EMBEDDING AGGREGATION FOR FAR-FIELD SPEAKER VERIFICATION WITH DISTRIBUTED MICROPHONE ARRAYS

Danwei Cai[1], *Ming Li*[1,2]

[1]Department of Electrical and Computer Engineering, Duke University, Durham, USA
[2]Data Science Research Center, Duke Kunshan University, Kunshan, China
ming.li369@duke.edu

ABSTRACT

With the successful application of deep speaker embedding networks, the performance of speaker verification systems has significantly improved under clean and close-talking settings; however, unsatisfactory performance persists under noisy and far-field environments. This study aims at improving the performance of far-field speaker verification systems with distributed microphone arrays in the smart home scenario. The proposed learning framework consists of two modules: a deep speaker embedding module and an aggregation module. The former extracts a speaker embedding for each recording. The latter, based on either averaged pooling or attentive pooling, aggregates speaker embeddings and learns a unified representation for all recordings captured by distributed microphone arrays. The two modules are trained in an end-to-end manner. To evaluate this framework, we conduct experiments on the real text-dependent far-field datasets Hi Mia. Results show that our framework outperforms the naive averaged aggregation methods by 20% in terms of equal error rate (EER) with six distributed microphone arrays. Also, we find that the attention-based aggregation advocates high-quality recordings and repels low-quality ones.

Index Terms— speaker verification, deep speaker embedding, far-field, distributed microphone arrays

1. INTRODUCTION

As a key technology in biometric authentication, automatic speaker recognition analyzes a given speech and recognizes the speaker identity using signal processing and pattern recognition algorithms. Typically, speaker recognition can be divided into two tasks, i.e., speaker identification and speaker verification. The former matches a voice with a specific speaker, while the latter determines whether a pair of speeches belong to the same speaker. It is commonly applied to e-commerce systems, call centers, smartphones, smart speakers, automobiles, etc. In the past few years, deep speaker embedding based methods have significantly improved the performance of speaker recognition systems under clean and close-talking settings [1, 2]. However, unsatisfactory performance persists under far-field and complex environmental settings due to the long-range fading, room reverberation, and complex environmental noises. These adverse effects result in the loss of speech intelligibility and quality, imposing challenges on speaker recognition.

Various approaches have been proposed to address this issue. At the signal level, dereverberation methods [3, 4], deep neural network (DNN)-based denoising methods [5, 6, 7, 8] and multichannel processing with beamforming [4, 9, 10] are employed for speaker recognition under complex environments. At the modeling level, data augmentation [11, 12] and transfer learning [13] prove to be useful for robust speaker embedding learning with limited target domain data. To learn a noise-invariant speaker representation, adversarial training [14, 15, 16] and variability-invariant loss [17] have been investigated within the deep speaker framework. To minimize the discrepancy between speech enhancement and speaker recognition, the speech enhancement network is trained with the guidance from a speaker network [18, 19]. The joint training of these two networks can also improve the robustness of the speaker embedding [20, 21]. With the microphone array, a deep speaker framework with multi-channel inputs is used for speaker embedding extraction [22]. Moreover, in the testing phase, enrollment data augmentation is applied to reduce the channel-mismatch between the enrollment and testing utterances [13].

With all these methods, far-field speaker recognition is still challenging and attracts increasing attention from the research community. The Voices Obscured in Complex Environmental Settings (VOiCES) Challenge launched in 2019 aims to benchmark state-of-the-art speech processing methods in far-field and noisy conditions [23]. A speaker recognition benchmark derived from the publicly-available CHiME-5 corpus, which is initially designed for far-field automatic speech recognition, is described in [24]. In addition, the wake-up word dataset *Hi Mia* has been released to facilitate researches in far-field speaker recognition [25]. Also, Far-Field Speaker Verification Challenge (FFSVC 2020) was launched to boost the speaker verification research with spe-

cial focus on far-field distributed microphone arrays under noisy conditions in real scenes [26].

This study focuses on far-field speaker verification with distributed microphone arrays in the smart home scenario, where recorders at different locations make multiple recordings of the same utterance. Several studies has investigated automatic speech recognition (ASR) with far-field distributed microphone arrays [27, 28, 29], yet this special topic in speaker recognition still remains to be explored.

In this study, we propose a learning framework to learn a unified speaker embedding from multiple recordings of a single recognition utterance. It consists of a speaker embedding network and an aggregation network. The former maps each recording to a speaker embedding, and the latter aggregates embeddings of multiple recordings to form a unified representation for recognition. Also, considering that the recordings vary in quality due to variations in room acoustics at various locations, an attention-based aggregation approach is put forward to fuse the utterance-level speaker embeddings from distributed microphone arrays adaptively. We observe that the attention-based aggregation automatically learns to advocate high-quality recordings and repel those of lower quality.

2. METHODS

Figure 1 shows the proposed embedding aggregation network architecture for far-field speaker recognition with distributed microphone arrays. It takes a set of recordings as input and produces a unified speaker representation for recognition. The proposed network consists of a speaker embedding network to learn speaker representation for each recording and an attention-based aggregation network to fuse these representations into a speaker embedding for a recognition utterance.

2.1. Deep Speaker Embedding

The deep speaker embedding network learns a speaker embedding from a single speech. It consists of a frame-level local pattern extractor, an utterance-level encoding layer, and a speaker classification layer. A local pattern extractor is typically a time-delayed neural network (TDNN) or convolutional neural network (CNN). It learns speaker representation from the spectral feature sequence of varying length. This representation, still a temporally-ordered frame-level feature sequence, is then fed into an encoding layer to get an utterance level representation known as speaker embedding. The most common encoding method is the average pooling layer, which aggregates the mean or (and) standard deviation statistics from the frame-level representation [1, 2]. Other encoding layers include self-attentive pooling layer [30, 31], learnable dictionary encoding layer [32], and dictionary-based NetVLAD layer [33, 34]. Once the utterance-level representation is extracted, fully connected layer(s) are employed to further abstract the speaker embedding and classifies the speakers in the training set.

In this work, we train the speaker network with the residual convolutional neural network (ResNet) [35] as the local pattern extractor and the global statistics pooling (GSP) layer as the encoding layer. Specifically, from the ResNet's output feature map $\mathbf{F} \in \mathbb{R}^{C\times F\times T}$, GSP layer calculates the mean μ_c and standard deviation σ_c of the c^{th} feature map to get a utterance-level representation $\mathbf{V} = [\mu_1, \mu_2, \cdots, \mu_C, \sigma_1, \sigma_2, \cdots, \sigma_C]$:

$$\begin{aligned} \mu_c &= \frac{1}{F\times T}\sum_{f=1}^{F}\sum_{t=1}^{T}\mathbf{F}_{c,f,t} \\ \sigma_c &= \sqrt{\frac{1}{F\times T}\sum_{f=1}^{F}\sum_{t=1}^{T}(\mathbf{F}_{c,f,t}-\mu_c)^2} \end{aligned} \tag{1}$$

and C, F, T denote the dimension of channels, frequency and time axis of the ResNet's output feature map respectively. Then, a fully connected layer is used to map this utterance-level representation $\mathbf{V} \in \mathbb{R}^{2C}$ to a low-dimensional (i.e, 128) speaker embedding space.

2.2. Embedding Aggregation

With the setting of distributed microphone arrays, multiple recordings are captured simultaneously for a single recognition utterance. The speaker network extracts a speaker embedding for each of these recordings. The aggregation network, which is either an averaged pooling or an attentive pooling, gathers these embeddings and learns a speaker representation for recognition.

Specifically, for a single recognition utterance, the unified speaker representation $\mathbf{r}$ is a linear combination of the all speaker embeddings $\{\mathbf{f}_k\}_{k=1}^{K}$ extracted from K recordings of the distributed microphone arrays:

$$\mathbf{r} = \sum_{k=1}^{K} w_k \mathbf{f}_k \tag{2}$$

where w is the weight and $\sum_{k=1}^{K} w_k = 1$.

2.2.1. Averaged Embedding Aggregation

Averaged pooling equally weights all of the speaker embeddings from different recording channels. Therefore, weights w_k in equation (2) are

$$w_k = \frac{1}{K}, \forall k \tag{3}$$

2.2.2. Attentive Embedding Aggregation

For all the recordings captured by distributed recording devices, speech qualities may vary due to variations in the

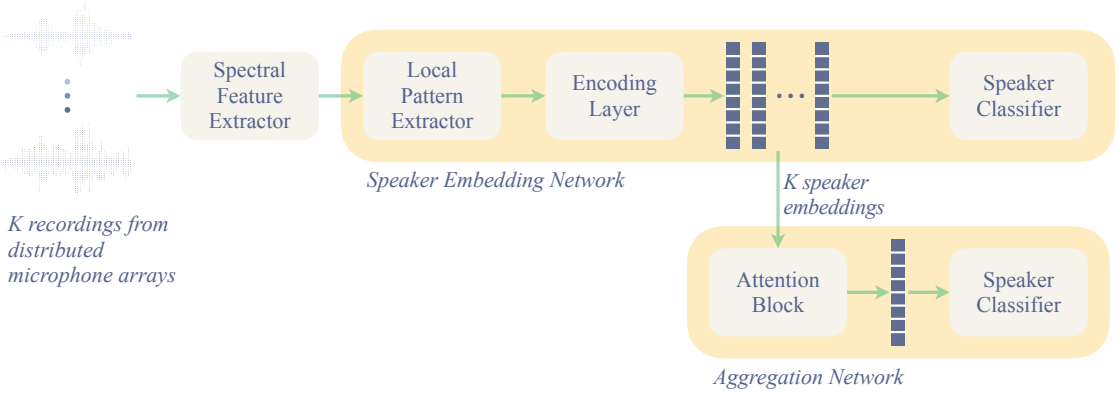

Fig. 1. The proposed embedding aggregation framework for far-field speaker recognition with distributed microphone array.

acoustic environment at different locations. Considering that these recordings of varying qualities may contribute differently to the final speaker representation, we introduce a self-attentive pooling layer to learn weights w_k in equation (2) and adaptively aggregate speaker embeddings from various devices.

Specifically, speaker embeddings $\{\mathbf{f}_k\}_{k=1}^K$ of various recordings from various channels are firstly fed into a one-layer perceptron to get hidden representations $\{\mathbf{h}_k\}_{k=1}^K$:

$$\mathbf{h}_k = \tanh(\mathbf{W}\mathbf{f}_k + \mathbf{b}) \tag{4}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight matrix and bias vector of the perceptron respectively, and $\tanh(\cdot)$ imposes the hyperbolic tangent nonlinearity. Then, dot products of hidden representations $\mathbf{h}_k$ and the learnable parameter $\mathbf{q}$ are calculated, yielding a set of corresponding significances, from which a $\mathrm{softmax}$ operator is employed to generate positive attentive weights $\{w_k\}_{k=1}^K$:

$$w_k = \frac{\exp(\mathbf{q}^T\mathbf{h}_k)}{\sum_{j=1}^K \exp(\mathbf{q}^T\mathbf{h}_j)} \tag{5}$$

With the aggregated speaker embedding, fully connected layers are employed to classify the speakers in the training set as in the speaker embedding network.

2.3. Network Training

Typically, the scale of the real world far-field dataset is small comparing to the general speaker recognition dataset. To learn a robust feature descriptor for speaker recognition, the speaker embedding network is firstly trained on a large-scale general speaker recognition dataset with single-channel utterances. With the converged speaker embedding network, the aggregation network is trained on the real far-field dataset.

The aggregation network can be trained either simultaneously in an end-to-end manner or separately one by one. To enable end-to-end training, the aggregation layer is plugged between the encoding layer and the speaker classifier of the speaker embedding network. All parameters are updated simultaneously during training. Also, the aggregation network can be trained separately. Specifically, we train the aggregation network on top of the speaker representations extracted by the encoding layer. The speaker classification layers in the aggregation network are initialized from the speaker embedding network.

Both the speaker embedding network and the aggregation network are trained with cross-entropy loss.

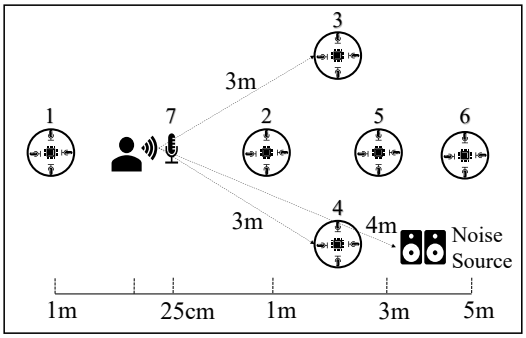

Fig. 2. Recording environment of *Hi Mia* dataset [25].

3. EXPERIMENTS

3.1. Dataset

3.1.1. *Far-field text-dependent dataset*

We conduct the experiments on the far-field text-dependent dataset *Hi Mia*. The dataset is recorded in both English and Mandarin Chinese [25]. We only choose the Mandarin Chinese part. In each utterance, recordings are captured by one close-talking microphone as well as six 16-channel circular microphone arrays. The recording setting is shown in figure 2. The average duration of the recordings is around 1 second. In this study, the testing set is selected as described in

[25], which contains 3,520 utterances from 44 speakers. The remaining 23,680 utterances from 296 speakers are saved for training.

To simulate the real-life scenario, we use the close-talking clean data for enrollment and employ the real far-field noisy utterances for testing, as described in [13]. During testing, enrolling utterances are from the close-talking channel, while testing utterances are from the far-field microphone array(s). In each microphone array, we select six recording channels (channel 1, 3, 6, 9, 11, and 14). Each close-talking enrolling utterance is scored with all far-field testing utterances, yielding 12,390,400 (3520×3520) testing trials. Among the 3,520 testing utterances, there are 2,640 and 880 utterances recorded in quiet and noisy environments. We further split the testing trials into clean testing part with 9,292,800 (3520×2640) trials and noisy testing part with 3,097,600 (3520×880) trials.

3.1.2. Close-talking text-independent dataset

As described in section 2.3, the speaker embedding network is trained with a general large-scale dataset. The *AISHELL-ASR0009*[1], which is a Mandarin speech recognition dataset, is used for this purpose. In this study, 1947 speakers with 959,902 utterances from the high-quality microphone channel of the dataset are selected for training. The average duration of the utterances is 3.54 seconds.

3.1.3. Data augmentation

We perform data augmentation with MUSAN dataset [36]. Either background additive noise or reverberation is added to close-talking utterances during training. For the additive noise, the signal-to-noise ratios (SNR) are set between 0 to 20 dB, and the noise type includes ambient noise, music, television, and babble noise. The television noise is generated with one music file and one speech file. The babble noise is constructed by mixing three to eight speech files into one. For the reverberation, the convolution operation is performed with the simulated room impulse responses (RIR) in MUSAN. We only use RIRs from small and medium rooms.

3.2. Experimental Setup

For the input features, speech signals are converted to 64-dimensional log Mel-filterbank energies with a frame-length of 25 ms. The features are mean normalized before fed into the network. We adopt an online data preparation and augmentation strategy for training [37]. A random length of duration between 2 to 4 seconds is generated for each data batch on-the-fly. Also, random background noise or reverberative noise is added for each training sample when generating the data batch. With this online data preparation strategy, the network never "sees" the same training sample, which helps to improve the generalization ability of the model.

Table 1. The network architecture, $\mathbf{C}$(kernal size, stride) denotes the convolutional layer, $[\cdot]$ denotes the residual block; K is the number of recordings channels and L relates to the duration of the speech.

Layer	Output Size	Structure
Conv1	$K \times 32 \times 64 \times L$	$\mathbf{C}(3 \times 3, 1)$
Residual Layer 1	$K \times 32 \times 64 \times L$	$\begin{bmatrix}\mathbf{C}(3 \times 3, 1)\\ \mathbf{C}(3 \times 3, 1)\end{bmatrix} \times 3$
Residual Layer 2	$K \times 64 \times 32 \times \frac{L}{2}$	$\begin{bmatrix}\mathbf{C}(3 \times 3, 2)\\ \mathbf{C}(3 \times 3, 1)\\ \mathbf{C}(3 \times 3, 1)\\ \mathbf{C}(3 \times 3, 1)\end{bmatrix} \times 3$
Residual Layer 3	$K \times 128 \times 16 \times \frac{L}{4}$	$\begin{bmatrix}\mathbf{C}(3 \times 3, 2)\\ \mathbf{C}(3 \times 3, 1)\\ \mathbf{C}(3 \times 3, 1)\\ \mathbf{C}(3 \times 3, 1)\end{bmatrix} \times 5$
Residual Layer 4	$K \times 256 \times 8 \times \frac{L}{8}$	$\begin{bmatrix}\mathbf{C}(3 \times 3, 2)\\ \mathbf{C}(3 \times 3, 1)\\ \mathbf{C}(3 \times 3, 1)\\ \mathbf{C}(3 \times 3, 1)\end{bmatrix} \times 2$
Utterance Encoding	$K \times 512$	Global Statistics Pooling Layer
Embedding Aggregation	512	Self-Attentive Pooling Layer
Speaker Classifier	128 # speakers	Fully Connected Layer Fully Connected Layer

The detailed network architecture is in table 1. The front-end local pattern extractor is based on the well known ResNet-34 architecture [35]. ReLU activation and batch normalization are applied to each convolutional layer. Dropout [38] is added before the speaker classification layer to prevent overfitting. During the testing phase, the 128-dimensional speaker embedding is extracted from the output of the penultimate fully-connected layer.

Since the final verification task is conducted on the text-dependent dataset, we train the text-dependent speaker network with model fine-tuning [13] to ensure a robust text-dependent speaker network for single-channel utterance. Specifically, we firstly train a text-independent speaker embedding model with 1947 speakers in *AISHELL-ASR0009*. Network parameters are updated using stochastic gradient descent (SGD) algorithm with a momentum of 0.95 and a weight decay of 1e-4. The learning rate is initially set to 0.1 and is divided by ten whenever the training loss reaches

[1]More details can be found at `https://aishell-asr009.oss-cn-beijing.aliyuncs.com/AISHELL-ASR0009.pdf`

Table 2. Verification performance (DCF and EER[%]) with the setting of distributed circular microphone arrays. Each array contains six recording channels. The boldface indicates the best results tested with the whole trials.

Aggregation	Training	Trials	1 Array		2 Arrays		3 Arrays		4 Arrays		5 Arrays		6 Arrays	
Averaged	One by One	All	0.666	4.20	0.628	3.60	0.616	3.36	0.608	3.26	0.598	3.23	0.596	3.18
		Clean	0.643	3.81	0.607	3.27	0.593	3.07	0.586	3.01	0.576	2.99	0.574	2.95
		Noisy	0.733	5.17	0.684	4.49	0.678	4.16	0.668	3.91	0.652	3.86	0.651	3.79
Attentive	One by One	All	0.666	4.20	0.624	3.44	0.606	3.27	0.596	3.19	0.584	3.12	0.580	3.06
		Clean	0.642	3.82	0.606	3.20	0.587	3.05	0.579	3.00	0.570	2.95	0.566	2.90
		Noisy	0.732	5.17	0.673	4.07	0.658	3.86	0.642	3.70	0.616	3.59	0.611	3.50
Averaged	End to End	All	0.622	4.03	0.563	3.22	0.553	3.04	0.542	2.92	0.534	2.89	0.532	2.84
		Clean	0.591	3.64	0.539	2.90	0.526	2.76	0.518	2.66	0.508	2.63	0.507	2.58
		Noisy	0.712	5.14	0.632	4.13	0.634	3.89	0.615	3.69	0.609	3.64	0.607	3.58
Attentive	End to End	All	**0.612**	**3.72**	**0.552**	**2.94**	**0.527**	**2.77**	**0.518**	**2.66**	**0.506**	**2.58**	**0.500**	**2.52**
		Clean	0.590	3.47	0.538	2.77	0.515	2.60	0.507	2.53	0.498	2.48	0.492	2.42
		Noisy	0.673	4.37	0.586	3.40	0.555	3.28	0.536	3.03	0.513	2.87	0.506	2.77

a plateau. Then, this model is employed to initialize the text-dependent model. Single-channel utterances from 296 speakers in *Hi Mia* dataset are used to fine-tune the text-dependent speaker embedding model. The learning rate is set to 1e-3 and is divided by ten whenever the training loss reaches a plateau.

To train the aggregation network, N microphone arrays are randomly selected for a training utterance. Then one recording channel for each of the N arrays is chosen for training, resulting in $K = N$ recording channels from different microphone arrays. For the close-talking utterances, N noisy copies are generated. N is set between 1 to 6 to match the number of microphone arrays in the dataset.

At the testing stage, cosine similarity is used for scoring. We use equal error rate (EER) and detection cost function (DCF) as the performance metrics. The reported DCF is the average of two minimum DCFs when P_{target} is 0.01 and 0.001.

3.3. Experimental Results

3.3.1. Single channel results

We first show the single-channel verification results of *HI MIA* dataset in table 3. For each microphone array, the verification performance of the best and worst channels, as well as the embedding level fusion result, are shown. We observe that the verification performance of microphone arrays is correlated with their spatial positions (as shown in figure 2). Both microphone arrays 1 and 2 are nearest to the speaker. However, the second one is most likely to record high-quality speech as it is in the direction of the speaker's talking, while the first one may capture less signal and be influenced by reflections. Therefore, Array 2 achieves the best performance among all the arrays. Although array 3, 4, and 5 are of the same distance from the speaker, array 4 is much closer to the noise sources. As a result, the recordings of array 4 are more likely to have lower SNRs and worse verification performance compared with array 3 and 5.

Table 3. Single channel results (DCF and EER[%]). Each microphone array (as indexed in figure 2) contains 6 recording channels. For each array, the verification performance of the best and worst channels, as well as the embedding level fusion result, are shown. The boldface indicates the microphone arrays with best and worst verification performance.

Microphone	Best		Worst		Averaged	
Closed-talking			0.519	2.59		
Array 1	0.729	4.53	0.764	4.81	0.710	4.13
Array 2	**0.628**	**3.46**	**0.647**	**3.50**	**0.618**	**3.23**
Array 3	0.692	4.66	0.727	4.95	0.655	4.27
Array 4	**0.744**	**5.49**	**0.774**	**5.81**	**0.720**	**5.22**
Array 5	0.695	4.45	0.736	4.74	0.684	4.13
Array 6	0.672	4.54	0.706	4.71	0.651	4.15

3.3.2. Distributed microphone arrays results

Table 2 shows the performances with distributed microphone arrays. We investigate six testing conditions in which different numbers of microphone arrays are used for a testing utterance. For the testing condition with N microphone arrays, N arrays are randomly sampled for each testing utterance, yielding $K = N \times 6$ recording channels in total. With only one microphone array and without end-to-end training, the performances of attentive pooling and average pooling

Table 4. Verification performance (DCF and EER[%]) with the setting of distributed microphones. Each channel of the microphone array is tested independently, resulting in 6 verification results for each experiment setting. The means of the DCF and EER of the six results are reported. The boldface indicates the best results.

Aggregation	Training	1 Mic		2 Mics		3 Mics		4 Mics		5 Mics		6 Mics	
Averaged	One-by-One	0.706	4.64	0.652	3.74	0.631	3.49	0.620	3.37	0.608	3.32	0.605	3.26
Attentive	One-by-One	0.706	4.64	0.652	3.71	0.628	3.45	0.616	3.33	0.599	3.24	0.593	3.16
Averaged	End-to-End	0.664	4.52	0.589	3.47	0.571	3.21	0.555	3.05	0.546	3.00	0.542	2.93
Attentive	End-to-End	**0.649**	**4.18**	**0.577**	**3.18**	**0.546**	**2.95**	**0.532**	**2.79**	**0.519**	**2.68**	**0.510**	**2.59**

are the same, since the acoustic environments of the recording channels within one microphone array have no significant differences. When more microphone arrays are used, attentive pooling outperforms over average pooling. With the same speaker embedding, the attentive pooling, which can be easily computed, outperforms the average pooling by 3.8% in terms of EER when the aggregation network is separately trained. Moreover, the aggregation model trained in end-to-end fashion further improves performance. We argue that this improvement is attributed to the enhanced modeling ability of the speaker embedding network when it is trained with the aggregation network. Training the attentive aggregation module with the speaker embedding network in an end-to-end fashion improves the simple embedding averaged aggregation by 20% in terms of EER with six distributed microphone arrays.

In figure 3, we visualize the means and standard deviations of the attentive weights learned from the end-to-end model. For each utterance in the *Hi Mia* test set, the attentive weights includes six microphone arrays, each with six selected recording channels. Attentive weights are correlative to the position of microphone arrays, as shown in figure 2. This observation is the same as what we found in the single-channel experiments.

3.3.3. Distributed microphones results

Table 4 shows the results with the setting of distributed microphones. The microphone is selected from one channel in the microphone array. The microphone devices selected to construct the testing trial are same as section 3.3.2. The recordings from the same channel index of the microphone arrays are selected for testing, yielding six testing sets for each testing condition. For each testing condition with K distributed microphones, the EERs and DCFs of 6 testing sets are averaged. We observe the same as what is found in the distributed microphone arrays experiments. With six distributed microphones, the proposed attentive aggregation method trained in an end-to-end manner outperforms the simple embedding averaged aggregation by 20.6% and 15.7% in terms EER and DCF respectively.

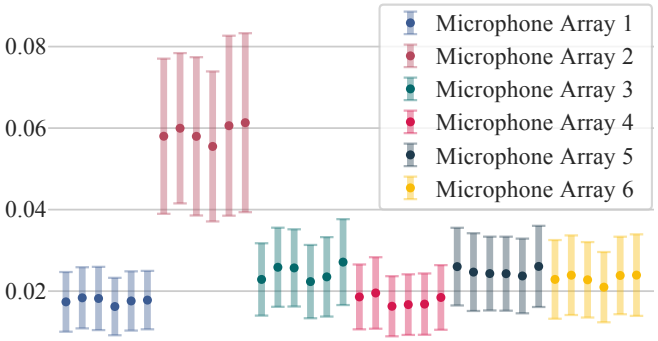

Fig. 3. Means and standard deviations of the attentive weights of 6 microphone arrays (each with 6 channels) on *Hi Mia* test set.

4. CONCLUSION

This study presents an embedding aggregation framework for far-field speaker recognition with distributed microphone arrays. We investigate the simple averaged aggregation as well as the attentive aggregation. The attention-based method produces a set of attentive weights to adaptively fuse speaker embeddings of all recording channels within a recognition utterance. The embedding aggregation module is simple and can be trained with any speaker embedding network in an end-to-end manner. Experiments conducted on the real world far-field text-dependent datasets show that the proposed method outperforms the simple aggregation methods.

5. ACKNOWLEDGEMENT

This research is funded by Duke Kunshan University.

6. REFERENCES

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "x-vectors: Robust DNN Embeddings for Speaker Recognition," in *ICASSP*, 2018, pp. 5329–5333.

[2] Weicheng Cai, Jinkun Chen, and Ming Li, "Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System," in *Odyssey*, 2018, pp. 74–81.

[3] Bengt J. Borgstrom and Alan McCree, "The Linear Prediction Inverse Modulation Transfer Function (lP-IMTF) Filter for Spectral Enhancement, with Applications to Speaker Recognition," in *ICASSP*, 2012, pp. 4065–4068.

[4] Ladislav Mosner, Pavel Matejka, Ondrej Novotny, and Jan Honza Cernocky, "Dereverberation and Beamforming in Far-Field Speaker Recognition," in *ICASSP*, 2018, pp. 5254–5258.

[5] Xiaojia Zhao, Yuxuan Wang, and DeLiang Wang, "Robust Speaker Identification in Noisy and Reverberant Conditions," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 836–845, 2014.

[6] Morten Kolboek, Zheng-Hua Tan, and Jesper Jensen, "Speech Enhancement Using Long Short-Term Memory based Recurrent Neural Networks for Noise Robust Speaker Verification," in *SLT*, 2016, pp. 305–311.

[7] Zeyan Oo, Yuta Kawakami, Longbiao Wang, Seiichi Nakagawa, Xiong Xiao, and Masahiro Iwahashi, "DNN-Based Amplitude and Phase Feature Enhancement for Noise Robust Speaker Identification," in *Interspeech*, 2016, pp. 2204–2208.

[8] Oldrich Plchot, Lukas Burget, Hagai Aronowitz, and Pavel Matejka, "Audio enhancing with DNN autoencoder for speaker recognition," in *ICASSP*, 2016, pp. 5090–5094.

[9] Hassan Taherian, Zhong-Qiu Wang, and DeLiang Wang, "Deep Learning Based Multi-Channel Speaker Recognition in Noisy and Reverberant Environments," in *Interspeech*, 2019, pp. 4070–4074.

[10] Ladislav Mošner, Oldřich Plchot, Johan Rohdin, and Jan Černocký, "Utilizing VOiCES Dataset for Multichannel Speaker Verification with Beamforming," in *Odyssey*, 2020, pp. 187–193.

[11] Danwei Cai, Xiaoyi Qin, Weicheng Cai, and Ming Li, "The DKU System for the Speaker Recognition Task of the 2019 VOiCES from a Distance Challenge," in *Interspeech*, 2019, pp. 2493–2497.

[12] Pavel Matejka, Oldrich Plchot, Hossein Zeinali, Ladislav Mosner, Anna Silnova, Lukas Burget, Ondrej Novotny, and Ondrej Glembek, "Analysis of BUT Submission in Far-Field Scenarios of VOiCES 2019 Challenge," in *Interspeech*, 2019, pp. 2448–2452.

[13] Xiaoyi Qin, Danwei Cai, and Ming Li, "Far-Field End-to-End Text-Dependent Speaker Verification Based on Mixed Training Data with Transfer Learning and Enrollment Data Augmentation," in *Interspeech*, 2019, pp. 4045–4049.

[14] Jianfeng Zhou, Tao Jiang, Lin Li, Qingyang Hong, Zhe Wang, and Bingyin Xia, "Training Multi-task Adversarial Network for Extracting Noise-robust Speaker Embedding," in *ICASSP*, 2019, pp. 6196–6200.

[15] Zhong Meng, Yong Zhao, Jinyu Li, and Yifan Gong, "Adversarial Speaker Verification," in *ICASSP*, 2019, pp. 6216–6220.

[16] Raghuveer Peri, Monisankha Pal, Arindam Jati, Krishna Somandepalli, and Shrikanth Narayanan, "Robust Speaker Recognition Using Unsupervised Adversarial Invariance," in *ICASSP*, 2020, pp. 6614–6618.

[17] Danwei Cai, Weicheng Cai, and Ming Li, "Within-Sample Variability-Invariant Loss for Robust Speaker Recognition Under Noisy Environments," in *ICASSP*, 2020, pp. 6469–6473.

[18] Suwon Shon, Hao Tang, and James Glass, "VoiceID Loss: Speech Enhancement for Speaker Verification," in *Interspeech*, 2019, pp. 2888–2892.

[19] Saurabh Kataria, Phani Sankar Nidadavolu, Jess Villalba, Nanxin Chen, Paola Garca-Perera, and Najim Dehak, "Feature Enhancement with Deep Feature Losses for Speaker Verification," in *ICASSP*, 2020, pp. 7584–7588.

[20] Fei Zhao, Hao Li, and Xueliang Zhang, "A Robust Text-independent Speaker Verification Method Based on Speech Separation and Deep Speaker," in *ICASSP*, 2019, pp. 6101–6105.

[21] Joon-Young Yang and Joon-Hyuk Chang, "Joint Optimization of Neural Acoustic Beamforming and Dereverberation with x-Vectors for Robust Speaker Verification," in *Interspeech*, 2019, pp. 4075–4079.

[22] Danwei Cai, Xiaoyi Qin, and Ming Li, "Multi-Channel Training for End-to-End Speaker Recognition Under Reverberant and Noisy Environment," in *Interspeech*, 2019, pp. 4365–4369.

[23] Mahesh Kumar Nandwana, Julien van Hout, Colleen Richey, Mitchell McLaren, Maria A. Barrios, and Aaron Lawson, "The VOiCES from a Distance Challenge 2019," in *Interspeech*, 2019, pp. 2438–2442.

[24] Daniel Garcia-Romero, David Snyder, Shinji Watanabe, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker Recognition Benchmark Using

the CHiME-5 Corpus," in *Interspeech*, 2019, pp. 1506–1510.

[25] Xiaoyi Qin, Hui Bu, and Ming Li, "HI-MIA: A Far-Field Text-Dependent Speaker Verification Database and the Baselines," in *ICASSP*, 2020, pp. 7609–7613.

[26] Xiaoyi Qin, Ming Li, Hui Bu, Wei Rao, Rohan Kumar Das, Shrikanth Narayanan, and Haizhou Li, "The INTERSPEECH 2020 Far-Field Speaker Verification Challenge," in *Interspeech*, 2020, pp. 3456–3460.

[27] Seyedmahdad Mirsamadi and John H.L. Hansen, "Multichannel feature enhancement in distributed microphone arrays for robust distant speech recognition in smart rooms," in *SLT*, 2014, pp. 507–512.

[28] Shoko Araki1, Nobutaka Ono, Keisuke Kinoshita, and Marc Delcroix, "Meeting Recognition with Asynchronous Distributed Microphone Array Using Block-Wise Refinement of Mask-Based MVDR Beamformer," in *ICASSP*, 2018, pp. 5694–5698.

[29] Feifei Xiong, Jisi Zhang, Bernd Meyer, Heidi Christensen, and Jon Barker, "Channel Selection using Neural Network Posterior Probability for Speech Recognition with Distributed Microphone Arrays in Everyday Environments," in *CHiME Workshop*, 2018, pp. 19–24.

[30] Gautam Bhattacharya, Jahangir Alam, and Patrick Kenny, "Deep Speaker Embeddings for Short-Duration Speaker Verification," in *Interspeech*, 2017, pp. 1517–1521.

[31] F A Rezaur Rahman Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan, "Attention-Based Models for Text-Dependent Speaker Verification," in *ICASSP*, 2018, pp. 5359–5363.

[32] Weicheng Cai, Zexin Cai, Xiang Zhang, Xiaoqi Wang, and Ming Li, "A Novel Learnable Dictionary Encoding Layer for End-to-End Language Identification," in *ICASSP*, 2018, pp. 5189–5193.

[33] Jinkun Chen, Weicheng Cai, Danwei Cai, Zexin Cai, Haibin Zhong, and Ming Li, "End-to-end Language Identification using NetFV and NetVLAD," in *ISCSLP*, 2018, pp. 319–323.

[34] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level Aggregation For Speaker Recognition In The Wild," in *ICASSP*, 2019, pp. 5791–5795.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016, pp. 770–778.

[36] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484 [cs]*, 2015.

[37] Weicheng Cai, Jinkun Chen, Jun Zhang, and Ming Li, "On-the-Fly Data Loader and Utterance-Level Aggregation for Speaker and Language Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1038–1051, 2020.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.