INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

ACA-Net: Towards Lightweight Speaker Verification using Asymmetric Cross Attention

Jia Qi Yip[13], Duc-Tuan Truong[2], Dianwen Ng[12], Chong Zhang[1], Yukun Ma[1], Trung Hieu Nguyen[1], Chongjia Ni[1], Shengkui Zhao[1], Eng Siong Chng[2], Bin Ma[1]

[1]Speech Lab of DAMO Academy, Alibaba Group
[2]SCSE, Nanyang Technological University (NTU), Singapore
[3] Alibaba-NTU Singapore JRI, Interdisciplinary Graduate Programme, NTU, Singapore.
jiaqi006@e.ntu.edu.sg

Abstract

In this paper, we propose ACA-Net, a lightweight, global context-aware speaker embedding extractor for Speaker Verification (SV) that improves upon existing work by using Asymmetric Cross Attention (ACA) to replace temporal pooling. ACA is able to distill large, variable-length sequences into small, fixed-sized latents by attending a small query to large key and value matrices. In ACA-Net, we build a Multi-Layer Aggregation (MLA) block using ACA to generate fixed-sized identity vectors from variable-length inputs. Through global attention, ACA-Net acts as an efficient global feature extractor that adapts to temporal variability unlike existing SV models that apply a fixed function for pooling over the temporal dimension which may obscure information about the signal's non-stationary temporal variability. Our experiments on the WSJ0-1talker show ACA-Net outperforms a strong baseline by 5% relative improvement in EER using only 1/5 of the parameters.
Index Terms: Speaker Verification, Asymmetric Cross Attention, Lightweight

1. Introduction

Speaker Verification (SV) is the task of determining if a given speech segment belongs to a claimed enrolled speaker. This task is typically achieved by the comparison of fixed-length speaker embeddings computed from variable-length utterances. For speaker verification, a speaker embedding extractor must produce close embeddings for different utterances of the same speaker, and distant embeddings for utterances from different speakers. In addition, speaker embeddings can also be used in other speech domains such as speaker diarization [1] and speaker extraction [2] to create a speaker targeted frontend to ASR models like [3, 4] or for keyword spotting [5].

Temporal statistics pooling [6, 7, 8] is commonly used by embedding extraction networks in SV models to handle variable input lengths. Temporal statistics pooling refers to the channel-wise pooling of a model's embedding vector, commonly by taking the mean, max or standard deviation of all time steps in that channel, to obtain a single representative value for that channel. However, depending on the statistics used, the pooling method may obscure variability across time steps that may be important in discriminating between speakers. Additionally, statistics pooling assumes that the speech signal has statistical properties that remain stationary over time, which may not always hold true. Recent models such as RawNet3 [9], LargeResNet-MagFace [10], MFA-Conformer [11] and ECAPA-TDNN [12] have used context-aware or attentive statistics pooling [13] to vary the weight of each time-step during the pooling operation.

Besides temporal pooling, it has been shown that embedding extractors benefit from having global information even while modeling the local features in each time step [11]. For fully convolutional models, the Squeeze-and-Excitation layer [14] in the commonly used Res2Net block [15] was used to bring global information into the model. More recently, the transformer architecture was used to encode global information in MFA-Conformer [11]. Elsewhere, transformers [16] have also been applied successfully in many speech applications such as speech separation [17], and speech enhancement [18]. Nevertheless, a drawback of using transformers is the high cost of computing self-attention over large matrices.

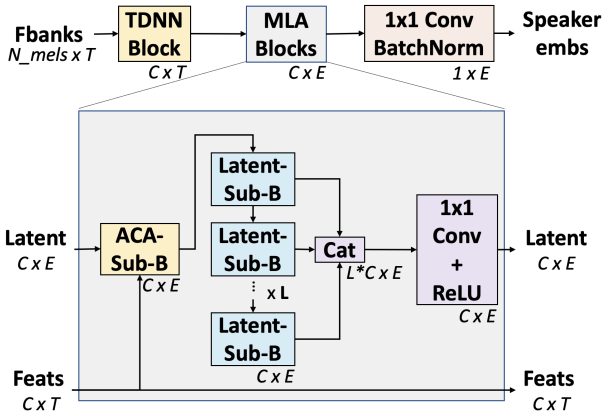

Figure 1: *Overall architecture of ACA-Net. The model consists of a single 1x1 TDNN block, followed by the Multi-Layer Aggregation (MLA) block. The details of the MLA block are shown in the bottom part of the figure. The MLA block accepts a Latent and Features (Feats) as inputs and outputs a new Latent vector. The Features remain unchanged by the block. The ACA- and Latent-Sub-Blocks (Sub-B) are detailed in Figure 3.*

Here we propose ACA-Net, as shown in Figure 1, which uses Asymmetric Cross Attention (ACA) [19, 20] to avoid the high computational cost of self-attention while eliminating the need for temporal pooling. ACA computes attention between a small latent query and a large feature sequence as the key and value matrices, shown in Figure 2. This distils the temporal dimension of the feature input down to the embedding dimension. ACA-Net is thus a lightweight, computationally efficient model with strong global context modeling which can capture more fine-grained information about the speech signal [17] and provides better discrimination between speakers [11]. The obtained latent is then refined through Multi-Layer Aggregation (MLA) over multiple self-attention sub-blocks, further enhancing speaker verification performance. Experiments on WSJ0-1talker [2] show that the proposed ACA-Net surpasses strong baselines such as ECAPA-TDNN [12] and RawNet3 [9] while using only 1/5 of the parameters.

10.21437/Interspeech.2023-1725

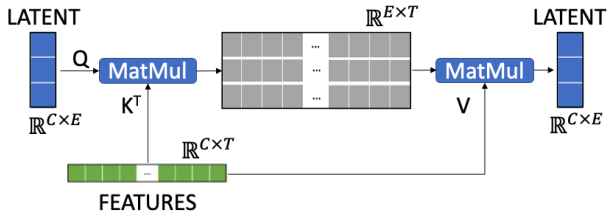

Figure 2: *Illustration of Asymmetric Cross Attention (ACA). Applying the $(QK^T)V$ series of matrix multiplications (MatMul) in the attention operation using a fixed-sized Query (Q) with variable length Key (K) and Values (V) always results in an output of the same dimensions as Q.*

While ACA-based methods have recently been used in a variety of domains, including natural language processing [21, 22], remote sensing [23], image-text matching [24] and most notably in the Perceiver class of neural networks [20, 25, 26, 27], to our knowledge ACA-Net is the first speaker embedding extractor to use ACA instead of the temporal pooling methods commonly used in current SV models.

2. Methodology

The overall architecture of the proposed ACA-Net[1] is shown in Figure 1. The model accepts audio input processed through a filterbank and consists of a single TDNN block followed by the Multi-Layer Aggregation (MLA) Block and a final 1x1 convolution used to reduce the channel dimension back to 1 for the final embedding.

2.1. TDNN Block

The TDNN block used after the filterbank layer follows the implementation of [12] in [28]. The TDNN block in ACA-Net consists of a single depth-wise 1D Convolutional layer, followed by ReLU activation and 1D batch normalization. The purpose of this block is to serve as a further feature extractor to decouple the number of filterbanks from the input channels to the MLA block.

2.2. Asymmetric Cross Attention

The ACA sub-block shown in Figure 3 makes use of the standard Multi-Head Attention (MHA) as per Pytorch, which can be defined as follows:

$$\begin{aligned} \text{MultiHead}(Q,K,V) &= \text{Concat}(head_1,\dots head_h)W^O \\ head_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ \text{Attention}(Q,K,V) &= \text{softmax}(\frac{QK^T}{\sqrt{d}})V \end{aligned} \tag{1}$$

where d denotes the number of channels and Q, K, V denote the Query, Key and Value of the MHA respectively. W_i^Q, W_i^K, W_i^V,W_i^O are the projection parameter matrices.

The standard transformer makes use of MHA where $Q \in \mathbb{R}^{C\times T}$, $K \in \mathbb{R}^{C\times T}$ and $V \in \mathbb{R}^{C\times T}$, whereas ACA uses a $Q \in \mathbb{R}^{C\times E}$ where $E \lll T$. Furthermore, E is the embedding size which is a fixed hyperparameter while T is the time dimension which is variable. C represents the number of channels. While the output of MHSA will be $\mathbb{R}^{C\times T}$ the output of ACA will be $\mathbb{R}^{C\times E}$, which is a much smaller latent vector. Importantly, the dimensions of the ACA output will be independent of T.

[1]The model is available at github.com/Yip-Jia-Qi/ACA-Net

2.3. ACA and Latent Sub-Blocks

As shown in Figure 3, the ACA sub-block and the latent sub-block share the same block architecture. The key difference between the sub-blocks is the dimensions of the K and V inputs. The Q, K and V for the latent sub-block are the latent produced by the ACA sub-block. The K and V for the ACA sub-block is the feature sequence while Q comes from random initialization. Additionally, for the ACA sub-block, sinusoidal positional encoding is added to the feature sequence before being passed into the MHA layer.

When given Feature input of dimensions $\mathbb{R}^{C\times T}$ the ACA sub-block reduces the time dimension to the embedding size resulting in an output of $\mathbb{R}^{C\times E}$. Meanwhile, since the latent sub-block performs self-attention on the latent, it results in no change in dimensions to the latent of $\mathbb{R}^{C\times E}$.

2.4. The MLA Block

The MLA block as shown at the bottom of Figure 1 represents our key contribution. It consists of a single ACA sub-block (ACA-Sub-B) followed by a variable number of latent sub-blocks. Both blocks share the same design shown in Figure 3. Finally, the concatenated outputs of the latent sub-blocks are passed into a depth-wise 1D convolution and batch normalization layer.

$$\begin{aligned} Latent_{init} &\sim \mathcal{N}(\mu,\sigma^2) \\ Latent^{'} &= \text{ACA-Sub-B}(Latent_{init}, Features) \\ Latent^{''} &= \text{MLA}(Latent^{'}) \\ Latent_{out} &= \text{ReLU}(\text{Conv1D}(Latent^{''})) \end{aligned} \tag{2}$$

where $\mathcal{N}$ denotes a truncated normal distribution with mean of μ and standard deviation of σ^2. $Latent \in \mathbb{R}^{C\times E}$ is randomly initialised and $Features \in \mathbb{R}^{C\times T}$ refers to the output from the TDNN block.

The purpose of the ACA sub-block is to compute an initial latent while the latent sub-block (Latent-Sub-B) refines the latent. The latent is refined through MLA, where the latent vector is passed through multiple latent sub-blocks, with the output of each latent sub-block aggregated by concatenation along the channel dimension and passed through a depth-wise convolution at the end to return the channel dimension back to its original size. MLA can be described as follows:

$$\begin{aligned} &\text{MLA}(L) = \text{Conv1D}(\text{Concat}(Layer_1,\dots Layer_j)) \\ &\text{for } Layer_{j+1} = \text{Latent-Sub-B}(Layer_j) \end{aligned} \tag{3}$$

where $L = Layer_0 \in \mathbb{R}^{C\times E}$ and j denotes the number of latent sub-blocks.

This MLA is similar to the multi-scale feature aggregation method employed in [11] and [12] although the latents are not multi-scale since the ACA produces the latent using the full global context, resulting in only a single scale.

3. Experiments

3.1. Dataset

For all experiments, we train and evaluate the models on the WSJ0-1talker speaker verification dataset [2] which is drawn from the WSJ0 corpus [29]. This dataset is designed to test speaker verification performance for speaker embedding extractors to be used in speaker extraction systems such as [30], where

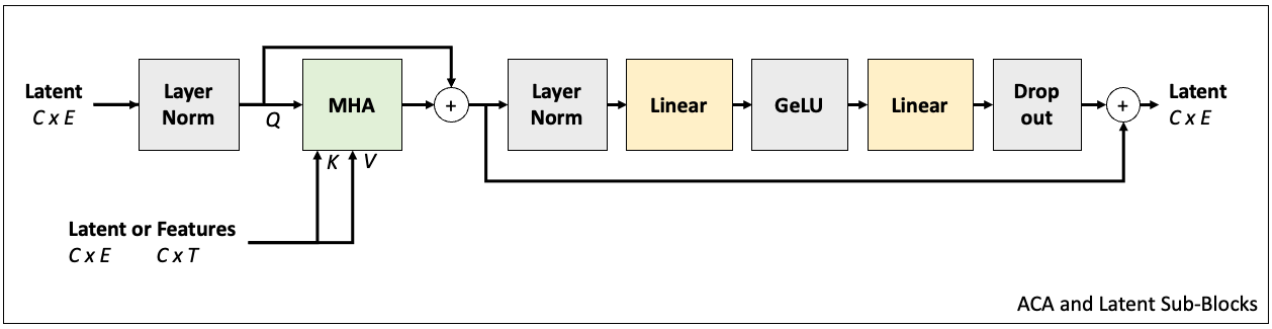

Figure 3: *Detailed view of the ACA- and Latent-Sub-Blocks within an MLA block represented as"ACA-Sub-B" and "Latent-Sub-B" in the overall architecture of ACA-Net shown in Figure 1. ACA- and Latent-Sub-Blocks share the same architecture but differ in their inputs to the Multi-head Attention (MHA) layer. The ACA sub-block passes a Latent into the Query (Q) and Features into the Key (K) and Value (V) of the MHA while the Latent-Sub-Block performs self-attention on the same Latent vector.*

a lightweight speaker embedding extractor like ACA-Net is important since the speaker embedding is auxiliary to the main speech separation network. Speaker extraction is commonly benchmarked on the WSJ0-2mix [31] dataset, which consists of mixtures generated from the WSJ0 Corpus [29].

The WSJ0-1talker dataset [2] consists of 101 speaker training and development sets drawn from the "si_tr_s" collection from the WSJ0 corpus [29] and a test set of 18 speakers drawn from the"si_dt_05" and "si_et_05" collections. While the training (20,000 utterances) and development (5,000 utterances) sets share speakers but have different utterances, the testing (3,000 utterances) set consists of 18 separate speakers unseen during training. The verification pairs for testing are randomly selected from the training set. All utterances were down-sampled from 48kHz to 8kHz.

3.2. Experimental Setup

All models were trained for 25 epochs using the Adam optimizer [32] with a Cyclical Learning Rate Scheduler [33] with a base learning rate of 10^{-7} and a maximum learning rate of 10^{-2}. For larger models, we use a maximum learning rate of 10^{-3} for better stability. The batch size was set to 32 or reduced to 16 on larger models due to memory limitations. All experiments were done on 1 GPU with 16GB RAM.

The loss function used for all model training was Additive Angular Margin (AAM), or ArcFace [34] with a margin of 0.2 and a scale of 30. During Testing, use the Equal Error Rate (EER) metric and the minimum detection cost function (minDCF) metric to measure performance. The minDCF metric is calculated with the hyperparameters P_{target}=0.01 and $C_{falsealarm}$ = C_{miss}=1. AAM, EER and minDCF functions are implemented by the Speechbrain [28] training framework.

3.3. Model Hyperparameters

We train two baseline models, ECAPA-TDNN [12] and RawNet3 [9], which have not previously been reported on the WSJ0-1talker dataset, in addition to ACA-Net. All models have been trained on the Speechbrain [28] framework.

ECAPA-TDNN. The ECAPA-TDNN model [12] is a recent state-of-the-art model incorporating time-delay neural networks (TDNN) and Multi-scale Feature Aggregation across three layers. We use the existing implementation of the ECAPA-TDNN model included as part of the Speechbrain [28] framework. All model hyperparameters are set per the defaults in Speechbrain.

RawNet3. To train the RawNet3 model we made minimal modifications to the code provided by the authors of [9] for it to work in Speechbrain. We train two versions of the models by adjusting the "C" hyperparameter which controls the number of channels in the convolutional layers of the model. The original model in [9] has C=1024 while we train a smaller version with C=512 for a model that is closer in parameter size to ACA-Net.

ACA-Net. The base ACA-Net model consists of 1 MLA block with 1 ACA sub-block and 3 latent sub-blocks. The embedding size of the base ACA-Net model is 512. Throughout all experiments, the dropout of the sub-blocks is set to 0.2, the channel dimension is fixed at 256 with the size of all linear layers is set to 1024. Input features are derived using a filter bank with 80 filters, a hop length of 10 and a window length of 25. Positional Encoding is added to the features using a standard sinusoidal positional encoding function. Where applicable, the initial latent for ACA layers within the ACA sub-blocks were initialized according to a truncated normal distribution with mean 0, standard deviation 0.02, and truncation bounds [-2, 2]

Table 1: *Performance comparison of ACA-Net against other popular SV models on the WSJ0-1talker verification test set. Number of parameters for the model are reported where available. SV-T and RawNet3 are time domain models while the rest operate in the frequency domain.*

Model	Params (M)	EER↓ (%)	minDCF↓
x-vector [35]	-	5.87	0.69
SV-T [2]	-	4.40	0.45
SV-F [2]	-	4.37	0.42
RawNet3 (C=512) [9]	6.5	3.94	0.38
RawNet3 (C=1024) [9]	16.3	3.46	0.38
ECAPA-TDNN [12]	20.8	2.99	0.32
SV-FA [2]	-	2.90	0.36
ACA-Net	3.6	**2.85**	**0.31**

3.4. Experimental Results

In Table 1 we report the verification performance of ACA-Net on the WSJ0-1talker dataset compared with two reimplemented baselines ECAPA-TDNN and RawNet3. Additionally, we also compare the results against existing baselines, x-vector PLDA, SV-T, SV-F, and SV-FA reported in [2], although the number of parameters for the existing baselines was not reported.

Based on the WSJ0-1talker dataset as shown in Table 1, ACA-Net achieves the lowest EER and minDCF out of all the models while using only 1/5 of the parameters of ECAPA-TDNN and RawNet3. Except for ACA-Net, all baseline models make use of some sort of temporal pooling method. The x-vector model, an older model from 2018 [6], unsurprisingly performs the worst without the benefit of recent innovations. When comparing the results of SV-T with SV-F as well as the results of ECAPA-TDNN and SV-FA with RawNet3, we find that the frequency domain approaches have an advantage over time domain models. This aligns with findings in [9]. Additionally, the result that the 512-channel version of RawNet3 with fewer parameters performs more poorly than the 1024-channel version suggests that the poorer performance of the larger baseline models, ECAPA-TDNN and RawNet3, is not simply due to the large models overfitting on the WSJ0-1talker dataset.

3.5. Ablation Study

We conduct an ablation study on various components of the design of ACA-Net to show their contribution to the performance of the model in Table 2. In the subtractive ablation experiments, we remove the concatenation step of the MLA block, leaving the output of the last latent sub-block to be passed through a depth-wise convolution with no change in dimension. We also experimented with removing the positional encoding of the features before the ACA sub-block as well as removing latent sub-block from the MLA block. In the additive ablation experiments, we used weight sharing across the 3 latent sub-blocks.

The drop in performance after the removal of MLA and latent sub-blocks validates the importance of these design features in the model. Since the model relies on attention, performance falls when the positional encoding of the features is removed because information about the relative positions of the input features is lost. The relative position of input features is important because speaker identity is determined through speech patterns that are only identifiable if they occur in sequence. Weight sharing across the latent sub-blocks, turns the latent refinement into a recursive processes, resulting in a 40% reduction in parameter size. While this is accompanied by a drop in performance, we note that this smaller model still achieves EER on par with the 1024-channel RawNet3 model from Table1.

Table 2: *Ablation study of ACA-Net with the decomposition of different components in terms of parameter sizes, EER and minDCF, respectively.*

Model	**Param (M)**	**EER↓ (%)**	**minDCF↓**
ACA-Net	3.6	2.85	0.31
- MLA	3.3	3.90	0.24
- Latent-Sub-Blocks	3.6	4.68	0.42
- Positional Encoding	3.6	4.68	0.47
+ weight sharing	2.0	3.46	0.45

Next, we conduct a series of ablation experiments on various model dimensions of the base ACA-Net. Specifically, we experiment with changing the number of latent sub-blocks (Figure 4) to determine if more latent sub-blocks can result in better performance and embedding size (Table 3) to determine if giving the model more space to place speakers can help it better differentiate speakers.

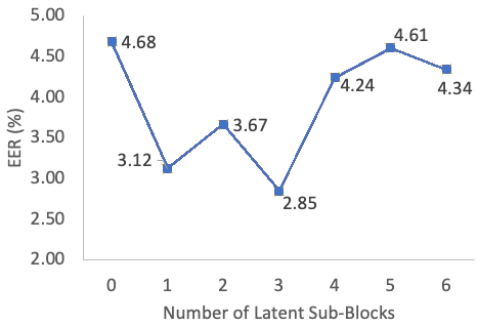

Figure 4: *Ablation Study over a number of latent sub-blocks in the base ACA-Net. Increasing the number of blocks increases the depth of the model.*

Based on the results shown in Figure 4, the number of latent sub-blocks for ACA-Net seems to be optimal at 3 sub-blocks, with significantly worse performance observed beyond 3 sub-blocks. One possibility is that after the 3rd sub-block, the additional parameters introduced do not contribute to the further refinement of the latent as there are now a too many layers between these deeper sub-blocks and the original features distilled by the ACA sub-block.

Table 3: *Verification performance of ACA-Net with different embedding sizes (i.e., $E = 256, 512, 1024$). Embedding size determines the length of the vector used by the model to discriminate between speakers during speaker verification.*

Model	**EER↓ (%)**	**minDCF↓**
ACA-Net (E=256)	5.16	0.32
ACA-Net (E=512)	2.85	0.31
ACA-Net (E=1024)	4.96	0.49

In SV, is has been shown that embedding dimension size is an important hyperparameter since it determines the volume of high-dimensional space available in the vector used to discriminate between speakers [36]. Increasing embedding vector size could improve discriminability by increasing the volume of space available for the model to encode speakers, however, having a space that is too large could also result in utterances from the same speaker being wrongly separated [36]. We see this effect in Table 3 where embedding sizes larger and smaller than the 512 used by the base ACA-Net result in worse performance.

4. Conclusion

Here we presented ACA-Net, a lightweight speaker embedding extractor for SV. ACA-Net is the first model to apply ACA to SV and achieves impressive performance by replacing temporal pooling with global feature extraction through attention. On the WSJ0-1talker dataset, ACA-Net outperforms strong baselines, ECAPA-TDNN and RawNet3, on both EER and minDCF, despite using only 1/5 of the parameters. Overall, our experiments highlight the potential of ACA as an alternative to typical temporal pooling methods.

5. Acknowledgements

This work was supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore.

6. References

[1] N. Dawalatabad, M. Ravanelli, F. Grondin, J. Thienpondt, B. Desplanques, and H. Na, "ECAPA-TDNN embeddings for speaker diarization," in *INTERSPEECH*. ISCA, 2021.

[2] C. Xu, W. Rao, J. Wu, and H. Li, "Target speaker verification with selective auditory attention for single and multi-talker speech," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29. IEEE/ACM, 2021, pp. 2696–2709.

[3] D. Ng *et al.*, "De'hubert: Disentangling noise in a self-supervised model for robust speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[4] C. Chen, Y. Hu, Q. Zhang, H. Zou, B. Zhu, and E. S. Chng, "Leveraging modality-specific representations for audio-visual speech recognition via reinforcement learning," in *AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence (AAAI), 2023.

[5] D. Ng *et al.*, "Contrastive speech mixup for low-resource keyword spotting," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[6] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*. IEEE, 2018.

[7] M. Rouvier, P.-M. Bousquet, and J. Duret, "Study on the temporal pooling used in deep neural networks for speaker verification," in *29th European Signal Processing Conference (EUSIPCO)*, 2021.

[8] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] J. weon Jung, Y. J. Kim, H.-S. Heo, B.-J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," in *INTERSPEECH*. ISCA, 2022.

[10] N. Kuzmin, I. Fedorov, and A. Sholokhov, "Magnitude-aware probabilistic speaker embeddings," in *The Speaker and Language Recognition Workshop (Odyssey 2022)*. ISCA, 2022.

[11] Z. Yang *et al.*, "MFA-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," in *INTERSPEECH*. ISCA, 2022.

[12] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *INTERSPEECH*. ISCA, 2020.

[13] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *INTERSPEECH*. ISCA, 2018.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE/CVF, 2018.

[15] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[16] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[17] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 21–25.

[18] D. de Oliveira, T. Peer, and T. Gerkmann, "Efficient transformer-based speech enhancement using long frames and stft magnitudes," in *INTERSPEECH*, 2022.

[19] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 3744–3753.

[20] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," in *International Conference on Machine Learning*, 2021.

[21] H. Wang, C. Deng, J. Yan, and D. Tao, "Asymmetric cross-guided attention network for actor and action video segmentation from natural language query," in *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE/CVF, 2019.

[22] Z. Ji, J. Hu, D. Liu, L. Y. Wu, and Y. Zhao, "Asymmetric cross-scale alignment for text-based person search," *IEEE Transactions on Multimedia*, pp. 1–11, 2022.

[23] X. Zhang, S. Cheng, L. Wang, and H. Li, "Asymmetric cross-attention hierarchical network based on cnn and transformer for bitemporal remote sensing images change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[24] D. Wu, H. Li, Y. Tang, L. Guo, and H. Liu, "Global-guided asymmetric attention network for image-text matching," *Neurocomputing*, vol. 481, pp. 77–90, 2022.

[25] J. Andrew *et al.*, "Perceiver IO: A general architecture for structured inputs & outputs," in *International Conference on Learning Representations*, 2022.

[26] C. Hawthorne *et al.*, "General-purpose, long-context autoregressive modeling with perceiverAR," in *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[27] J. Carreira *et al.*, "Hip: Hierarchical perceiver." arXiv preprint arXiv:2202.10890, 2022.

[28] M. Ravanelli *et al.*, "Speechbrain: A general-purpose speech toolkit." arXiv, 2021.

[29] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) complete LDC93S6A." in *Electronic Article*, 1993.

[30] C. Xu, W. Rao, E. S. Chng, and H. Li, "SpEx: Multi-scale time domain speaker extraction network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1370–1384, 2020.

[31] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 31–35.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, (ICLR)*, 2015.

[33] L. N. Smith, "Cyclical learning rates for training neural networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.

[34] J. Deng, J. Guo, J. Yang, N. Xue, I. Cotsia, and S. P. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[35] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016.

[36] W. Gu, A. Tandon, Y.-Y. Ahn, and F. Radicchi, "Principled approach to the selection of the embedding dimension of networks," *Nature Communications*, vol. 12, no. 1, jun 2021.