Interspeech 2022
18-22 September 2022, Incheon, Korea

Attentive Feature Fusion for Robust Speaker Verification

Bei Liu, Zhengyang Chen, Yanmin Qian[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{beiliu, zhengyang.chen, yanminqian}@sjtu.edu.cn

Abstract

As the most widely used technique, deep speaker embedding learning has become predominant in speaker verification task recently. This approach utilizes deep neural networks to extract fixed dimension embedding vectors which represent different speaker identities. Two network architectures such as ResNet and ECAPA-TDNN have been commonly adopted in prior studies and achieved the state-of-the-art performance. One omnipresent part, feature fusion, plays an important role in both of them. For example, shortcut connections are designed to fuse the identity mapping of inputs and outputs of residual blocks in ResNet. ECAPA-TDNN employs the multi-layer feature aggregation to integrate shallow feature maps with deep ones. Traditional feature fusion is often implemented via simple operations, such as element-wise addition or concatenation. In this paper, we propose a more effective feature fusion scheme, namely **A**ttentive **F**eature **F**usion (AFF), to render dynamic weighted fusion of different features. It utilizes attention modules to learn fusion weights based on the feature contents. Additionally, two fusion strategies are designed: sequential fusion and parallel fusion. Experiments on Voxceleb dataset show that our proposed attentive feature fusion scheme can result in up to 40% relative improvement over the baseline systems.

Index Terms: speaker verification, deep speaker embedding learning, feature fusion

1. Introduction

Speaker verification (SV) is a task to verify a person's claimed identity based on their voice characteristics. Given two utterances, a typical SV system can extract speaker embeddings and automatically determine whether two utterances belong to the same speaker or not. In general, two parts exist in a SV system. One is an embedding extractor which is used to extract speaker embedding from variable-length utterances. The other is similarity scorer based on the extracted embeddings. Before the era of deep learning, i-vector [1] along with probabilistic linear discriminant analysis (PLDA) [2] is the most popular method in the speaker verification field.

Recent years have witnessed the wide application of deep embedding learning in this task and the state-of-the-art performance has been obtained by DNN-based methods [3, 4, 5, 6, 7, 8, 9, 10, 11]. Deep learning-based systems typically consist of three main components: a frame-level feature extractor, a segment-level embedding aggregator and a speaker classifier [5]. Given an utterance, the neural network firstly extracts high-level feature representation. Then a pooling layer aggregates the frame-level representation across the temporal dimension and projects the pooled vector into a low-dimensional speaker embedding.

Mostly used network architectures in SV task include convolutional neural network such as ResNet [12] and time-delay neural network such as ECAPA-TDNN [10]. Recently, many efforts have been made in architectural improvements and optimization procedures to further improve the performance. Apart from network re-designs, in this paper, we focus on an omnipresent component of network architectures used in SV task, i.e. the feature fusion, to further boost the representation power of SV systems. Whether explicitly or implicitly, intentionally or unintentionally, feature fusion is an indispensable part in DNN-based SV systems. For instance, ResNet employs shortcut connections to fuse the identity mapping features and residual learning features. In ECAPA-TDNN, a multi-layer feature aggregation module is utilized to integrate shallow features with deep ones. However, the current feature fusion schemes used in SV systems such as element-wise addition or direct concatenation are fixed and non-learnable, which lacks the ability of modeling dynamic interactions between features.

To deal with the limitations of the traditional feature fusion described above, this paper introduces a novel and more effective feature fusion scheme called **A**ttentive **F**eature **F**usion (AFF) for speaker verification task. Compared to fixed and non-learnable feature fusion schemes, attentive feature fusion is designed to dynamically fuse different features, where attention modules are employed to learn fusion weights based on the contents of features. Moreover, two different fusion strategies are presented to model dynamic interactions between features, including sequential fusion and parallel fusion. Experiments conducted on Voxceleb [13, 14] demonstrate that our proposed attentive feature fusion scheme can lead to significant improvements over the baseline systems.

2. Related Work

Deep Speaker Embedding Learning: d-vector [3] is the first attempt to investigate the use of deep neural networks (DNNs) for a small footprint text-dependent speaker verification task. Subsequently, four types of deep features are introduced and used in a tandem fashion in [4]. Recently, time-delay neural network (TDNN) and convolutional neural network (CNN) are most-commonly used architectures. x-vector [6] is a famous TDNN-based deep speaker embedding extractor, which provides solid performance in the SV field. Furthermore, ECAPA-TDNN [10] makes several architectural modifications upon vanilla x-vector and achieves the state-of-the-art performance. In the meantime, [9] adopts ResNet as the speaker embedding extractor in VoxSRC 2019, which also releases a strong baseline on Voxceleb [13, 14].

Attention Modules: Attention module has been exten-

[†] corresponding author

10.21437/Interspeech.2022-478

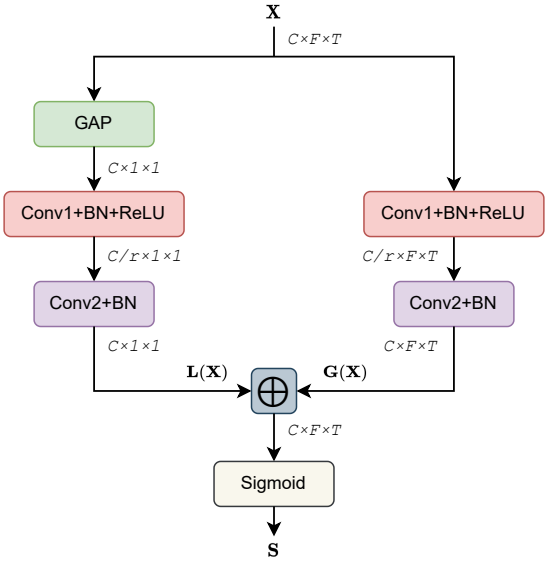

Figure 1: *Illustration of MS-CAM.*

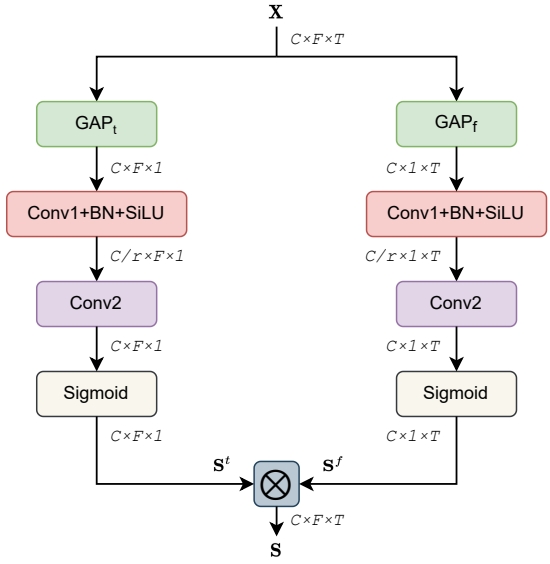

Figure 2: *Illustration of CA.*

sively adopted in a variety of deep learning tasks. In computer vision, a series of attention modules have been put forward. The squeeze-and-excitation (SE) module [15] simply squeezes global spatial information into a channel descriptor to capture channel-wise dependencies. Multi-scale channel attention module (MS-CAM) [16] aggregates the local and global context information along the channel dimension. Coordinate attention (CA) [17] encodes both channel relationships and long-range dependencies with precise positional information. In this paper, attention modules are utilized to learn fusion weights in our proposed attentive feature fusion scheme.

3. Proposed Method

In this section, we present details of the proposed attentive feature fusion (AFF) scheme and its application in ResNet architecture.

3.1. Attention Modules

In our proposed attentive feature fusion scheme, attention modules are adopted to learn fusion weights based on the contents of features. We study two different attention mechanisms for ResNet architecture, namely MS-CAM [16] and CA [17], in the experiments. Fig. 1 and Fig. 2 schematically depicts the overview of them.

MS-CAM: As shown in Fig. 1, MS-CAM aggregates the multi-scale context information along the channel dimension by varying the spatial pooling size. Specifically, local and global contexts are explored inside the attention module. For an input feature $\mathbf{X} \in \mathbb{R}^{C\times F\times T}$ where C, F and T represent the channel, frequency and time dimension respectively, the local channel context $\mathbf{L}(\mathbf{X}) \in \mathbb{R}^{C\times F\times T}$ is computed via a bottleneck structure as follows:

$$\mathbf{L}(\mathbf{X}) = \mathcal{B}(\text{Conv2}(\text{ReLU}(\mathcal{B}(\text{Conv1}(\mathbf{X}))))) \tag{1}$$

where Conv1 and Conv2 are point-wise convolution with output channel sizes of C/r and C respectively. r is the channel reduction ratio. $\mathcal{B}$ stands for BatchNorm. ReLU is the non-linear function.

Similarly, the global channel context $\mathbf{G}(\mathbf{X}) \in \mathbb{R}^{C\times F\times T}$ is obtained by:

$$\mathbf{G}(\mathbf{X}) = \mathcal{B}(\text{Conv2}(\text{ReLU}(\mathcal{B}(\text{Conv1}(\text{GAP}(\mathbf{X})))))) \tag{2}$$

where GAP is the global average pooling.

Given the local channel context $\mathbf{L}(\mathbf{X})$ and global channel context $\mathbf{G}(\mathbf{X})$, the attention map $\mathbf{S} \in \mathbb{R}^{C\times F\times T}$ can be calculated by:

$$\mathbf{S} = \sigma(\mathbf{L}(\mathbf{X}) \oplus \mathbf{G}(\mathbf{X})) \tag{3}$$

where $\oplus$ denotes the broadcasting addition. σ is the sigmoid function.

The attention map $\mathbf{S}$ is used as the fusion weights when implementing attentive feature fusion in the following section.

CA: [17] firstly proposes CA in order to embed direction-aware information, where channel attention is factorized into two parallel one-dimensional feature encoding processes to effectively integrate spatial coordinate information into the generated attention maps. As Fig. 2 shows, given an input feature $\mathbf{X} \in \mathbb{R}^{C\times F\times T}$, two attention maps are separately generated along the temporal and frequency directions respectively. For the temporal attention map $\mathbf{S}^t \in \mathbb{R}^{C\times F\times 1}$, the calculation process is presented below:

$$\mathbf{S}^t = \sigma(\text{Conv2}(\text{SiLU}(\mathcal{B}(\text{Conv1}(\text{GAP}_\text{t}(\mathbf{X})))))) \tag{4}$$

where GAP_t is one-dimensional global average pooling along the temporal dimension. SiLU [18] is the non-linear function.

Similarly, the frequency attention map $\mathbf{S}^f \in \mathbb{R}^{C\times 1\times T}$ is computed via:

$$\mathbf{S}^f = \sigma(\text{Conv2}(\text{SiLU}(\mathcal{B}(\text{Conv1}(\text{GAP}_\text{f}(\mathbf{X})))))) \tag{5}$$

where GAP_f is one-dimensional global average pooling along the frequency dimension.

The final attention map $\mathbf{S} \in \mathbb{R}^{C\times F\times T}$ is obtained by:

$$\mathbf{S} = \mathbf{S}^t \otimes \mathbf{S}^f \tag{6}$$

where $\otimes$ represents the broadcasting multiplication.

Similar to MS-CAM, this final attention map $\mathbf{S}$ is also adopted as the fusion weights in our proposed attentive feature fusion scheme.

3.2. Attentive Feature Fusion

Inspired by [16], we design two different strategies for attentive feature fusion (AFF): sequential AFF (S-AFF) and parallel AFF (P-AFF). Fig. 3 is the overview of our proposed methods.

S-AFF: As Fig. 3 shows, given two feature maps $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{C\times F\times T}$, S-AFF firstly adds $\mathbf{X}$ and $\mathbf{Y}$ in an element-wise

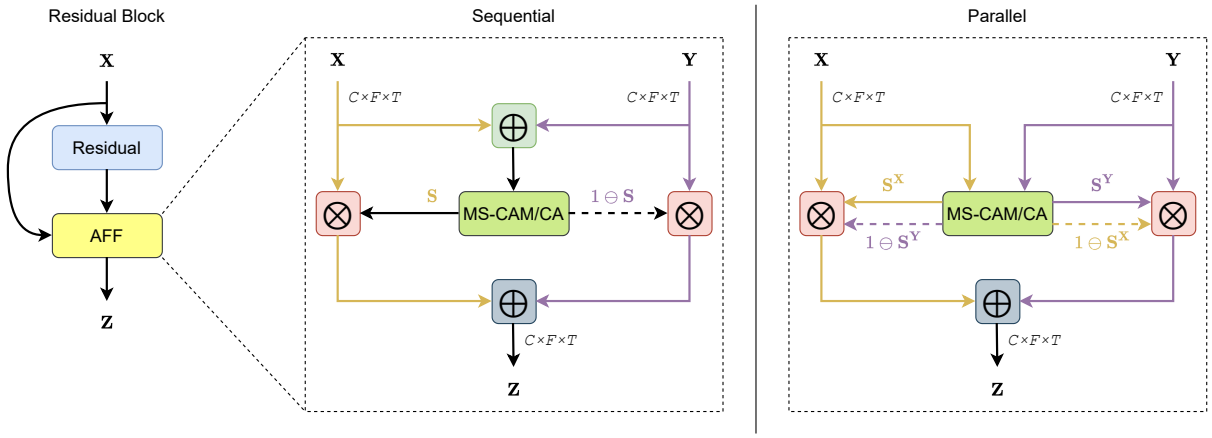

Figure 3: *ResNet-based attentive feature fusion. We design two different strategies when implementing attentive feature fusion.* ***Sequential:*** *two features are added first. Then the resulting feature is fed into the attention module to generate fusion weights.* ***Parallel:*** *feed two features into the attention module in parallel and generate fusion weights separately.*

manner. Then MS-CAM or CA takes the resulting feature as input and generates the attention map $\mathbf{S}$ as fusion weights. Subsequently, the original $\mathbf{X}$ and $\mathbf{Y}$ are scaled by the attention map $\mathbf{S}$ and the broadcasting subtraction of $\mathbf{S}$ respectively. Finally, the element-wise addition of weighted features is the attentively fused feature $\mathbf{Z} \in \mathbb{R}^{C \times F \times T}$. S-AFF can be expressed as:

$$\mathbf{S} = \text{MS-CAM/CA}(\mathbf{X} + \mathbf{Y}) \tag{7}$$

$$\mathbf{Z} = \mathbf{S} \otimes \mathbf{X} + (1 \ominus \mathbf{S}) \otimes \mathbf{Y} \tag{8}$$

where MS-CAM/CA represents the attention module introduced in Section 3.1. $\mathbf{S}$ is the attention map generated by MS-CAM/CA. $\ominus$ is the broadcasting subtraction. $\otimes$ denotes the element-wise multiplication.

P-AFF: Similarly, for two feature maps $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{C \times F \times T}$, P-AFF firstly feeds them into MS-CAM or CA in parallel and computes the attention map $\mathbf{S}^{\mathbf{X}}$ and $\mathbf{S}^{\mathbf{Y}}$ separately. Then the resulting attention maps are used to weight the original $\mathbf{X}$ and $\mathbf{Y}$. The calculation process is as follows:

$$\mathbf{S}^{\mathbf{X}} = \text{MS-CAM/CA}(\mathbf{X}) \tag{9}$$

$$\mathbf{S}^{\mathbf{Y}} = \text{MS-CAM/CA}(\mathbf{Y}) \tag{10}$$

$$\mathbf{Z} = \mathbf{S}^{\mathbf{X}} \otimes \mathbf{X} \otimes (1 \ominus \mathbf{S}^{\mathbf{Y}}) + (1 \ominus \mathbf{S}^{\mathbf{X}}) \otimes \mathbf{Y} \otimes \mathbf{S}^{\mathbf{Y}} \tag{11}$$

where $\mathbf{S}^{\mathbf{X}}$ and $\mathbf{S}^{\mathbf{Y}}$ are the generated attention maps based on the contents of $\mathbf{X}$ and $\mathbf{Y}$ respectively.

3.3. Application in ResNet

To validate the proposed attentive feature fusion scheme, we apply it to ResNet, which is a commonly-used architecture in SV task. The original feature fusion method in ResNet is the element-wise addition between the identity mapping feature and the residual feature. As shown in Fig. 3, alternatively, we apply AFF to ResNet by simply replacing the original addition with the proposed AFF in every residual block. In addition, it is worth mentioning that our proposed AFF module is very lightweight and efficient. The performance can be significantly improved over the baseline systems with only a slight increase in parameter.

4. Experimental Setup

4.1. Dataset and Data Augmentation

Our experiments are conducted on Voxceleb1&2 [13, 14] to evaluate the proposed methods, where the development set of Voxceleb2 is adopted as training data and the whole Voxceleb1 is used as the testing data. Specifically, performance is evaluated on three official trial lists: Vox1-O, Vox1-E and Vox1-H. Plus, three data augmentation techniques are utilized to improve the robustness of systems.

- Online Data Augmentation [19]: Extra data samples are generated by adding noise or reverberation from MUSAN dataset [20] and RIR dataset [21] to original training utterances in an online manner.
- SpecAugment [22]: We add frequency and time-steps masking to the input acoustic features.
- Speed Perturb [23]: We use sox to speed up or down each utterance by 0.9 or 1.1 times. Finally, the training data has 1,092,009×3 = 3,276,027 utterances from 5,994×3 = 17,982 speakers.

4.2. Training Details

We use 80-dimensional filter bank with 25ms windows and 10ms shift as the input acoustic features. All systems are trained on 200-frame chunks which are randomly cropped from training utterances. In addition, AAM-softmax [24] with a margin of 0.2 and a scale of 32 is adopted as the loss function. The stochastic gradient descent (SGD) with momentum of 0.9 and weight decay of $1e$-4 is employed as optimizer to train models. The training epoch number is 165 with the learning rate exponentially decreasing from 0.1 to $1e$-5.

4.3. Evaluation Metrics

During testing, we use cosine distance as the scoring criterion. Subsequently, all the scores are normalized using adaptive score normalization (AS-Norm) [25, 26] where the size of the imposter cohort is set to 600. Performance is measured in terms of the equal error rate (EER) and the minimum detection cost function (MinDCF) with the settings of $P_{target} = 0.01$ and $C_{FA} = C_{Miss} = 1$.

5. Results and Analysis

The results of the baseline systems and our proposed attentive feature fusion (AFF) systems are listed in Table 1. We adopt ResNet18 and ResNet34 as the baselines individually. Two different fusion strategies (S-AFF and P-AFF) are implemented based on MS-CAM and CA respectively. It can be obviously observed that our proposed AFF module can significantly improve the performance over the baselines with only a slight in-

Table 1: *Results comparison of different systems on the Voxceleb1 dataset in terms of EER and MinDCF.*

Architecture	Fusion Strategy	# Params	Voxceleb-O		Voxceleb-E		Voxceleb-H	
			EER(%)	MinDCF	EER(%)	MinDCF	EER(%)	MinDCF
ResNet18	—	4.11M	1.48	0.1737	1.52	0.1751	2.72	0.2444
+S-AFF(MS-CAM)	Sequential	+0.18M	1.29	0.1520	1.36	0.1613	2.49	0.2370
+S-AFF(CA)		+0.13M	0.93	0.0942	1.05	0.1211	1.94	0.1871
+P-AFF(MS-CAM)	Parallel	+0.36M	1.19	0.1444	1.29	0.1543	2.37	0.2275
+P-AFF(CA)		+0.26M	**0.86**	**0.0894**	**0.99**	**0.1138**	**1.82**	**0.1796**
ResNet34	—	6.63M	0.96	0.0885	1.01	0.1206	1.86	0.1769
+S-AFF(MS-CAM)	Sequential	+0.33M	0.79	0.0823	0.89	0.1099	1.71	0.1652
+S-AFF(CA)		+0.24M	0.65	0.0605	0.82	0.1006	1.59	0.1538
+P-AFF(MS-CAM)	Parallel	+0.66M	0.75	0.0814	0.84	0.1076	1.68	0.1641
+P-AFF(CA)		+0.48M	**0.62**	**0.0599**	**0.79**	**0.1001**	**1.57**	**0.1531**

crease in parameter. In particular, ResNet34-P-AFF(CA) system achieves a new state-of-the-art performance on Voxceleb1, which demonstrates the superiority of the proposed attentive feature fusion scheme over traditional ones.

5.1. Sequential Attentive Feature Fusion

For sequential attentive feature fusion (S-AFF), we examine the effect of different attention modules based on MS-CAM and CA respectively. Notably, both MS-CAM and CA are light-weight, which merely increase the parameter slightly. And CA based S-AFF can achieve much better performance than MS-CAM based S-AFF, and meawhile has fewer parameters. Specifically, for CA based S-AFF, the relative improvements in EER by 37.2%, 30.9%, 28.7% and in MinDCF by 46.8%, 31.9%, 23.5% are obtained with ResNet18 system in the three official trial lists. Similarly, ResNet34-S-AFF(CA) decreases the EERs to 0.65%, 0.82% and 1.59% on Vox1-O, Vox1-E and Vox1-H respectively. We attribute the effectiveness of CA based S-AFF to the fact that CA has the ability of encoding coordinate-aware information along frequency and time dimension separately. Some evidences [27, 28] have shown that it is crucial for speaker verification to model the frequency and time domain of a spectrogram separately instead of treating them equally.

5.2. Parallel Attentive Feature Fusion

Also, MS-CAM and CA are implemented for parallel attentive feature fusion (P-AFF). The increase in the parameter size of P-AFF is twice as much as that of S-AFF. Meanwhile, P-AFF leads to better performance than S-AFF. Likewise, CA based P-AFF outperforms MS-CAM based P-AFF by a large margin which can result in the relative improvements in EER by 41.9%, 34.9%, 33.1% and in MinDCF by 48.6%, 35.1%, 26.6% over ResNet18. Additionally, ResNet34-P-AFF(CA) achieves a new state-of-the-art result on Voxceleb1, namely 0.62%, 0.79% and 1.57% EER on the three official trial lists.

5.3. Comparison with Other Systems

In this section, we present a comprehensive comparison between the proposed method and four different types of systems from recent works [9, 10, 27, 29, 30, 31, 32] on VoxCeleb1. According to the embedding extractor architecture, other systems are divided into ResNet-based, TDNN-based, Transformer-based and MLP-based. As Table 2 shows, our best system ResNet34-P-AFF(CA), i.e. ResNet with the proposed parallel attentive feature fusion, outperforms all listed methods by a large margin, especially on Vox1-E and Vox1-H, which reveals that the proposed attentive feature fusion scheme is effective and powerful.

Table 2: *Comparison with other systems on Voxceleb1.*

System	Vox1-O	Vox1-E	Vox1-H
ResNet-based			
ResNet34 [9]	1.46	1.55	2.76
ResNet34-ft-CBAM [27]	1.08	1.43	2.67
ResNet34-DTCF [29]	0.79	1.13	2.09
TDNN-based			
ECAPA(C=512) [10]	1.01	1.24	2.32
ECAPA(C=1024) [10]	0.87	1.12	2.12
Transformer-based			
SAEP [30]	2.91	2.87	4.75
GCSA [31]	1.96	2.07	3.65
MLP-based			
MLP-SVNet [32]	1.36	1.46	2.49
ResNet34-P-AFF(CA)	**0.62**	**0.79**	**1.57**

6. Conclusions

In this paper, we introduce a novel attentive feature fusion (AFF) scheme to replace the conventional feature fusion method for DNN-based speaker verification. Two different fusion strategies are elaborately designed, including sequential AFF (S-AFF) and parallel AFF (P-AFF), where we utilize MS-CAM and CA attention module to learn fusion weights. Compared to the conventional feature fusion, the proposed AFF is dynamic and learnable. Experiments on Voxceleb dataset demonstrate the efficiency and superiority of our proposed method, which can lead to significant improvements over the baselines consistently with only a slight increase in parameter.

7. Acknowledgement

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

8. 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, 2011.

[2] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 531–542.

[3] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[4] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[5] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 999–1003.

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

[7] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[8] D. Garcia-Romero, A. McCree, D. Snyder, and G. Sell, "Jhu-hltcoe system for the voxsrc speaker recognition challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7559–7563.

[9] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[10] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 3830–3834.

[11] B. Liu, H. Wang, Z. Chen, S. Wang, and Y. Qian, "Self-knowledge distillation via feature enhancement for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2616–2620.

[14] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: deep speaker recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1086–1090.

[15] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[16] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 3559–3568.

[17] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 713–13 722.

[18] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *arXiv preprint arXiv:1702.03118*, 2017.

[19] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 28, pp. 1038–1051, 2020.

[20] D. Snyder, G. Chen, and D. Povey, "Musan: a music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[21] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[22] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: a simple data augmentation method for automatic speech recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2613–2617.

[23] W. Wang, D. Cai, X. Qin, and M. Li, "The dku-dukeece systems for voxceleb speaker recognition challenge 2020," *arXiv preprint arXiv:2010.12731*, 2020.

[24] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2873–2877.

[25] Z. N. Karam, W. M. Campbell, and N. Dehak, "Towards reduced false-alarms using cohorts," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4512–4515.

[26] S. Cumani, P. Batzu, D. Colibro, C. Vair, P. Laface, and V. Vasilakakis, "Comparison of speaker recognition approaches for real applications," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 2365–2368.

[27] S. Yadav and A. Rai, "Frequency and temporal convolutional attention for text-independent speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6789–6793.

[28] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in tdnns and frequency positional information in 2d resnets to enhance speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2021, pp. 2302–2306.

[29] L. Zhang, Q. Wang, and L. Xie, "Duality temporal-channel-frequency attention enhanced speaker representation learning," *arXiv preprint arXiv:2110.06565*, 2021.

[30] P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 941–945.

[31] B. Han, Z. Chen, and Y. Qian, "Local information modeling with self-attention for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[32] B. Han, Z. Chen, B. Liu, and Y. Qian, "Mlp-svnet : a multi-layer perceptrons based network for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.