ConFusionformer: Locality-enhanced Conformer Through Multi-resolution Attention Fusion for Speaker Verification

Youzhi Tu, Man-Wai Mak*, Kong-Aik Lee, Weiwei Lin

Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong SAR

Abstract

Conformers are capable of capturing both global and local dependencies in a sequence. Notably, the modeling of local information is critical to learning speaker characteristics. However, applying Conformers to speaker verification (SV) has not witnessed much success due to their inferior locality modeling capability and low computational efficiency. In this paper, we propose an improved Conformer, ConFusionformer, to address these two challenges. For increasing model efficiency, the conventional Conformer block is modified by placing one feed-forward network between a self-attention module and a convolution module. The modified Conformer block has fewer model parameters, thus reducing the computation cost. The modification also enables a deeper network, boosting the SV performance. Moreover, multi-resolution attention fusion is introduced into the self-attention mechanism to improve locality modeling. Specifically, the restored map from a low-resolution attention score map produced by downsampled queries and keys is fused with the original attention score map to exploit the local information within the restored local region. The proposed ConFusionformer is shown to outperform the Conformer for SV on VoxCeleb, CNCeleb, SRE21, and SRE24, demonstrating the superiority of the ConFusionformer in speaker modeling.

Keywords: Speaker verification, Speaker embedding, Transformer, Conformer, Multi-resolution attention fusion

1. Introduction

Speaker verification (SV) aims to determine whether the identities of an enrollment utterance and a test utterance belong to the same speaker (Bai and Zhang, 2021; Tu et al., 2022). A typical SV system (see Fig. 1(a)) comprises a speaker embedding network (see Fig. 1(b)) for extracting fixed-dimensional speaker representations and a scoring back-end for computing the similarity between the enrollment and test embeddings. Usually, the speaker embedding method uses convolutional neural networks (CNNs) or Transformer encoders (Vaswani et al., 2017) to process the frame-level acoustic features (Snyder et al., 2018; Nagrani et al., 2020; Desplanques et al., 2020; Liu et al., 2022a; Wang et al., 2023; Heo et al., 2024; Wang et al., 2022; Han et al., 2022). On top of the frame-level processing, a temporal pooling operation is adopted to summarize the frame-level representations into an utterance-level embedding (Snyder et al., 2018; Okabe et al., 2018; Xie et al., 2019; Desplanques et al., 2020; Tu and Mak, 2022; Zhu et al., 2022). Softmax-based classification losses, such as AMSoftmax (Wang et al., 2018), are often employed to train the speaker embedding network.

CNNs have long been the dominant backbone architecture for speaker embedding. Benefiting from the inherent local connectivity and translation equivalence, a CNN can capture long-range dependencies of speech segments through progressive convolutions (Snyder et al., 2018). On the other hand, Transformers are inherently good at modeling global dependencies of feature vectors in a sequence, attributed to the self-attention mechanisms. However, compared with CNNs, Transformers are not widely applied in SV due to their unsatisfactory performance (Wang et al., 2022; Han et al., 2022). This situation is in contrast with natural language processing (NLP) where Transformers have been the mainstream architecture (Vaswani et al., 2017; Dai et al., 2019; Devlin et al., 2019; Radford et al., 2018; Raffel et al., 2020). In computer vision (CV), it is also acknowledged that Transformers have higher ceiling performance than CNNs even without large-scale pre-training (Dosovitskiy et al., 2021; Touvron et al., 2021a; Liu et al., 2021; d'Ascoli et al., 2021). The success of Transformers in NLP and CV motivates us to develop Transformer-based speaker embedding networks that can compete with or surpass the state-of-the-art CNN counterparts.

One challenge of applying Transformers to SV is that the standard self-attention mechanism does not have an explicit modeling of local dependencies. Although Cordonnier et al. (2020) theoretically proved that a self-attention layer with sufficient number of heads can have similar representation capability as a convolutional layer and suggested that the first few attention layers of a Transformer possess implicit convolutional locality, using vanilla Transformer layers for SV does not lead to satisfactory results in practice. Because the speaker characteristics are often reflected in local speech dynamics (Han et al., 2022), locality modeling is of vital importance in speaker embedding.

*Corresponding author
Email address: enmwmak@polyu.edu.hk (Man-Wai Mak)

There are mainly two strategies to improve the locality of Transformers. One strategy is to enhance the standard self-attention by either explicitly constraining the attention context within a local region of each query vector or by implicitly inducing soft convolution operations. The former is often called local attention (Ramachandran et al., 2019) or sliding window attention (Beltagy et al., 2020; Zaheer et al., 2020) and has been adopted in SV (Wang et al., 2022; Han et al., 2022). The latter can be achieved by regulating the relative positional attention (d'Ascoli et al., 2021; Shaw et al., 2018; Huang et al., 2019), which is inspired by the relationship between the attention operation and convolution (Cordonnier et al., 2020). Basically, this strategy aims to emphasize locality through improved attention operation and does not change the basic components of a Transformer block, i.e., a self-attention network (SAN) and a feed-forward network (FFN).

The second strategy is to introduce a hard convolutional modeling by incorporating a CNN into the conventional Transformer block. For instance, Wu et al. (2020) designed a parallel branch of self-attention and convolution to account for the global and local context information in sequences. Another widely-adopted example is the Conformer (Gulati et al., 2020). Rather than paralleling convolution with self-attention, a standard Conformer block cascades a self-attention layer with a CNN, sandwiched between a pair of FFNs. Conformers have achieved considerable success in automatic speech recognition (ASR) (Gulati et al., 2020; Prabhu et al., 2024) and have also obtained widespread attention in SV (Han et al., 2022; Zhang et al., 2022; Cai et al., 2023).

Although Conformers are superior to the conventional Transformers in locality modeling, they still face two challenges when used for speaker embedding. On the one hand, the vanilla Conformer block is not computationally efficient. Given an input sequence of length T and a standard Conformer block with an encoding dimension of D, the SAN would consume $O\left(4TD^2 + T^2D\right)$ multiply-accumulate operations (MAC), whereas the MAC of the two FFNs is about $O\left(16TD^2\right)$. For classical SV datasets such as VoxCeleb (Nagrani et al., 2020) and CNCeleb (Li et al., 2022), the average duration of evaluation utterances is around 8 seconds, which amounts to $T = 400$ after applying a subsampling rate of 2 to the acoustic features. Under this setting, the self-attention module has lower computational complexity than that of the FFNs for a medium-sized dimension of $D = 256$, i.e., 1.45×10^8 v.s. 4.19×10^8. The computation of the SAN is even lower for short utterances. Therefore, a large proportion of the computational load of a Conformer falls onto the FFNs. This observation is contrary to NLP where the number of tokens T is much larger than the encoder dimension D. Moreover, the FFNs have $16D^2$ parameters, whereas the number of parameters of an SAN is only around $4D^2$. Given that the FFNs bear heavier computational load and have a larger number of parameters than the SAN, it is necessary to design a more efficient architecture for the Conformer block that can better balance the load between the FFN and the SAN.

On the other hand, the relative positional attention method (Dai et al., 2019) used in the Conformer does not explicitly enforce a convolution-like inductive bias, leading to insufficient locality modeling. Although Ramachandran et al. (2019) and Cordonnier et al. (2020) observed that some of the self-attention heads can demonstrate convolutional behaviors, a Conformer without ASR pre-training still cannot compete with the state-of-the art ResNet in SV (Zhang et al., 2022; Cai et al., 2023; Liu et al., 2022a). Therefore, an advanced self-attention mechanism is required to obtain enhanced locality in Conformers.

In this paper, we aim to address these two challenges so that the Conformer can be better adapted for SV tasks. Firstly, we modify the standard Conformer block by placing only one FFN between the SAN and the CNN to save computation and increase parameter efficiency. The resulting "SAN-FFN-CNN" structure not only retains the hard inductive bias of locality introduced by the CNN but also has fewer parameters compared with the original Conformer block. Secondly, we propose a multi-resolution attention strategy to increase the locality modeling capacity of the attention blocks. Inspired by the U-Net (Ronneberger et al., 2015) for image segmentation, which exploits a contracting path to successively extract multi-scale representations and a symmetric expanding path to increase the resolution, we downsample the query and key vectors to produce a low-resolution attention score map via inner products, upsample the low-resolution score map to the original resolution, and then fuse the resulting score map with the standard attention score map through linear weighting prior to the Softmax function. During upsampling, because the query-key relationship represented by each low-resolution attention score is propagated to its adjacent regions in the full-resolution score map, locality is attained around such a low-resolution attention score. By fusing the resolution-restored score map with the standard attention score map, the local dependency is explicitly introduced to the fused attention score map across the upsampled local region. As a result, the proposed attention mechanism has superior locality modeling to the vanilla self-attention. Moreover, the computation involved in the attention fusion is neglectable. We call the improved Conformer with attention fusion ConFusionformer in this paper.

The contributions of this paper are summarized as follows:

1. Compared with the conventional Conformer, an improved Conformer block with a smaller number of parameters is proposed by breaking the original Macaron-like architecture to save computation.
2. A novel attention mechanism is introduced to enhance the locality of the Conformer by the fusion between the restored full-resolution attention score map and the standard score map. Unlike the U-Net that upsamples the feature maps, it is the attention score map that is upsampled in the fused attention. To the best of our knowledge, this strategy is the first attempt to directly process the attention scores at multiple resolutions in SV.

This paper is organized as follows. In Section 2, we briefly overview the related works. Preliminary on Transformer and Conformer is given in Section 3. Section 4 presents the principle of the proposed attention fusion. The experimental settings and results are detailed in Section 5 and Section 6, respectively. Conclusions is given in Section 7.

2. Related Work

In this section, we briefly overview the improvement on local information modeling of Transformer encoders in terms of the architectures and the self-attention mechanisms.

2.1. Locality Enhancement of Transformer Architectures

Transformers have ubiquitous applications in NLP, CV, and speech processing for their high scalability. Transformers have inherent advantages in capturing the global dependencies in sequences but do not perform well in local information modeling, which, however, is important for speaker embedding. Researchers have incorporated CNNs into Transformers to address this limitation. In Wu et al. (2020), a self-attention layer and a CNN were placed in parallel to provide global and local information modeling capacity for edge NLP. In the Conformer (Gulati et al., 2020), a self-attention network and a CNN were cascaded to model the global and local context information of speech segments for ASR. Prabhu et al. (2024) further improved the Conformer block by introducing multiple convolutional kernels into the CNN module to capture diverse ranges of local information.

Inspired by the Conformer, Han et al. (2022) replaced the linear projections in the self-attention and those in the feed-forward network with convolutions and observed that both replacements improved the SV performance compared with the Transformer backbone. In Zhang et al. (2022), a Conformer with multi-scale feature aggregation was proposed by concatenating the representations of each Conformer block to diversify the frame-level information for speaker embedding. Cai et al. (2023) explored the use of ASR pre-trained Conformers on SV and found that the SV performance of a Conformer pre-trained on ASR tasks is significantly better than that of a Conformer trained from scratch using SV data only.

2.2. Locality Enhanced Self-attention

Despite the theoretical relationship between self-attention and convolution (Cordonnier et al., 2020), self-attention does not explicitly take locality modeling into account. Several studies suggest that incorporating locality into the attention mechanism can be advantageous. For example, Ramachandran et al. (2019) introduced a local attention layer in which each query vector attends to its nearby pixels only. Their results show that a network with such local attention layers can achieve better performance than CNNs on several CV tasks. In Beltagy et al. (2020) and Zaheer et al. (2020), sliding window attention was adopted in combination with global attention and obtained improved performance in long-document processing. A similar local attention mechanism was employed in Wang et al. (2022), where each head attends to a different range of context so that the speaker embeddings can aggregate information of different contextual sizes. Considering a Gaussian function a soft version of local attention, Han et al. (2022) further introduced a Gaussian self-attention strategy to enhance local information modeling and gained performance improvement in SV.

On the other hand, self-attention with relative positional encoding has been used to obtain convolutional inductive biases (Cordonnier et al., 2020). d'Ascoli et al. (2021) proposed gated positional self-attention to achieve soft convolution, which regulates the contribution of the content vectors and the positional encoding. Their results demonstrated strong local attention patterns in addition to global patterns. In fact, the attention method in the Conformer also employs relative positional encoding and therefore has locality modeling capability in theory. In this paper, we aim to add further locality modeling to the relative self-attention by attention fusion.

3. Preliminary

This section presents the background of the Transformer, the relative positional self-attention mechanism, and the Conformer.

3.1. Transformer

A Transformer encoder (Vaswani et al., 2017) is composed of a stack of Transformer blocks, and a standard Transformer block comprises a self-attention network (SAN) and a feed-forward network (FFN). Given an input feature sequence $\mathbf{X} \in \mathbb{R}^{T\times D}$, where T and D respectively denote the sequence length and feature dimension, an H-head Transformer block can be expressed as follows:

$$\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^{\mathrm{Q}}, \quad \mathbf{K}_h = \mathbf{X}\mathbf{W}_h^{\mathrm{K}}, \quad \mathbf{V}_h = \mathbf{X}\mathbf{W}_h^{\mathrm{V}}, \tag{1}$$

$$\mathbf{S}_h = \mathbf{Q}_h\mathbf{K}_h^{\top}, \tag{2}$$

$$\mathbf{A}_h = \mathrm{Softmax}\left(\mathbf{S}_h / \sqrt{D_{\mathrm{H}}}\right), \tag{3}$$

$$\mathrm{SAN}(\mathbf{X}) \triangleq \mathbf{X}_{\mathrm{SAN}} = \mathbf{X} + \sum_{h=1}^{H} \mathbf{A}_h\mathbf{V}_h\mathbf{W}_h^{\mathrm{O}}, \tag{4}$$

$$\mathrm{FFN}(\mathbf{X}_{\mathrm{SAN}}) = \mathbf{X}_{\mathrm{SAN}} + \mathrm{ReLU}\left(\mathbf{X}_{\mathrm{SAN}}\mathbf{W}_1\right)\mathbf{W}_2, \tag{5}$$

where $\mathbf{W}_h^{\mathrm{Q}}, \mathbf{W}_h^{\mathrm{K}}, \mathbf{W}_h^{\mathrm{V}} \in \mathbb{R}^{D\times D_{\mathrm{H}}}$ and $\mathbf{W}_h^{\mathrm{O}} \in \mathbb{R}^{D_{\mathrm{H}}\times D}$ are linear projection matrices for the query, key, value, and output feature maps corresponding to the h-th head of the SAN. $\mathbf{W}_1 \in \mathbb{R}^{D\times D_{\mathrm{F}}}$ and $\mathbf{W}_2 \in \mathbb{R}^{D_{\mathrm{F}}\times D}$ represent the projections in the FFN. $D_{\mathrm{H}} = D/H$ and D_{F} are the dimension of each head and the expanding width of the FFN, respectively. $\mathbf{S}_h \in \mathbb{R}^{T\times T}$ and $\mathbf{A}_h \in \mathbb{R}^{T\times T}$ denote the attention score map and attention (coefficient) map of Head h, respectively.

Note that self-attention inherently cannot capture the order information of a sequence. However, the order information is essential in sequence modeling. Positional encoding is adopted to incorporate such information into the Transformer.[1] For example, in Vaswani et al. (2017), sinusoidal positional encoding was added to the content tokens before being fed to the Transformer.

[1] $\mathbf{X}$ in Eq. (1) already includes such positional encoding.

3.2. Relative Positional Attention

Sinusoidal positional encoding was claimed to help the Transformer generalize to sequences longer than the training length due to the extrapolation ability of sinusoidal functions. However, because such positional encoding can only represent absolute positions, the performance may be suboptimal if the test length is longer than the training length. To overcome this limitation, Shaw et al. (2018) proposed a self-attention mechanism with *relative positional encoding* (RPE) to focus on the relative relationship between the positions, obtaining better length generalization. Dai et al. (2019) introduced a new RPE that incorporates a global content bias and a global position bias to accommodate long sequences during the inference stage. Also, a disentangled RPE was proposed in He et al. (2021) by using separate projection matrices for the content and position vectors to retain the attention related to content only. These RPE mechanisms can be summarized in a general form by integrating an attention bias map $\mathbf{B}_h = \left(b_{h,ij}\right) \in \mathbb{R}^{T \times T}$ into Eq. (2):

$$\mathbf{S}_h = \mathbf{Q}_h \mathbf{K}_h^\top + \mathbf{B}_h. \tag{6}$$

The attention bias $b_{h,ij}$ takes different forms in different RPE methods. These methods are outlined below.

Vanilla RPE (Shaw et al., 2018):

$$b_{h,ij} = \boldsymbol{q}_{h,i} \boldsymbol{p}_{\delta(i,j)}^\top, \tag{7}$$

where $\boldsymbol{q}_{h,i}$ is the i-th row vector[2] of the query feature map $\mathbf{Q}_h \in \mathbb{R}^{T \times D_\mathrm{H}}$, and $\boldsymbol{p}_{\delta(i,j)} \in \mathbb{R}^{1 \times D_\mathrm{H}}$ is a trainable positional embedding vector that only depends on the relative distance $\delta(i, j)$ between positions i and j. $\delta(i, j)$ is defined as

$$\delta(i,j) = \begin{cases} 0 & \text{if} \quad j - i \leqslant -R \\ 2R & \text{if} \quad j - i \geqslant R \\ j - i + R & \text{otherwise} \end{cases}, \tag{8}$$

where R denotes the maximum relative distance on one side. In this case, an RPE matrix $\mathbf{P} \in \mathbb{R}^{(2R+1) \times D_\mathrm{H}}$ will be learned, which is shared across the heads. The maximum relative distance is restricted to R because Shaw et al. (2018) observed that precise relative information did not help improve performance beyond a certain distance.

Transformer-XL's RPE (Dai et al., 2019):

$$b_{h,ij} = \boldsymbol{q}_{h,i} \left(\boldsymbol{p}_{h,\delta(i,j)} \tilde{\mathbf{W}}_h^\mathrm{K}\right)^\top + \boldsymbol{u}_h \boldsymbol{k}_{h,i}^\top + \boldsymbol{v}_h \left(\boldsymbol{p}_{h,\delta(i,j)} \tilde{\mathbf{W}}_h^\mathrm{K}\right)^\top, \tag{9}$$

where $\boldsymbol{u}_h, \boldsymbol{v}_h \in \mathbb{R}^{1 \times D_\mathrm{H}}$ are learnable positional bias vectors, $\tilde{\mathbf{W}}_h^\mathrm{K} \in \mathbb{R}^{D_\mathrm{H} \times D_\mathrm{H}}$ is a learnable matrix, and $\boldsymbol{k}_{h,i}$ is the i-th row vector of the key feature map $\mathbf{K}_h \in \mathbb{R}^{T \times D_\mathrm{H}}$. $\boldsymbol{p}_{h,\delta(i,j)} \in \mathbb{R}^{1 \times D_\mathrm{H}}$ is a trainable positional vector that is unique to the h-th head.

Disentangled RPE (He et al., 2021):

$$b_{h,ij} = \boldsymbol{q}_{h,i} \left(\boldsymbol{p}_{\delta(i,j)} \check{\mathbf{W}}_h^\mathrm{K}\right)^\top + \boldsymbol{p}_{h,\delta(i,j)} \check{\mathbf{W}}_h^\mathrm{Q} \boldsymbol{k}_{h,i}^\top, \tag{10}$$

where $\check{\mathbf{W}}_h^\mathrm{K} \in \mathbb{R}^{D_\mathrm{H} \times D_\mathrm{H}}$ and $\check{\mathbf{W}}_h^\mathrm{Q} \in \mathbb{R}^{D_\mathrm{H} \times D_\mathrm{H}}$ are learnable matrices. For this RPE, the attention score map $\mathbf{S}_h$ in Eq. (6) is divided by $\sqrt{3}$ before applying Eq. (3).

3.3. Conformer

The Conformer (Gulati et al., 2020) incorporates a CNN into each Transformer block to improve its locality modeling capability. As shown in Fig. 1(c), a Conformer block is comprised of an SAN and a CNN, sandwiched by two FFNs. Unlike the residual paths of the SAN and CNN, the output of each FFN's residual path is divided by 2 prior to the addition. Layer normalization (LN) is placed within each residual path before addition. The default self-attention follows the Transformer-XL's configuration with relative sinusoidal positional encoding.

4. Methodology

This section details the proposed ConFusionformer for speaker embedding with respect to its architecture, attention mechanism, and rationale behind locality enhancement.

4.1. Modified Conformer Architecture

For modern SV tasks, the average duration of test utterances is usually less than 40s after applying voice activity detection. As mentioned in Section 1, for a standard Conformer-based speaker embedding network, the major computation of each block is consumed by the FFNs. To balance the computational load between the FFN and SAN, we break the Macaron structure of the Conformer block by removing one FFN and placing the remaining FFN between the SAN and CNN, as shown in Fig. 1(c). The modified Conformer block has a smaller number of parameters and less computation compared with the standard block, which benefits using a deeper Conformer for superior SV performance. The CNN and FFN of the modified block follow the the same architecture as those in the standard Conformer block.

4.2. Multi-resolution Attention Fusion

A new attention mechanism through multi-resolution attention fusion is proposed to improve the locality modeling of the Conformer. As shown in Fig. 1(d), the attention fusion pipeline consists of three stages: (1) computing a low-resolution attention score map, (2) restoring the score map to full resolution, and (3) fusing the restored score map with the standard score map. The details of each stage are explained below.

Creation of low-resolution attention score maps: In this stage, a low-resolution query feature map $\mathbf{Q}_h^\mathrm{DS} \in \mathbb{R}^{T_\mathrm{DS} \times D_\mathrm{H}}$ and a key feature map $\mathbf{K}_h^\mathrm{DS} \in \mathbb{R}^{T_\mathrm{DS} \times D_\mathrm{H}}$ are first extracted by decimating the original full-resolution query and key maps ($\mathbf{Q}_h$ and $\mathbf{K}_h$) along the temporal axis, respectively. Given a downsampling rate of r, the number of frames (analogous to tokens in NLP) after decimation is $T_\mathrm{DS} = \lceil T/r \rceil$. This downsampling operation can be expressed as follows:

$$\mathbf{Q}_h^\mathrm{DS}[m, :] = \mathbf{Q}_h[r(m-1)+1, :], \tag{11}$$

$$\mathbf{K}_h^\mathrm{DS}[m, :] = \mathbf{K}_h[r(m-1)+1, :], \tag{12}$$

where $m \in [1, T_\mathrm{DS}]$ indexes the row vectors of a feature map in the downsampled feature space. Linear projections are then

[2] In this paper, symbols in bold italics denote row vectors.

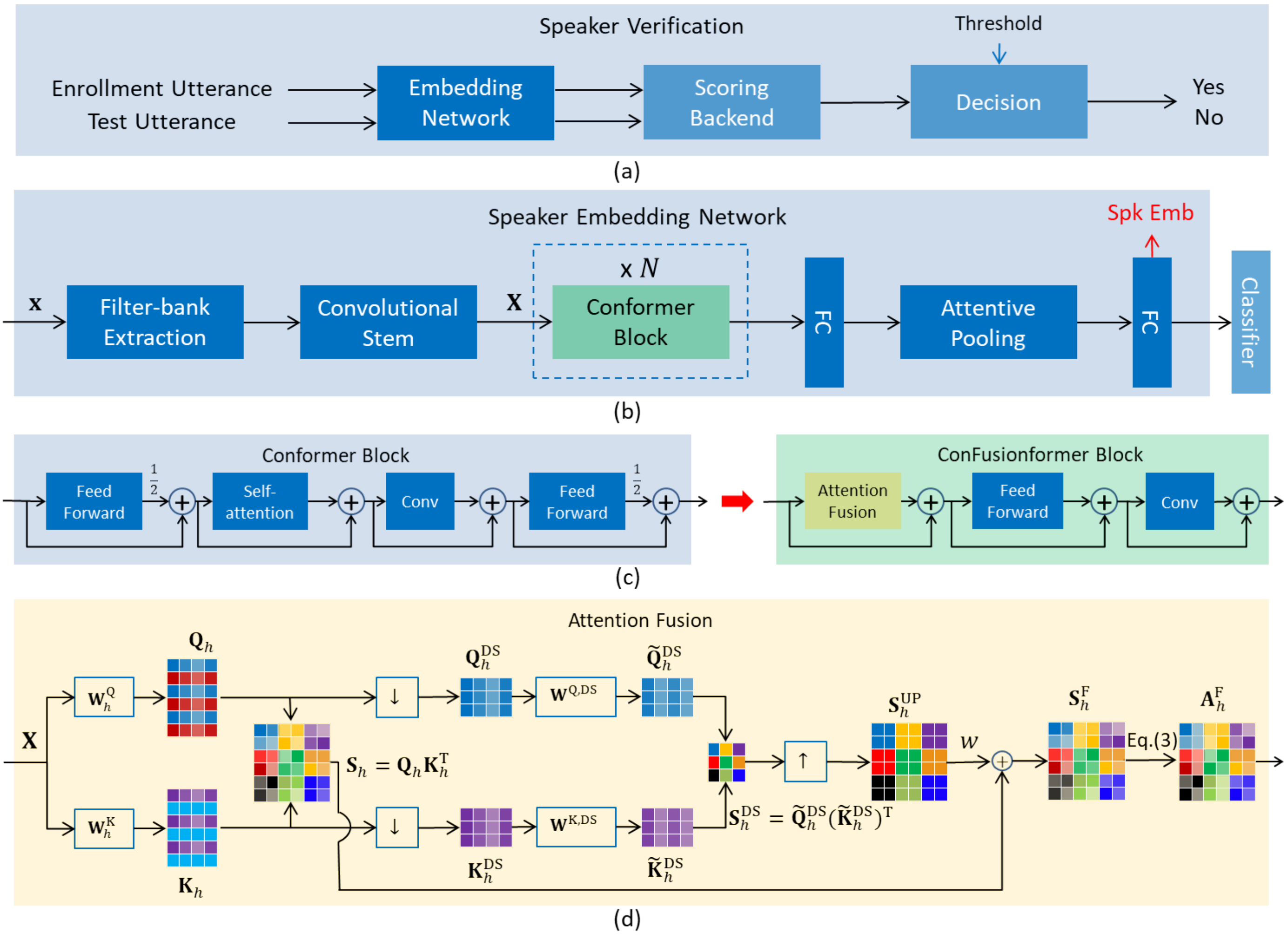
Speaker Verification
Threshold
Enrollment Utterance
Test Utterance
Embedding Network
Scoring Backend
Decision
Yes
No
(a)
Speaker Embedding Network
Spk Emb
x
Filter-bank Extraction
Convolutional Stem
X
x N
Conformer Block
FC
Attentive Pooling
FC
Classifier
(b)
Conformer Block
Feed Forward
1/2
Self-attention
Conv
Feed Forward
1/2
ConFusionformer Block
Attention Fusion
Feed Forward
Conv
(c)
Attention Fusion
X
$\mathbf{W}_h^{\mathrm{Q}}$
$\mathbf{Q}_h$
$\mathbf{W}_h^{\mathrm{K}}$
$\mathbf{K}_h$
$\mathbf{S}_h = \mathbf{Q}_h\mathbf{K}_h^{\mathrm{T}}$
↓
$\mathbf{Q}_h^{\mathrm{DS}}$
$\mathbf{W}^{\mathrm{Q,DS}}$
$\tilde{\mathbf{Q}}_h^{\mathrm{DS}}$
↓
$\mathbf{K}_h^{\mathrm{DS}}$
$\mathbf{W}^{\mathrm{K,DS}}$
$\tilde{\mathbf{K}}_h^{\mathrm{DS}}$
$\mathbf{S}_h^{\mathrm{DS}} = \tilde{\mathbf{Q}}_h^{\mathrm{DS}}(\tilde{\mathbf{K}}_h^{\mathrm{DS}})^{\mathrm{T}}$
↑
$\mathbf{S}_h^{\mathrm{UP}}$
w
$\mathbf{S}_h^{\mathrm{F}}$
Eq.(3)
$\mathbf{A}_h^{\mathrm{F}}$
(d)

Figure 1: (a) Overview of speaker verification (SV). (b) Schematic of a Conformer-based speaker embedding network. To train the network, a classification head is appended to the embedding layer. (c) Comparison of the architectures between the standard Conformer block and the modified version. (d) Pipeline of the proposed multi-resolution attention fusion. The ↓ and the ↑ denote the downsampling and upsampling operations, respectively. For illustration purpose, only one head (Head h) with a downsampling rate of $r = 2$ is used. In the downsampling operation, the query and key maps are decimated along the temporal (row) axis, respectively. During upsampling, each element of $\mathbf{S}_h^{\mathrm{DS}}$ is replicated to 4 elements (of the same color) in the $\mathbf{S}_h^{\mathrm{UP}}$ towards the right and the bottom directions.

applied to $\mathbf{Q}_h^{\mathrm{DS}}$ and $\mathbf{K}_h^{\mathrm{DS}}$, respectively:

$$\tilde{\mathbf{Q}}_h^{\mathrm{DS}} = \mathbf{Q}_h^{\mathrm{DS}}\mathbf{W}^{\mathrm{Q,DS}}, \tag{13}$$

$$\tilde{\mathbf{K}}_h^{\mathrm{DS}} = \mathbf{K}_h^{\mathrm{DS}}\mathbf{W}^{\mathrm{K,DS}}, \tag{14}$$

where $\mathbf{W}^{\mathrm{Q,DS}} \in \mathbb{R}^{D_{\mathrm{H}} \times D_{\mathrm{H}}}$ and $\mathbf{W}^{\mathrm{K,DS}} \in \mathbb{R}^{D_{\mathrm{H}} \times D_{\mathrm{H}}}$ are learnable matrices shared across the heads for the query and key, respectively. The low-resolution attention score map is obtained by

$$\mathbf{S}_h^{\mathrm{DS}} = \tilde{\mathbf{Q}}_h^{\mathrm{DS}}\left(\tilde{\mathbf{K}}_h^{\mathrm{DS}}\right)^{\top}. \tag{15}$$

Restoration of full-resolution attention score maps: This stage aims to recover the full-resolution attention score map from the low-resolution map by a simple upsampling operation called "equal replication" in this paper, inspired by Yao et al. (2024). Specifically, each element $s_{h,mn}^{\mathrm{DS}}$ ($m, n \in [1, T_{\mathrm{DS}}]$) of the low-resolution map $\mathbf{S}_h^{\mathrm{DS}}$ is replicated to a local region of size $r \times r$ in the restored score map, i.e., $\mathbf{S}_{h,ij}^{\mathrm{UP}}$ ($i \in [r(m-1)+1, rm]$ and $j \in [r(n-1)+1, rn]$), with a weight of $1/r$. An example of such upsampling with $r = 2$ is illustrated in Fig. 2, where each color of $\mathbf{S}_h^{\mathrm{DS}}$ is replicated to the same (light) color in the $\mathbf{S}_h^{\mathrm{UP}}$ with equal values.

Fusion of attention score maps: This stage fuses the restored attention score map with the standard attention score map by linear weighting:

$$\mathbf{S}_h^{\mathrm{F}} = \mathbf{S}_h + w\mathbf{S}_h^{\mathrm{UP}}, \tag{16}$$

where w is a learnable weight. Once $\mathbf{S}_h^{\mathrm{F}}$ is obtained, Eq. (3) is used to compute the attention coefficients.

Note that the above attention fusion only involves the attention computation between the query and key feature maps and is not related to the relative positional encoding. To incorporate positional information into the self-attention, we adopt the vanilla relative attention strategy (Shaw et al., 2018), where $(2R+1)$ positional encoding vectors $\boldsymbol{p}_{\delta(i,j)}$'s are learned, followed by a linear projection shared across the heads, i.e.,

$$b_{h,ij} = \boldsymbol{q}_{h,i}\left(\boldsymbol{p}_{\delta(i,j)}\mathbf{W}^{\mathrm{P}}\right)^{\top}, \tag{17}$$

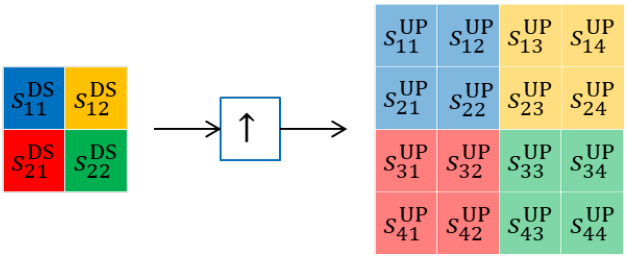
s_{11}^{DS}
s_{12}^{DS}
s_{21}^{DS}
s_{22}^{DS}
↑
s_{11}^{UP} s_{12}^{UP} s_{13}^{UP} s_{14}^{UP}
s_{21}^{UP} s_{22}^{UP} s_{23}^{UP} s_{24}^{UP}
s_{31}^{UP} s_{32}^{UP} s_{33}^{UP} s_{34}^{UP}
s_{41}^{UP} s_{42}^{UP} s_{43}^{UP} s_{44}^{UP}

Figure 2: Illustration of upsampling the low-resolution attention score map with $r = 2$. Each low-resolution attention score (e.g., s_{11}^{DS} in blue) is replicated to a local region of four elements (e.g., $\left\{s_{11}^{\mathrm{UP}}, s_{12}^{\mathrm{UP}}, s_{21}^{\mathrm{UP}}, s_{22}^{\mathrm{UP}}\right\}$ in light blue) in the full-resolution map weighted by 1/2, e.g., $s_{11}^{\mathrm{UP}} = s_{12}^{\mathrm{UP}} = s_{21}^{\mathrm{UP}} = s_{22}^{\mathrm{UP}} = s_{11}^{\mathrm{DS}}/2$.

where $\mathbf{W}^{\mathrm{P}} \in \mathbb{R}^{D_{\mathrm{H}} \times D_{\mathrm{H}}}$ is a learnable matrix. Because of the CNN in the architecture and the attention fusion operation, the new Conformer is called ConFusionformer in this paper.

4.3. Rationale on Locality Enhancement

The objective of multi-resolution attention fusion is to enhance local information modeling of the self-attention mechanism. In fact, such local information is introduced via the upsampling operation. From Eqs. (11–15), the low-resolution score map $\mathbf{S}_h^{\mathrm{DS}}$ contains the inner products of the decimated query and key vectors. As a result, a low-resolution score $s_{h,mn}^{\mathrm{DS}}$ in the downsampled score space corresponds to a single score $s_{h,r(m-1)+1,r(n-1)+1}$ in the original score space. Therefore, $\mathbf{S}_h^{\mathrm{DS}}$ can be seen as a decimated version of $\mathbf{S}_h$ in Eq. (2) with an additional linear projection. During upsampling, because each low-resolution score $s_{h,mn}^{\mathrm{DS}}$ is replicated to a full-resolution region $\mathbf{S}_{h,ij}^{\mathrm{UP}}$, where $i \in [r(m-1)+1, rm]$ and $j \in [r(n-1)+1, rn]$), the dependency between the m-th query and the n-th key vectors in the downsampled feature space is propagated to such local region. In other words, each restored region shares the same dependency originated from a single low-resolution score $s_{h,mn}^{\mathrm{DS}}$. This means that each restored region is localized to $s_{h,mn}^{\mathrm{DS}}$ in the downsampled score space and $s_{h,r(m-1)+1,r(n-1)+1}$ in the original score space.

To demonstrate how this kind of score localization reflects the locality at the attention output, we analyze the output of a single-head self-attention for simplicity:

$$\begin{aligned} \mathbf{O} = \mathbf{A}\mathbf{V} &= \mathrm{Softmax}\left(\mathbf{S}^{\mathrm{F}} / \sqrt{D_{\mathrm{H}}}\right)\mathbf{V} \\ &= \mathrm{Softmax}\left(\left(\mathbf{S} + w\mathbf{S}^{\mathrm{UP}}\right) / \sqrt{D_{\mathrm{H}}}\right)\mathbf{V}. \end{aligned} \tag{18}$$

Because the enhanced locality introduced by attention fusion depends on the term $\mathbf{S}^{\mathrm{UP}}$, we omit the Softmax function, dimensionality scale, and the original score $\mathbf{S}$, and we simplify the analysis of Eq. (18) to $\mathbf{Y} \triangleq \mathbf{S}^{\mathrm{UP}}\mathbf{V} \in \mathbb{R}^{T \times D}$. For any k-th and l-th row vectors of $\mathbf{Y}$, where $k, l \in [r(m-1)+1, rm]$ and $m \in [1, T_{\mathrm{DS}}]$, we have

$$\boldsymbol{y}_k = \sum_{t=1}^{T_{\mathrm{DS}}} \sum_{c=1}^{r} s_{k,tc}^{\mathrm{UP}} \mathbf{v}_{r(t-1)+c}, \tag{19}$$

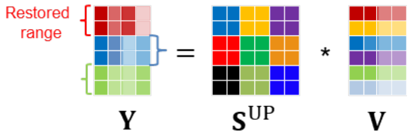

Figure 3: Illustration of the restored attention output $\mathbf{Y} = \mathbf{S}^{\mathrm{UP}}\mathbf{V}$. The row vectors in each restored range of $\mathbf{Y}$ (shown in the same color) are equal.

and

$$\boldsymbol{y}_l = \sum_{t=1}^{T_{\mathrm{DS}}} \sum_{c=1}^{r} s_{l,tc}^{\mathrm{UP}} \mathbf{v}_{r(t-1)+c}, \tag{20}$$

where $s_{k,tc}^{\mathrm{UP}}$ denotes the score at the k-th row and the tc-th column of $\mathbf{S}^{\mathrm{UP}}$. Similarly, $s_{l,tc}^{\mathrm{UP}}$ is the restored score at Row l and Column tc. $\mathbf{v}_{r(t-1)+c} \in R^{1 \times D}$ is the $(r(t-1)+c)$-th row vector of the value feature map $\mathbf{V}$. Because both row indices k and l locate in the same restored range $(k, l \in [r(m-1)+1, rm])$ corresponding to the m-th row of the downsampled score map $\mathbf{S}^{\mathrm{DS}}$, we have $s_{k,tc}^{\mathrm{UP}} = s_{l,tc}^{\mathrm{UP}}$. An example can be shown in Fig. 2, where the scores in the first row are the same as the second row, and the third row is the same as the fourth row. Therefore, $\boldsymbol{y}_k = \boldsymbol{y}_l$ is hold, which means that we enforce the same output within each group of the restored rows of $\mathbf{Y}$ (row vectors of the same color in Fig. 3). In this regard, $\mathbf{Y}$ can be seen as applying a 1-D convolutional kernel of equal weights with a kernel size of r and a stride of r (i.e., performing strided average pooling) to the value feature map along the temporal axis, weighted by $\mathbf{S}^{\mathrm{UP}}$. Because of the implicit convolution, the information of the output sequence $\boldsymbol{y}$'s is local to each restored range along the temporal direction. When $\mathbf{S}^{\mathrm{UP}}$ is incorporated in Eq. (18) with the original score $\mathbf{S}$, such locality is introduced to the attention mechanism, contributing to locality modeling enhancement.

5. Experimental Setup

In this section, the proposed ConFusionformer is compared with the ECAPA-TDNN (Desplanques et al., 2020), ResNet (He et al., 2016), Transformer (Vaswani et al., 2017), and Conformer (Gulati et al., 2020) as a frame-level encoder on VoxCeleb1 (Nagrani et al., 2020), CNCeleb1 (Li et al., 2022), SRE21 audio track (Sadjadi et al., 2022), and SRE24 audio track (NIST, 2024) test sets.

5.1. Datasets

VoxCeleb: The VoxCeleb corpus (Nagrani et al., 2020) consists of the VoxCeleb1 and VoxCeleb2 releases collected from the YouTube videos of more than 7,000 celebrities. These videos were recorded in the wild, covering a variety of real-world noise. The data are multilingual, but biased towards English. The VoxCeleb2 development subset (Vox2-dev) was used as the training set, covering 1,092,009 utterances from 5,994 speakers. Three VoxCeleb1 test sets, i.e., the original (Vox1-O), the extended (Vox1-E), and the hard (Vox1-H)), were used for evaluation. Vox1-O covers 37,611 enrollment-test pairs (trials) from 40 celebrities, whereas Vox1-E contains 579,818 pairs

Table 1: Statistics of the evaluation data. The duration of the SRE datasets is calculated after rVAD (Tan et al., 2020).

Eval. Set	Avg. Dur. (s)		#Speakers	#Trials
	Enroll	Test		
Vox1-O	8.2	8.2	40	37,611
Vox1-E			1,251	579,818
Vox1-H			1,190	550,894
CN1-eval	30.1	8.5	200	3,484,292
SRE21-dev	54.5	32.1	20	193,251
SRE21-eval	56.6	31.6	182	6,031,769
SRE24-dev	26.9	27.6	20	1,175,498

from 1,251 test speakers. Vox1-H consists of 550,894 pairs built within the same nationality and the same gender, leading to a more challenging set of 1,190 speakers. All of these data were sampled at 16kHz.

CNCeleb: CNCeleb (Li et al., 2022) is a 16kHz Mandarin corpus collected from the Chinese open media, covering around 3,000 speakers from 11 genres. It has two releases: CNCeleb1 and CNCeleb2. Both the CNCeleb1 and CNCeleb2 development sets (CN-dev) were used to train the speaker embedding networks, which contain around 560,000 utterances from 2,787 speakers. The CNCeleb1 evaluation (CN1-eval) subset was used for performance evaluation, covering about 3.5 million enrollment-test pairs from 200 speakers.

SRE21: The 2021 speaker recognition evaluation (SRE) advanced by the US National Institute of Standards and Technology (NIST) (Sadjadi et al., 2022) has three tracks: audio, visual, and audio-visual. We focus on the audio track only. The SRE21 data were curated from the multi-modal and multilingual WeCanTalk corpus, which consists of phone calls and video recordings spoken in Cantonese, English, and Mandarin. The audio track contains 8kHz conversational telephone speech (CTS) encoded as A-law and audio from video (AfV) in FLAC sampled at 16kHz. We evaluated the performance on two subsets: the SRE21 development set (SRE21-dev) and the SRE21 evaluation set (SRE21-eval). SRE21-dev has 193,251 trials, while SRE21-eval contains around six million test pairs.

SRE24: Similar to SRE21, SRE24 (NIST, 2024) also has three tracks and only the audio track was used in the experiments. The SRE24 data were compiled from the TELVID corpus spoken in Tunisian Arabic, French, and English. SRE24 also contains 8kHz CTS and 16kHz AfV. We used the SRE24 development subset for evaluation, which has around 1.1 million trials.

The statistics of these evaluation sets are summarized in Table 1. In addition to these data, the SRE CTS Superset (Sadjadi, 2021) and the SRE16 dataset (NIST, 2016) were used in the experiments. Specifically, the combination of the CTS Superset and the SRE16 development and evaluation subsets (SRE21-train) was used as the training set for SRE21, which amounts to 61,518 utterances from 7,088 speakers. For the SRE24 task, the SRE21-dev and SRE21-eval data were also added to the training set (SRE24-train), covering 632,471 utterances from 7,290 speakers. Note that because the majority of utterances in the CTS Superset were spoken in English, there is a language mismatch between the training set and the evaluation set of SRE21 and SRE24.

5.2. Acoustic Feature Extraction

Prior to acoustic feature extraction, rVAD (Tan et al., 2020) was adopted to remove the silent speech signals in the SRE CTS data. However, we did not perform any VAD on the VoxCeleb, CNCeleb, and SRE AfV utterances. 80-dimensional filter-bank features were extracted from each utterance with a 25ms window and a 10ms frame shift, followed by cepstral mean normalization. Reverberation, noise, music, and babble were added to the speech signals after rVAD for data augmentation before acoustic feature extraction. The additive noise sources come from the MUSAN corpus (Snyder et al., 2015). For reverberation, the original speech signals were convolved with the simulated room impulse responses generated from small- and medium-sized rooms.[3] Besides, speed perturbation (Ko et al., 2015) with speed factors of 0.9 and 1.1 was used as an additional augmentation.

5.3. Network Architecture

The ConFusionformer used the architecture in Fig. 1(b), where $N = 9$ and $N = 12$ blocks with an encoding dimension of 256 were stacked. The convolutional stem has similar configurations as that of the Zipformer (Yao et al., 2024), which comprises three 2-D convolutional layers with a kernel size of 3, followed by a ConvNeXt layer (Liu et al., 2022b). The convolutional layers have time × frequency strides of 1×2, 2×2, and 1×2, respectively, and the numbers of output channels are 8, 32, and 128, respectively. The ConvNeXt layer consists of a depth-wise convolutional layer with a kernel size of 7×7 and two point-wise convolutional layers of 512 and 128 output channels, respectively. The Gaussian error linear unit (GELU) (Hendrycks and Gimpel, 2016) was used as the activation function in the stem. The feature maps output by ConvNeXt were resized and projected with a linear layer of 256 nodes to match the input of the ConFusionformer.

For multi-resolution attention fusion, four heads were adopted and a downsampling rate of $r = 2$ was used for main results. We used the “skew” procedure in Huang et al. (2019) for relative positional attention (see Eq. (17)) to save memory. The number of learned positional embeddings R (see Eq. (8)) was set to 63. The feed-forward module and the convolutional module used the same architectures as those of the Conformer (Gulati et al., 2020). Stochastic depth (Huang et al., 2016; Touvron et al., 2021b) was employed to stabilize training and we adopted a drop-path rate of 0.15. On top of the ConFusionformer blocks, a 1×1 convolutional layer of 1,024 channels was used. We did not use feature aggregation (Zhang et al., 2022) across the blocks since worse performance was observed compared with the simply stacked architecture. Also, the mean

[3] https://www.openslr.org/28/rirs_noises.zip.

Table 2: Performance on VoxCeleb and CNCeleb. In Rows 4–9, the value following the hyphen denotes the number of Transformer/Conformer/ConFusionformer blocks. The number of FLOPs is calcuated by *fvcore* based on a 3.6s speech segment. For Rows 10 and 12, the number of FLOPs (with a superscript $*$) is estimated based on the models created from their GitHub releases.

Row	Model	#Para	#FLOPs	Vox1-O		Vox1-E		Vox1-H		CN1-eval	
		(M)	(G)	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
1	ECAPA-TDNN	14.7	2.57	0.77	0.075	0.96	0.104	1.79	0.177	7.61	0.411
2	ResNet-50	11.2	9.14	0.85	0.089	0.98	0.105	1.71	0.167	7.10	0.409
3	ResNet-101	16.0	17.7	0.63	0.063	0.79	**0.087**	1.40	**0.134**	**6.75**	0.410
4	Transformer-12	11.5	2.97	0.80	0.080	0.93	0.100	1.65	0.169	8.13	0.446
5	Transformer-16	14.7	3.67	0.85	0.091	0.96	0.103	1.68	0.163	8.41	0.472
6	Conformer-6	11.0	2.50	0.64	0.084	0.91	0.107	1.67	0.170	7.38	0.409
7	Conformer-8	14.1	3.04	0.67	0.063	0.82	0.096	1.60	0.163	7.68	0.417
8	ConFusionformer-9	10.9	2.45	0.68	0.064	0.93	0.104	1.66	0.166	7.36	0.402
9	ConFusionformer-12	13.9	2.97	**0.55**	**0.050**	**0.78**	**0.087**	**1.38**	0.143	6.76	**0.379**
10	CAM++ (Wang et al., 2023)	7.2	2.05^{*}	0.73	0.091	0.89	0.100	1.76	0.173	6.78	0.383
11	DF-ResNet233 (Liu et al., 2022a)	12.3	11.17	0.58	0.044	0.76	0.083	1.44	0.146	–	–
12	MFA-Conformer (Zhang et al., 2022)	20.5	3.76^{*}	0.64	0.081	–	–	–	–	–	–

and standard deviation vectors of the final frame-level representations were not used in the channel-wise attentive pooling layer (Desplanques et al., 2020) for better performance. The dimension of speaker embeddings is 192. The AM-Softmax loss function (Wang et al., 2018) was used as the training objective.

We used a channel size of 1,024 for the ECAPA-TDNN. The ResNet-50 and ResNet-101 followed the same settings as those in Chen et al. (2022). We investigated two Transformers with 12 and 16 blocks, respectively. The Transformer block adopted a similar structure as the ConFusionformer block except that the CNN module and attention fusion were not employed. The Conformer followed the same settings as the ConFusionformer except that the Macaron architecture was used. Six- and eight-block Conformers were studied.

5.4. Embedding Training

The stochastic gradient descent (SGD) optimizer was adopted during training. A linear learning rate warm-up from 0.01 to 0.1 was used during the first 5 epochs. The learning rate was then decayed to 0.001, following a cosine schedule. The mini-batch size was set to 256 and each mini-batch was composed of speech segments of 3.6 seconds randomly selected from the training set. The networks were trained for 40 epochs. For SRE21 and SRE24 tasks, the AfV data (in 16kHz) of the training set were downsampled to 8kHz before acoustic feature extraction.

5.5. Backend Processing

Cosine scoring was used in the VoxCeleb and CNCeleb experiments, whereas simplified Gaussian probabilistic linear discriminant analysis (SGPLDA) (Sizov et al., 2014) was employed for SRE21 and SRE24. The PLDA training set for each SRE task was selected from the corresponding embedding training data so that each speaker has 15 utterances. Prior to PLDA training, the speaker embeddings were projected onto a 150-dimensional space by LDA, followed by whitening and length normalization. The training set of the LDA model is the same as the PLDA training data. For SRE21-eval, domain adaptation based on correlation alignment (CORAL) (Alam et al., 2018) was used. The adaptation set adopted the SRE21-dev evaluation data. Adaptive score normalization (Matějka et al., 2017) was used for all evaluation tasks and the cohort was composed of the longest two utterances of each speaker in the corresponding PLDA training set.

6. Results

The performance was evaluated in terms of equal error rate (EER) and minimum detection cost function (minDCF) at $C_{\text{miss}} = 1$, $C_{\text{fa}} = 1$, and $P_{\text{target}} = 0.01$. The lower the EER and minDCF, the better the performance. Each result is an average of three independent runs.

6.1. Performance of ConFusionformer

For the VoxCeleb and CNCeleb tasks, both the training and evaluation sets use the 16kHz sampling rate. Besides, there is no mismatch between these two domains. Therefore, VoxCeleb and CNCeleb serve as ideal test benchmarks for speaker embedding methods. SRE21 and SRE24, on the other hand, are beneficial to further test the robustness of speaker embedding networks under language mismatch. Thus, the main results are reported in two groups.

6.1.1. Results on VoxCeleb and CNCeleb

The performance on VoxCeleb and CNCeleb is shown in Table 2. The numbers appended to the model names in Rows 4–9 represent the number of Transformer or Conformer blocks, and the number of FLOPs is computed from a segment of 3.6s by

Table 3: Performance on SRE21 and SRE24. In Rows 4–9, the value following the hyphen denotes the number of Transformer/Conformer/ConFusionformer blocks. The number of FLOPs is calcuated by *fvcore* based on a 3.6s speech segment.

Row	Model	#Para (M)	#FLOPs (G)	SRE21-dev		SRE21-eval		SRE24-dev	
				EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
1	ECAPA-TDNN	14.7	2.57	11.06	0.714	8.07	0.694	14.44	0.767
2	ResNet-50	11.2	9.14	10.71	0.715	8.13	0.705	14.91	0.697
3	ResNet-101	16.0	17.7	9.82	0.619	6.87	0.570	12.60	0.583
4	Transformer-12	11.5	2.97	15.78	0.784	8.04	0.650	13.89	0.719
5	Transformer-16	14.7	3.67	14.30	0.772	8.01	0.642	14.55	0.717
6	Conformer-6	11.0	2.50	11.09	0.712	7.96	0.667	14.04	0.685
7	Conformer-8	14.1	3.04	10.42	0.674	7.13	0.660	13.74	0.722
8	ConFusionformer-9	10.9	2.45	10.10	0.641	7.26	0.620	12.33	0.689
9	ConFusionformer-12	13.9	2.97	**9.12**	**0.606**	**6.72**	**0.533**	**10.87**	**0.565**

the *fvcore* library.[4] All of the Transformers, Conformers, and ConFusionformers used attention fusion in the self-attention mechanism.

From Table 2, we observe that the proposed ConFusionformer outperformed the other models under a similar model size with fewer computations on VoxCeleb. For example, the ConFusionformer with 12 blocks has a similar number of model parameters as that of the ECAPA-TDNN, ResNet-101, 16-block Transformer, and 8-block Conformer. However, ConFusionformer-12 obtained better performance than the others on the three VoxCeleb tasks. Although ResNet-101 achieved a lower minDCF than ConFusionformer-12, it requires much heavier computation. Similarly, ConFusionformer-9 surpassed ResNet-50, Transformer-12, and Conformer-6, confirming that the ConFusionformer with modified block structure and self-attention fusion is superior to the conventional Conformer in speaker modeling. We have the same conclusion on CNCeleb. Although ResNet-101 marginally outperformed ConFusionformer-12 in EER, it obtained worse minDCF than the latter with more FLOPs. On the other hand, by comparing Row 4 with Row 6 and Row 5 with Row 7, we observe that the Conformer achieved better performance, which verifies that incorporating a CNN into the Conformer block benefits speaker information extraction.

Rows 10–12 of Table 2 list existing works for further comparison. We see that ConFusionformer-12 remarkably outperforms the CAM++ (Wang et al., 2023) on VoxCeleb, and it obtains similar performance as the CAM++ on CNCeleb. The DF-ResNet233 (Liu et al., 2022a) obtains better performance than ConFusionformer-12 on Vox1-E but the situation reversed when Vox1-H was considered. In summary, under similar model size and similar computational cost, ConFusionformer has larger capacity in speaker modeling than the other models.

6.1.2. Results on SRE21 and SRE24

Table 3 shows the SRE performance. It is observed that ConFusionformer-12 outperformed the other models by a large margin although it has smaller computational cost under a similar number of parameters. Especially, ConFusionformer-12 clearly surpassed ResNet-101 in both EER and minDCF. This observation is in contrast with the situation of VoxCeleb and CNCeleb (see Table 2), where ResNet-101 can obtain better performance than ConFusionformer-12. By comparing Conformer-8 (Row 7) with Transformer-16 (Row 5) and Conformer-6 (Row 6) with Transformer-12 (Row 4), we see that the Conformer significantly outperformed the Transformer with a similar number of model parameters. This observation suggests that explicitly including a convolutional inductive bias in the Conformer for locality modeling is necessary for learning speaker information. The superior performance of the ConFusionformer again verifies the benefit of using modified Conformer blocks and the self-attention fusion mechanism.

6.2. Effect of Multi-resolution Attention Fusion

In this section, we investigated the effect of self-attention fusion on Vox1-O, Vox1-E, and Vox1-H. Three attention mechanisms with relative positional encoding (RPEs), namely, modified vanilla RPE (see Eq. (17)), Transformer-XL's RPE (see Eq. (9)), and disentangled RPE (see Eq. (10)), were exploited on Conformer-8 and ConFusionformer-12. The results were shown in Table 4.

For ConFusionformer-12, self-attention with modified vanilla RPEs generally outperformed the Transformer-XL's attention and disentangled attention under similar attention fusion settings. When attention fusion was enabled, clear performance gains were observed for all three self-attention methods with only a marginal increase in computational cost and model size, suggesting the benefit of attention fusion in capturing speaker information across the speech frames. On the other hand, the performance improvement due to attention fusion for Transformer-XL's attention and disentangled attention is not as significant as the vanilla relative positional self-attention. Recall that attention fusion only involves the attention score between the content query and content key (see Fig. 1(d)), and the positional attention bias (see Eq. (6)) is a calibration to the

[4] https://github.com/facebookresearch/fvcore

Table 4: Performance of Conformer-8 and ConFusionformer-12 on VoxCeleb with or without attention fusion. Three self-attention mechanisms were compared and the difference across these self-attentions lies in the computation of the bias term related to the relative positional encoding according to Eq. (17), Eq. (9), and Eq. (10). The number of FLOPs is calcuated by *fvcore* based on a 3.6s speech segment.

Row	Model	Self-attention	Attention	#Para	#FLOPs	Vox1-O		Vox1-E		Vox1-H	
		Mechanism	Fusion	(M)	(G)	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
1	Conformer-8	Modified vanilla attention (Eq. (17))	✗	14.7	1.50	0.73	**0.062**	0.86	0.099	1.67	0.174
2			✓	14.1	3.04	**0.67**	0.063	**0.82**	**0.096**	1.60	**0.163**
3		Transformer-XL's attention (Eq. (9))	✗	14.4	3.24	0.70	0.105	0.85	0.101	1.70	0.172
4			✓	14.4	3.32	0.68	0.092	0.86	0.101	1.68	0.172
5		Disentangled attention (Eq. (10))	✗	15.5	3.37	0.68	0.095	**0.82**	0.099	**1.58**	0.169
6			✓	15.5	3.39	0.68	0.092	0.86	0.101	1.70	0.172
7	ConFusionformer-12	Modified vanilla attention (Eq. (17))	✗	13.9	2.95	0.64	0.067	0.81	0.089	1.54	0.146
8			✓	13.9	2.97	**0.55**	**0.050**	**0.78**	**0.087**	**1.38**	**0.143**
9		Transformer-XL's attention (Eq. (9))	✗	14.5	3.28	0.76	0.090	0.90	0.098	1.57	0.156
10			✓	14.5	3.41	0.70	0.079	0.87	0.097	1.54	0.153
11		Disentangled attention (Eq. (10))	✗	16.0	3.48	0.62	0.062	0.84	0.095	1.53	0.154
12			✓	16.0	3.50	0.58	0.070	0.81	0.089	1.46	0.143

content attention score. However, excessively complex positional attention bias terms can over-calibrate the content attention scores. Therefore, the benefit of attention fusion would decrease for Transformer-XL's attention and disentangled attention. We have similar conclusions on Conformer-8.

6.3. Locality Enhancement Evaluation

Section 6.2 has shown that attention fusion is advantageous to speaker embedding, which benefits from locality modeling of the embedding network. In this section, we inspected the local information modeling ability of ConFusionformer-12 based on 500 speech segments of 3.6s randomly selected from the VoxCeleb1 test set. To measure the degree of locality, a concept of "nonlocality" proposed in d'Ascoli et al. (2021) was adopted as follows:

$$d_n^{\mathrm{NL}} := \frac{1}{T} \sum_{h,i,j} a_{h,ij,n} |i - j|, \tag{21}$$

$$d^{\mathrm{NL}} = \frac{1}{N} \sum_{n} d_n^{\mathrm{NL}}, \tag{22}$$

where $a_{h,ij,n} \in \mathbf{A}_h$ is an attention coefficient as defined in Eq. 3, n indexes the ConFusionformer block, and N is the number of blocks. The nonlocality d^{NL} denotes the average attention distance between the query frame and the key frame. The further the query frame attends to, the larger the nonlocality, and therefore the lower the locality.

The block-wise nonlocality d_n^{NL} and overall nonlocality d^{NL} of ConFusionformer-12 evolved during training are shown in Figs. 4(b) and 3(c), respectively. From Fig. 4(b), we see that the lower blocks (Block0–Block4) show a larger nonlocality than the upper blocks (Block6–Block11), indicating that the lower blocks have higher locality while the upper blocks mainly focus on global information modeling. This observation matches the experimental results of Luo et al. (2022), which demonstrated convolutional behaviors in the first few layers of a Transformer. During training, the locality of the first five blocks gradually increased and became stable after 30 epochs. This tendency is consistent with the evolution behavior of the fusion weight w (see Eq. (16)) as shown in Fig. 4(d), where w keeps a high value until convergence. Especially, Block1 and Block4 are closely related to the convergence of the w's in the first 5 blocks because the variation of locality of these two blocks is in perfect accordance with that of the fusion weights for Block0–Block4. Because w denotes the strength of attention fusion, this observation suggests that Block1 and Block4 largely determine the fusion process and that attention fusion indeed introduces local information modeling into the self-attention mechanism. As a further comparison, the block-wise nonlocality of ConFusionformer-12 without attention fusion is plotted in Fig. 4(a). We see that the ConFusionformer without attention fusion generally has higher nonlocality and thus lower locality than the standard ConFusionformer with respect to each block. Fig. 4(c) also confirms the priority of attention fusion in locality modeling.

6.4. Significance of the Downsampling Rate r

The downsampling rate r determines the decimation step size on the original queries and keys and also the size of the local regions in the restored attention score map. Thus, it could influence local information modeling during attention fusion. In this section, the impact of setting the downsampling rate r to 1, 2, 4, 8, 16, and 32 was inspected on ConFusionformer-12. The results were shown in Table 5.

When the downsampling rate was set to $r = 1$, the performance is similar to that without using attention fusion. This observation is expected because there is no frame interaction in this case, and thus no additional local information is introduced to the fusion process. Also, we see that the performance remained rather stable from $r = 4$ to $r = 16$. When r was increased to 32, the performance became worse than that without applying attention fusion. The best performance was achieved

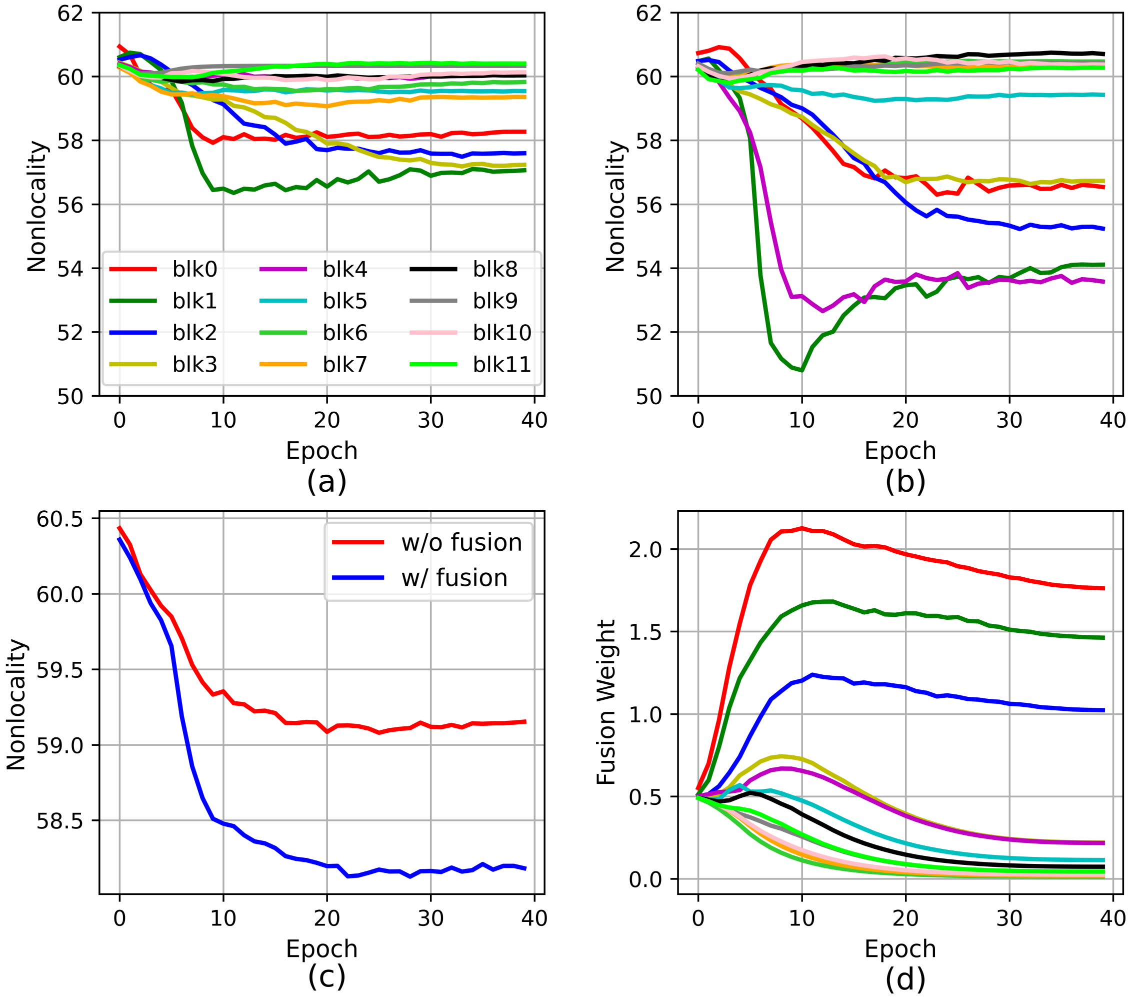

Figure 4: (a) Block-wise nonlocality d_n^{NL} (see Eq. (22)) of ConFusionformer-12 without attention fusion. (b) Block-wise nonlocality d_n^{NL} of ConFusionformer-12 (with attention fusion). (c) Overall nonlocality d^{NL} (see Eq. (21)) of ConFusionformer-12 with and without attention fusion. (d) Evolution of attention fusion weight w (see Eq. (16)) of ConFusionformer-12.

Table 5: Effect of downsampling rate r on attention fusion. The performance was evaluated on ConFusionformer-12. r ="–" in the first row means that attention fusion was not used.

r	Vox1-O		Vox1-E		Vox1-H	
	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
–	0.64	0.067	0.81	0.089	1.54	0.146
1	0.69	0.070	0.83	0.095	1.51	0.149
2	**0.55**	**0.050**	**0.78**	**0.087**	**1.38**	**0.143**
4	0.63	0.069	0.81	0.090	1.43	0.148
8	0.62	0.070	0.82	0.091	1.48	0.144
16	0.68	0.067	0.81	0.091	1.46	0.140
32	0.65	0.080	0.85	0.096	1.52	0.150

when r = 2. That is why we used this setting in previous experiments.

6.5. *Significance of Attention Fusion Strategy*

In attention fusion, a scaler w is learned to weight the restored attention score map before adding to the original score map, as shown in Eq. (16). Besides the learnable linear weight, we investigated other strategies of attention fusion, such as fixed linear weighting where $w = 1$ was kept (fixed w in Table 6) and block-shared linear weighting where w is a learnable parameter shared across all ConFusionformer blocks (shared w in Table 6). Multiple $\mathbf{S}_h^{\mathrm{UP}}$'s with w_r's were also experimented, i.e., $\mathbf{S}_h^{\mathrm{F}} = \mathbf{S}_h + \sum_r w_r \mathbf{S}_{h,r}^{\mathrm{UP}}$, where r corresponds to the sampling rate.

As shown in Table 6, when either the original score map $\mathbf{S}_h$ only or the restored score map $\mathbf{S}_h^{\mathrm{UP}}$ only was used in Eq. (16), the performance was worse than the proposed attention fusion. This result shows that using multi-resolution score restoration alone does not provide additional advantage. After all, some useful information is lost when producing the low-resolution attention score map from downsampled queries and keys. A fusion strategy with fixed weighting can improve performance and our fusion method achieved the best performance. However, when w is shared across the blocks, the performance dropped by a large margin, which suggests that different blocks

Table 6: Effect of attention fusion strategy on ConFusionformer-12. "Only $\mathbf{S}_h$" and "Only $\mathbf{S}_h^{\mathrm{UP}}$" in the first two rows means that only the original attention score map and only the restored map were used in Eq. (16), respectively. "Fixed w" keeps $w = 1.0$ in Eq. (16). "Shared w" denotes that the learnable w is shared across 12 blocks. "w_2 & w_4" represents that the score maps restored with $r = 2$ and $r = 4$ were fused with the original score map. "w_2 & w_4 & w_8" means the fusion between the restored maps using $r = 2$, $r = 4$, and $r = 8$ and the original attention score map.

Fusion	Vox1-O		Vox1-E		Vox1-H	
Strategy	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
Only $\mathbf{S}_h$	0.64	0.067	0.81	0.089	1.54	0.146
Only $\mathbf{S}_h^{\mathrm{UP}}$	0.66	0.062	0.84	0.091	1.50	0.149
Fixed w	0.60	0.067	0.83	0.092	1.44	0.147
Shared w	0.65	0.080	0.85	0.096	1.52	0.150
Ours	**0.55**	**0.050**	**0.78**	**0.087**	**1.38**	**0.143**
w_2 & w_4	0.64	0.062	0.83	0.089	1.48	0.146
w_2 & w_4 & w_8	0.69	0.068	0.83	0.092	1.50	0.151

require different fusion strengths. When multiple restored score maps were adopted in the fusion, the performance witnessed a slight degradation, indicating that there is local information interference among the multiple restored local regions.

7. Conclusions

In this paper, an improved Conformer—ConFusionformer—was proposed to enhance local information modeling. Using modified Conformer blocks, the ConFusionformer obtained better computational efficiency with fewer model parameters. Moreover, through multi-resolution attention fusion, local information were enhanced in the self-attention mechanism. Under similar computations, ConFusionformers achieved superior performance to the conventional Conformers and Transformers on VoxCeleb, CNCeleb, SRE21, and SRE24. These results indicate that the proposed ConFusionformer is beneficial in modeling the speaker dependencies in the speech signals.

8. Acknowledgment

This work was supported by the Research Grant Council of the Hong Kong SAR, Grant Nos. 15210122 and 15228223.

References

Alam, J., Bhattacharya, G., Kenny, P., 2018. Speaker verification in mismatched conditions with frustratingly easy domain adaptation, in: Proc. Odyssey: The Speaker and Language Recognition Workshop, pp. 176–180.

Bai, Z., Zhang, X., 2021. Speaker recognition based on deep learning: An overview. Neural Networks 140, 65–99.

Beltagy, I., Peters, M., Cohan, A., 2020. Longformer: The long-document transformer, in: arXiv preprint arXiv:2004.05150.

Cai, D., Wang, W., Li, M., Xia, R., Huang, C., 2023. Pretraining Conformer with ASR for speaker verification, in: Proc. International Conference on Acoustics, Speech, and Signal Processing, pp. 1–5.

Chen, Z., Liu, B., Han, B., Zhang, L., Qian, Y., 2022. The SJTU X-LANCE lab system for CNSRC 2022, in: arXiv preprint arXiv:2206.11699.

Cordonnier, J., Loukas, A., Jaggi, M., 2020. On the relationship between self-attention and convolutional layers, in: Proc. International Conference on Learning Representations.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., Salakhutdinov, R., 2019. Transformer-XL: Attentive language models beyond a fixed-length context, in: Proc. Annual Meeting of the Association for Computational Linguistics, pp. 2978–2988.

d'Ascoli, S., Touvron, H., Leavitt, M., Morcos, A., Biroli, G., Sagun, L., 2021. ConViT: Improving vision transformers with soft convolutional inductive biases, in: Proc. International Conference on Machine Learning, pp. 2286–2296.

Desplanques, B., Thienpondt, J., Demuynck, K., 2020. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification, in: Proc. Annual Conference of the International Speech Communication Association, pp. 3830–3834.

Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proc. North American Chapter of the Association for Computational Linguistics, pp. 4171–4186.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: Proc. International Conference on Learning Representations.

Gulati, A., Qin, J., Chiu, C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., Pang, R., 2020. Conformer: Convolution-augmented transformer for speech recognition, in: Proc. Annual Conference of the International Speech Communication Association, pp. 5036–5040.

Han, B., Chen, Z., Qian, Y., 2022. Local information modeling with self-attention for speaker verification, in: Proc. International Conference on Acoustics, Speech, and Signal Processing, pp. 6727–6731.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

He, P., Liu, X., Gao, J., Chen, W., 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention, in: Proc. International Conference on Learning Representations.

Hendrycks, D., Gimpel, K., 2016. Gaussian error linear units (GELUs), in: arXiv preprint arXiv:1606.08415.

Heo, H., Shin, U., Lee, R., Cheon, Y., Park, H., 2024. NeXt-TDNN: Modernizing multi-scale temporal convolution backbone for speaker verification, in: Proc. International Conference on Acoustics, Speech, and Signal Processing, pp. 11186–11190.

Huang, C., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A., Hoffman, M., Dinculescu, M., Eck, D., 2019. Music Transformer: Generating music with long-term structure, in: Proc. International Conference on Learning Representations.

Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K., 2016. Deep networks with stochastic depth, in: Proc. European Conference on Computer Vision, pp. 646–661.

Ko, T., Peddinti, V., Povey, D., Khudanpur, S., 2015. Audio augmentation for speech recognition, in: Proc. Annual Conference of the International Speech Communication Association, pp. 3586–3589.

Li, L., Liu, R., Kang, J., Fan, Y., Cui, H., Cai, Y., Vipperla, R., Zheng, F., Wang, D., 2022. CN-Celeb: multi-genre speaker recognition. Speech Communication 137, 77–91.

Liu, B., Chen, Z., Wang, S., Wang, H., Han, B., Qian, Y., 2022a. DF-ResNet: Boosting speaker verification performance with depth-first design, in: Proc. Annual Conference of the International Speech Communication Association, pp. 296–300.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin Transformer: Hierarchical vision transformer using shifted windows, in: Proc. International Conference on Computer Vision, pp. 9992–10002.

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., Xie, S., 2022b. A ConvNet for the 2020s, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 11976–11986.

Luo, S., Li, S., Zheng, S., Liu, T., Wang, L., He, D., 2022. Your Transformer may not be as powerful as you expect, in: Advances in Neural Information Processing Systems, pp. 4301–4315.

Matějka, P., Novotný, O., Plchot, O., Burget, L., Sáchez, M., Černocký, J., 2017. Analysis of score normalization in multilingual speaker recognition, in: Proc. Annual Conference of the International Speech Communication Association, pp. 1567–1571.

Nagrani, A., Chung, J.S., Xie, W., Zisserman, A., 2020. Voxceleb: Large-scale speaker verification in the wild. Computer Speech & Language 60.

NIST, 2016. NIST 2016 speaker recognition evaluation plan. `https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016`.

NIST, 2024. NIST 2024 speaker recognition evaluation plan. `https:`

//www.nist.gov/system/files/documents/2024/06/11/NIST_2024_Speaker_Recognition_Evaluation_Plan.pdf.

Okabe, K., Koshinaka, T., Shinoda, K., 2018. Attentive statistics pooling for deep speaker embedding, in: Proc. Annual Conference of the International Speech Communication Association, pp. 2252–2256.

Prabhu, D., Peng, Y., Jyothi, P., Watanabe, S., 2024. Multi-Convformer: Extending conformer with multiple convolution kernels, in: Proc. Annual Conference of the International Speech Communication Association, pp. 232–236.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training, in: OpenAI.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P., 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. The Journal of Machine Learning Research 21, 5485–5551.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J., 2019. Stand-alone self-attention in vision models, in: Advances in Neural Information Processing Systems, pp. 68–80.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation, in: Proc. Medical Image Computing and Computer-Assisted Intervention, pp. 234–241.

Sadjadi, S., 2021. NIST SRE CTS Superset: A large-scale dataset for telephony speaker recognition, in: arXiv preprint arXiv:2108.07118.

Sadjadi, S., Greenberg, C., Singer, E., Mason, L., Reynolds, D., 2022. NIST 2021 speaker recognition evaluation plan, in: arXiv preprint arXiv:2204.10242.

Shaw, P., Uszkoreit, J., Vaswani, A., 2018. Self-attention with relative position representations, in: Proc. North American Chapter of the Association for Computational Linguistics, pp. 464–468.

Sizov, A., Lee, K., Kinnunen, T., 2014. Unifying probabilistic linear discriminant analysis variants in biometric authentication, in: Proc. Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition, pp. 464–475.

Snyder, D., Chen, G., Povey, D., 2015. MUSAN: A music, speech, and noise corpus, in: arXiv preprint arXiv:1510.08484.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-vectors: Robust DNN embeddings for speaker recognition, in: Proc. International Conference on Acoustics, Speech, and Signal Processing, pp. 5329–5333.

Tan, Z., Sarkar, A., Dehak, N., 2020. rVAD: An unsupervised segment-based robust voice activity detection method. Computer Speech & Language 59.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H., 2021a. Training data-efficient image transformers & distillation through attention, in: Proc. International Conference on Machine Learning, pp. 10347–10357.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H., 2021b. Going deeper with image transformers, in: Proc. International Conference on Computer Vision, pp. 32–42.

Tu, Y., Lin, W., Mak, M., 2022. A survey on text-dependent and text-independent speaker verification. IEEE Access 10, 99038–99049.

Tu, Y., Mak, M., 2022. Aggregating frame-level information in the spectral domain with self-attention for speaker embedding. IEEE/ACM Transactions on Audio, Speech, and Language Processing 30, 944–957.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems, pp. 6000–6010.

Wang, F., Cheng, J., Liu, W., Liu, H., 2018. Additive margin softmax for face verification. IEEE Signal Processing Letters 25, 235–238.

Wang, H., Zheng, S., Chen, Y., Cheng, L., Chen, Q., 2023. CAM++: A fast and efficient network for speaker verification using context-aware masking, in: Proc. Annual Conference of the International Speech Communication Association, pp. 5301–5304.

Wang, R., Ao, J., Zhou, L., Liu, S., Wei, Z., Ko, T., Li, Q., Zhang, Y., 2022. Multi-view self-attention based transformer for speaker recognition, in: Proc. International Conference on Acoustics, Speech, and Signal Processing, pp. 6732–6736.

Wu, Z., Liu, Z., Lin, J., Lin, Y., Han, S., 2020. Lite transformer with long-short range attention, in: Proc. International Conference on Learning Representations.

Xie, W., Nagrani, A., Chung, J.S., Zisserman, A., 2019. Utterance-level aggregation for speaker recognition in the wild, in: Proc. International Conference on Acoustics, Speech, and Signal Processing, pp. 5791–5795.

Yao, Z., Guo, L., Yang, X., Kang, W., Kuang, F., Yang, Y., Jin, Z., Lin, L., Povey, D., 2024. Zipformer: A faster and better encoder for automatic speech recognition, in: Proc. International Conference on Learning Representations.

Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., Ahmed, A., 2020. Big Bird: Transformers for longer sequences, in: Advances in Neural Information Processing Systems, pp. 17283–17297.

Zhang, Y., Lv, Z., Wu, H., Zhang, S., Hu, P., Wu, Z., Lee, H., Meng, H., 2022. MFAConformer: Multi-scale feature aggregation conformer for automatic speaker verification, in: Proc. Annual Conference of the International Speech Communication Association, pp. 306–310.

Zhu, H., Lee, K., Li, H., 2022. Discriminative speaker embedding with serialized multi-layer multi-head attention. Speech Communication 144, 89–100.