IEEE Xplore. Restrictions apply.

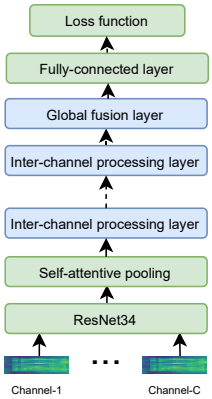

Fig. 3. Multi-channel speaker verfication. The green block is first trained by a single-channel ASV on clean speech, and then fixed. The blue block is trained with the multichannel data collected from ad-hoc microphone arrays in noisy environments.

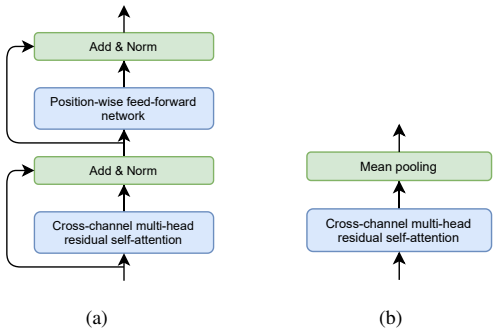

Fig. 4. (a) Inter-channel processing layer. (b) Global fusion layer.

multi-head self-attention layer as well as the FFN layer to mitigate the gradient vanishing problem.

2) Global fusion layer: After multiple layers of inter-channel processing, the global fusion layer is added above the top inter-channel processing layer. As shown in Figure 4(b), it is implemented by a cross-channel multi-head residual self-attention module, followed by a mean pooling operator. It fuses the information of all channels in a way that is independent to the number of the input channels.

C. Cross-channel multi-head self-attention with sparsemax

The common softmax has a limitation for ad-hoc microphone arrays that the elements of its output can never be zero, which cannot be used for channel selection. As we know, if some channels are very noisy, they may hurt the system when their weights are nonzero. To address this problem, we replace softmax with sparsemax [26] in (2), where sparsemax is defined as:

$$\mathrm{Sparsemax}(\boldsymbol{z}) = \underset{\boldsymbol{p}\in\Delta^{K-1}}{\arg\min}\|\boldsymbol{p}-\boldsymbol{z}\|^2 \tag{5}$$

where $\Delta^{K-1} = \left\{\boldsymbol{p}\in\mathbb{R}^K \mid \sum_{i=1}^K p_i = 1, p_i \geq 0\right\}$ represents a $(K-1)$-dimensional simplex. Sparsemax will return the Euclidean projection of the input vector $\boldsymbol{z}$ onto the simplex, which is a sparse vector. Its solution has the following closed-form:

$$\mathrm{Sparsemax}_i(\boldsymbol{z}) = \max\left(z_i - \tau(\boldsymbol{z}), 0\right) \tag{6}$$

where $\tau : \mathbb{R}^K \to \mathbb{R}$ is a function to find a soft threshold. Let $z_{(1)} \geq z_{(2)} \geq \ldots \geq z_{(K)}$ be the sorted coordinates of $\boldsymbol{z}$, and define $k(\boldsymbol{z}) := \max\left\{k\in[K] \mid 1 + kz_{(k)} > \sum_{j\leq k} z_{(j)}\right\}$. Then,

$$\tau(\boldsymbol{z}) = \frac{\left(\sum_{j\leq k(\boldsymbol{z})} z_{(j)}\right) - 1}{k(\boldsymbol{z})}. \tag{7}$$

III. Experiments

A. Dataset

Our experiments use three data sets, which are the Librispeech corpus [27], Librispeech simulated with ad-hoc microphone arrays (Libri-adhoc-simu), and Librispeech played back in real-world scenarios with 40 distributed microphone receivers (Libri-adhoc40) [20]. Librispeech contains more than 1000 hours of read English speech. In our experiments, we selected 960 hours of data including 2338 speakers to train single-channel ASV system, and selected 10 hours of data including 40 speakers for development.

Libri-adhoc-simu uses 'train-clean-100' subset of the Librispeech data as the training data, which contain 251 speakers. It uses 'dev-clean' subset as development data, and takes 'test-clean' subset as test data, which contain different 40 speakers respectively. For each utterance, we simulate a room. The length and width of the room are selected randomly from a range of $[5, 25]$ meters. The height is selected randomly from $[2.7, 4]$ meters. Multiple microphones and one speaker source are placed randomly in the room. We constrain the distance between the source and the walls to be greater than 0.2 meters, and the distance between the source and the microphones to be at least 0.3 meters [15]. We use an image-source model[1] to simulate a reverberant environment and selected T_{60} from a range of $[0.2, 0.4]$ second. A diffuse noise generator[2] is used to simulated uncorrelated diffuse noise. The noise source for training and development is a large-scale noise library containing over 20000 noise segments [28], and the noise source for test is the noise segments from CHiME-3 dataset [29] and NOISEX-92 corpus [30]. We randomly generate 20 channels for training and development, and 20, 30 and 40 channels respectively for test.

Libri-adhoc40 is collected by playing back the 'train-clean-100', 'dev-clean', and 'test-clean' subset of Librispeech in a large room [20]. The recording environment is a real office room with one loudspeaker and 40 microphones. It has strong reverberation with little additive noise. The distances between the loudspeaker and microphones are in a range of $[0.8, 7.4]$ meters. The positions of the loudspeaker and microphones are different in the training and test set. We randomly select 20

[1] https://github.com/ehabets/RIR-Generator
[2] https://github.com/ehabets/ANF-Generator

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:45:48 UTC from IEEE Xplore. Restrictions apply.

channels for each training and development utterances, and 20, 30, 40 channels for each test utterance which corresponds to three test scenarios.

B. Model structure

Figure 1 shows the architecture of the proposed single-channel model. We set the widths of the residual blocks to $\{16, 32, 64, 128\}$. The embedding size in the fully-connected layer is 512. As shown in Figure 3, multi-channel speaker verification system adds the Inter-channel processing layer and the Global fusion layer after the SAP layer. Four inter-channel processing layers are stacked. The output dimensions of each self-attention layer and FFT layer are both 256. The number of the attention heads in each attention layer is 4. We use voxceleb_trainer[3] to build our models.

C. Model training

During training, we use a fixed length 2 second temporal segment, extracted randomly from each utterance. A Hamming window with a width of 25ms and a step length of 10ms is used to extract the spectrum. The 40-dimensional Mel filterbanks are used as the input. Mean and variance normalisation (MVN) is performed by applying instancde normalisation [31] to the network input.

For each epoch, we randomly sample a maximum of 100 utterances from each speaker to reduce class imbalance. No data augmentation is performed during training, apart from the random sampling. We use the Adam optimizer with an initial learning rate of 0.001 decreasing by 5% every 10 epochs. First, the single-channel ASV system is trained for 200 epochs on the Librispeech corpus. Then, the parameters of the single-channel ASV are fixed and sent to the multi-channel ASV. Finally, we trained the multi-channel ASV system with Libri-adhoc-simu data and Libri-adhoc40 data respectively.

After training, we sample five 4-second temporal crops at regular intervals from each test segment, and compute the similarities between all possible combinations ($5 \times 5 = 25$) from each pair of segments. The mean of the 25 similarities is used as the score. We arrange all utterances in test set to get 6861780 trails for testing, including 183922 positive trails and 6677858 negative trails.

D. Results

We compare the proposed multi-channel ASV with softmax and sparsemax. Moreover, we construct an *oracle one-best* baseline, which picks the channel that is physically closest to the sound source as the input of the single-channel ASV model. Note that, for the *oracle one-best* baseline, the distances between the speaker and microphones are known beforehand.

Table I lists the preformance of the comparison methods on Libri-adhoc-simu. From the table, we see that all of the proposed methods perform well in both test scenarios. Particularly, the multi-channel ASV with softmax achieves an EER of 31.9% lower than *oracle one-best* baseline on the 20-channel test scenario, and 27.7% on the mismatched 30-channel test scenario. The generalization performance in the mismatched 30-channel and 40-channel test environment is even better than the performance in the matched 20-channel environment, which demonstrates the advantage of adding channels to ad-hoc microphone arrays. Sparsemax achieves significant performance improvement over softmax. For example, sparsemax achieves a relative EER reduction of 6.2% over softmax on the matched 20-channel.

[3]https://github.com/clovaai/voxceleb_trainer

TABLE I
EER(%) COMPARISON ON **LIBRI-ADHOC-SIMU**. THE TERM "CH" IS SHORT FOR CHANNELS IN TEST. MC-ASV IS SHORT FOR MULTI-CHANNEL ASV.

Method	20-ch	30-ch	40-ch
Oracle one-best	11.90	10.99	10.77
MC-ASV with softmax (proposed)	8.10	7.94	7.88
MC-ASV with sparsemax (proposed)	7.59	7.53	7.47

TABLE II
EER(%) COMPARISON ON **LIBRI-ADHOC40** SEMI-REAL DATA.

Method	20-ch	30-ch	40-ch
Oracle one-best	17.29	14.86	13.56
MC-ASV with softmax (proposed)	11.18	10.95	10.84
MC-ASV with sparsemax (proposed)	10.89	10.62	10.53

Table II lists the results on the Libri-adhoc40 semi-real data. From the table, we see that the proposed model performs well. The model with softmax achieves a relative EER reduction of 35.3% over the *oracle one-best* baseline on the 20-channel test scenario, and 26.3% on the mismatched 30-channel test scenario. Sparsemax is slightly better than softmax, with a relative EER reduction of 3% in all test scenarios. The above results show the proposed models is effective for ad-hoc microphone arrays.

IV. CONCLUSIONS

In this paper, we first present the multi-channel ASV model with ad-hoc microphone arrays using the inter-channel processing layer based residual self-attention and global fusion layer, which is proposed to take advantage of multi-channel information in a way that is independent of the number and permutation of the microphones. The inter-channel processing layer, which aims to learn the channel weights, consists of a residual self-attention and FFN. Multiple inter-channel processing layers are stacked, followed by a mean pooling layer for global information fusion. Then, we further replace the softmax opterator in the residual self-attention with sparsemax, which forces the channel weights of the noisy channels to zero. We evaluate our model on Libri-adhoc-simu with background noise and Libri-adhoc40 with high reverberation. Experimental results show that the proposed multi-channel ASV with sparsemax outperforms that with softmax and the oracle baseline in both simulated data and semi-real corpus. The results also demonstrate the importance of channel selection to ASV with large-scale ad-hoc microphone arrays.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:45:48 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010. I

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333. I

[3] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018. I

[4] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013. I

[5] A. Pandey and D. Wang, "A new framework for cnn-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 7, pp. 1179–1188, 2019. I

[6] Y. Jiang, D. Wang, R. Liu, and Z. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2112–2121, 2014. I

[7] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 196–200. I

[8] Z.-Q. Wang and D. Wang, "All-neural multi-channel speech enhancement." in *Interspeech*, 2018, pp. 3234–3238. I

[9] H. Taherian, Z.-Q. Wang, J. Chang, and D. Wang, "Robust speaker recognition based on single-channel and multi-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1293–1302, 2020. I

[10] B. J. Borgström and A. McCree, "The linear prediction inverse modulation transfer function (lp-imtf) filter for spectral enhancement, with applications to speaker recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4065–4068. I

[11] L. Mošner, P. Matějka, O. Novotnỳ, and J. H. Černockỳ, "Dereverberation and beamforming in far-field speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5254–5258. I

[12] Q. Jin, R. Li, Q. Yang, K. Laskowski, and T. Schultz, "Speaker identification with distant microphone speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 4518–4521. I

[13] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (pncc) for robust speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 24, no. 7, pp. 1315–1329, 2016. I

[14] T. Yamada, L. Wang, and A. Kai, "Improvement of distant-talking speaker identification using bottleneck features of dnn." in *Interspeech*, 2013, pp. 3661–3664. I

[15] J. Chen and X.-L. Zhang, "Scaling sparsemax based channel selection for speech recognition with ad-hoc microphone arrays," *arXiv preprint arXiv:2103.15305*, 2021. I, III-A

[16] X.-L. Zhang, "Deep ad-hoc beamforming," *Computer Speech & Language*, vol. 68, p. 101201, 2021. I

[17] Z. Yang, S. Guan, and X.-L. Zhang, "Deep ad-hoc beamforming based on speaker extraction for target-dependent speech separation," *arXiv preprint arXiv:2012.00403*, 2020. I

[18] D. Wang, T. Yoshioka, Z. Chen, X. Wang, T. Zhou, and Z. Meng, "Continuous speech separation with ad hoc microphone arrays," *arXiv preprint arXiv:2103.02378*, 2021. I

[19] X. Qin, H. Bu, and M. Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7609–7613. I

[20] S. Guan, S. Liu, J. Chen, W. Zhu, S. Li, X. Tan, Z. Yang, M. Xu, Y. Chen, J. Wang *et al.*, "Libri-adhoc40: A dataset collected from synchronized ad-hoc microphone arrays," *arXiv preprint arXiv:2103.15118*, 2021. I, III-A

[21] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020. II-A

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. II-A

[23] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification." in *Interspeech*, vol. 2018, 2018, pp. 3573–3577. II-A

[24] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *arXiv preprint arXiv:1703.05175*, 2017. II-A

[25] R. He, A. Ravula, B. Kanagal, and J. Ainslie, "Realformer: Transformer likes residual attention," *arXiv e-prints*, pp. arXiv–2012, 2020. II-B1

[26] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1614–1623. II-C

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210. III-A

[28] X. Tan and X.-L. Zhang, "Speech enhancement aided end-to-end multi-task learning for voice activity detection," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6823–6827. III-A

[29] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chime-speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 504–511. III-A

[30] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993. III-A

[31] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016. III-C

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:45:48 UTC from IEEE Xplore. Restrictions apply.