Yao et al. EURASIP Journal on Audio, Speech, and Music Processing (2021) 2021:45
https://doi.org/10.1186/s13636-021-00234-3

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Anchor voiceprint recognition in live streaming via RawNet-SA and gated recurrent unit

Jiacheng Yao[1,2], Jing Zhang[1,2*], Jiafeng Li[1,2] and Li Zhuo[1,2]

Abstract

With the sharp booming of online live streaming platforms, some anchors seek profits and accumulate popularity by mixing inappropriate content into live programs. After being blacklisted, these anchors even forged their identities to change the platform to continue live, causing great harm to the network environment. Therefore, we propose an anchor voiceprint recognition in live streaming via RawNet-SA and gated recurrent unit (GRU) for anchor identification of live platform. First, the speech of the anchor is extracted from the live streaming by using voice activation detection (VAD) and speech separation. Then, the feature sequence of anchor voiceprint is generated from the speech waveform with the self-attention network RawNet-SA. Finally, the feature sequence of anchor voiceprint is aggregated by GRU to transform into a deep voiceprint feature vector for anchor recognition. Experiments are conducted on the VoxCeleb, CN-Celeb, and MUSAN dataset, and the competitive results demonstrate that our method can effectively recognize the anchor voiceprint in video streaming.

Keywords: Voiceprint recognition, Live streaming, Anchor, RawNet-SA, GRU

1 Introduction

With the substantial advances in computing technology, live video streaming is becoming increasingly popular. Due to the low employment threshold and acute competition of anchors, there are some issues in the online live streaming industry, such as unreasonable content ecology and uneven anchor quality. For seeking profits and accumulating popularity, some anchors mix inappropriate content into live programs. These offending anchors are usually found and banned after a period of time. However, they can still live by registering their sub-accounts as other anchors or occupying the rooms of other anchors after being blacklisted, which has caused great harm to the network environment. Therefore, it is indispensable to apply intelligent analysis techniques to identify anchors according to the specific characteristics of live streaming, so that regulators can prevent these banned anchors from continuing to live in various ways.

The anchor is the host and guide of live streaming, who performs the show to attract viewers. In general, the anchor's voice is often relatively stable and constant because he/she needs to create a fixed impression in the audience. If the anchor does not use a voice changer, the voiceprint of the anchor can be used to recognize the anchor identity, furthermore, to prevent the blocked anchor from entering the online live streaming platform again. Figure 1 shows the architecture of a live streaming system working with an anchor voiceprint recognition system, including three parts of camera, server, and client, of which the camera is used to capture live streaming, the server is used to encode and push video, and the client is used to decode and play video. The anchor voiceprint recognition system obtains a certain length of audio from the server through sampling, and stores it in the buffer as the system input. The sampling rules are determined by the live streaming platform, usually at the

* Correspondence: zhj@bjut.edu.cn
[1]Faculty of Information Technology, Beijing University of Technology, Beijing, China
[2]Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing, China

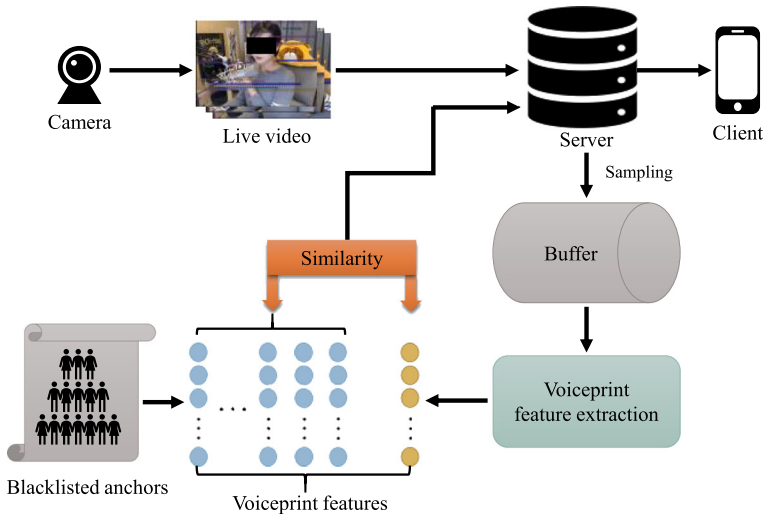

Fig. 1 The architecture of a live streaming system working with an anchor voiceprint recognition system

beginning or intervals of live streaming. The voiceprint features of audio are extracted and the similarity between the voiceprint features of input audio and those of blacklisted anchors is calculated and returned to the server. If the similarity is too high, the live streaming will be interrupted or manual review will be conducted.

Traditional speaker recognition methods usually use handcrafted features to recognize the speaker. For example, Reynolds et al. [1] proposed a speaker recognition method based on the Gaussian mixture model and universal background model (GMM-UBM). Firstly, acoustic features, such as Mel-scale frequency cepstral coefficients (MFCC), are projected onto high-dimensional space to generate high-dimensional mean hyper vector, and then to train a UBM. After that, taking UBM as the initial model, the target GMM of the speaker is constructed by adaptive training based on the maximum posterior probability with the target speaker data. Finally, the speaker is scored by calculating the likelihood value to make a recognition judgment. Although this method can reduce the speech demand for the target speaker as well as speed up the GMM training process, it is greatly affected by the channel type, training duration, male/female ratio, and other factors. Dehak et al. [2] proposed I-Vector (identity-vector) by using a specific space to replace the speaker space defined by the eigentone space matrix and the channel space defined by the channel space matrix. The new space can become a global difference space, including the differences between speakers and channels, thus reducing the impact of channel type and male/female ratio, but being sensitive to noise. Since live streaming is usually mixed with background music, game sound, and other noise, even though it is intractable to completely separate it by speech separation. Obviously, the traditional methods are not available to anchor voiceprint recognition.

Recently, deep learning has demonstrated powerful representation ability and anti-noise ability in speech processing. By training massive data, robust features can be obtained by using deep neural network (DNN). Consequently, a series of deep learning-based speaker recognition methods have been explored. For instance, Variani et al. [3] took the FBank features stacked into 1-D vectors as the input of DNN and extracted voiceprint features through continuous fully connected layers for speaker recognition. Compared with the traditional methods, voiceprint features extracted by DNN have stronger anti-noise ability, but parameters of fully connected layers are larger, hard to train, and easy to overfit. Snyder et al. [4] extracted voiceprint features through a time delay neural network (TDNN) like dilated convolution, to expand receptive field and share network parameters, effectively reducing the number of network parameters and training difficulty, and achieving 4.16% equal error rate (EER) on the SITW [5] dataset.

With the significant advantages of deep convolutional neural network (CNN) in image processing, some researchers refer to the idea of image processing, directly regarding the acoustic features as two-dimensional images, and further apply CNN to obtain voiceprint features. For example, Qian et al. [6] compared the effects of three deep models in automatic speaker verification (ASV) spoofing detection, including DNN, CNN, and bi-directional long short-term memory recurrent neural network (BLSTM-RNN), of which CNN performed best. Besides, Lavrentyeva et al. [7] proposed a CNN + bidirectional GRU (Bi-GRU) structure to extract voiceprint deep features for ASV spoof detection. Gomez-Alanis et al. [8] proposed a gated recurrent CNN (GRCNN) to extract voiceprint deep features by combining the ability of convolutional layer to extract discriminative feature

sequences with the capacity of recurrent neural network (RNN) for learning long-term dependencies. Furthermore, they proposed a light convolutional gated RNN (LC-GRNN) [9], which solves the high complexity by using a GRU-based RNN learning long-term dependency. Gomez-Alanis et al. [10] proposed an integration neural network, which is composed of LC-GRNN [9, 11], TDNN, and well-designed loss function to generate the deep features for ASV spoof detection, reaching the state-of-the-art (SOTA). Nagrani et al. [12] directly extracted the voiceprint features using CNN after representing the acoustic features as two-dimensional images, reaching an EER of 7.8% on the VoxCeleb1 [12] dataset. Hajavi et al. [13] improved the CNN structure to produce multi-scale voiceprint features, and the EER of VoxCeleb1 dataset was reduced to 4.26%. Jiang et al. [14] increased the depth of CNN and constrained the network through channel attention mechanism to enhance its representation ability. As a result, the EER on the VoxCeleb1 dataset is reduced to 2.91%. Although the above methods can reduce the input dimension of neural network, the hyperparameters of acoustic feature extraction methods may affect speaker recognition so that it is difficult to control their positive or negative. Similar to the idea of paraconsistent feature engineering [15], whether these handcrafted features are suitable as inputs to neural network depends not only on the features themselves but also on the network adopted to process them. Therefore, the hyperparameters are set empirically without a theoretical explanation.

Moreover, in the task of visual information processing, the first several layers of CNN are used to extract the local features at the low level, such as edge and texture features. In the subsequent convolutional layer, higher-level features are extracted layer by layer from these local features until semantic features are obtained. In speaker recognition tasks, we treat the input acoustic feature as a two-dimensional image to extract their features with CNN, similar to a local feature in physical meaning. Therefore, Jung et al. [16] further proposed a RawNet that can directly generate voiceprint features from the waveform of audio with 1-D residual CNN and GRU [17], achieving a 4.0% EER on the VoxCeleb1 dataset. This method does not need to extract any acoustic features, in which each 1-D convolutional layer can be regarded as a series of filters. Therefore, the final deep voiceprint feature can be extracted from a series of filters of the input audio. However, in view of the simple structure of RawNet, the deep voiceprint features extracted by RawNet will produce the performance of RawNet in speaker recognition inferior to that of methods using acoustic features as input. To improve the representation ability of the RawNet, Jung et al. [18] proposed RawNet2 by adding channel attention mechanism to the network to reduce EER to 2.48%, outperforming the method of taking acoustic features as input while eliminating the computational overhead of acoustic features. As shown in Fig. 2, the feature sequence can be segmented by channel dimension or frame/temporal dimension, yet the channel attention mechanism only regards the importance of different channels as well as ignores the relationship between frames. In fact, the relationship of frames is an important indication reflecting voiceprint information, yet channel attention alone cannot guide the network to pay attention to more important frames and ignore less important ones.

The transformer originally proposed by Vaswani et al. [19] has been applied to speech recognition. It can guide the network to learn the long-range dependence between feature sequence frames to enhance the representation ability of the model. Now, it has been extended to CNN. For instance, India et al. [20] proposed a voiceprint feature extraction method, which utilizes the multi-head self-attention module to replace the global pooling layer, aggregates the voiceprint feature sequence and transforms it into a deep voiceprint feature vector, dropping the EER by 0.9%. Safari et al. [21] also improved the performance of the speaker recognition model by replacing the global pooling layer with the self-attention pooling layer. This shows that proper use of self-attention structure can effectively improve the feature learning ability of neural network and contribute to voiceprint recognition.

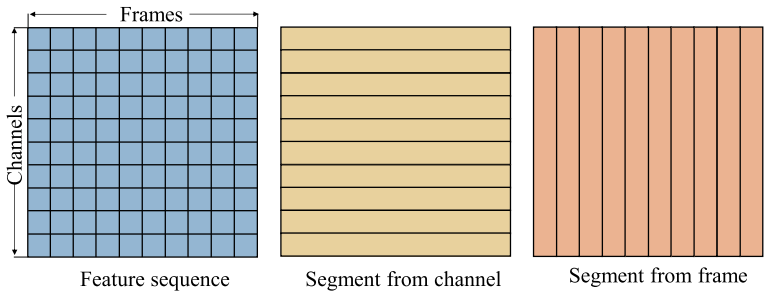

Fig. 2 Feature sequence segmented by channel dimension or frame dimension

When using the raw waveform as the input, the output of each layer of the model will retain the temporal context information that plays an important role in speaker recognition. Notably, the RNN can enhance the overall performance of the model owing to temporal information. As a representative one of RNN, GRU is a structure replacing long short-term memory (LSTM) [22] structure, which removes the forget gate and uses compliment of update gate vector to discard the information. Compared with LSTM, GRU can not only make use of the temporal relationship of feature sequences but also increase computational efficiency of long sequence modeling, effectively improving the representation ability of the model.

Through the analysis above, we choose the deep learning method for anchor voiceprint recognition, and take the waveform as the input of neural network. The self-attention mechanism is applied to the network to improve the feature learning ability of the model. Thereby, we propose an anchor voiceprint recognition method in live video streaming using RawNet-SA and GRU. The overall process of anchor voiceprint recognition system is as follows. First, the anchor's speech is extracted from the live streaming by using voice activation detection (VAD) and speech separation. Then, the feature sequence of the anchor voiceprint is generated from the waveform of the speech with the self-attention network RawNet-SA. RawNet-SA combines channel attention and self-attention to obtain the relationship between channel and frame in voiceprint feature sequence, to precisely distinguish the identity of anchors. And the input of RawNet-SA is waveform rather than acoustic features, so that makes the extracted deep features are not affected by the acoustic feature extraction process, and the network has better interpretability. Finally, the feature sequence of anchor voiceprint is aggregated by GRU and transformed into deep voiceprint feature vector for anchor recognition. The main contributions of this paper can be summarized as follows:

1. An effective RawNet-SA is designed to generate the feature sequence of anchor voiceprint from the speech waveform by adding channel/self-attention to obtain the relationship between channel and frame in the voiceprint feature sequence to precisely distinguish the identity of anchors.
2. The input of the proposed RawNet-SA is waveform rather than acoustic features, so that the extracted deep features are not affected by the acoustic feature extraction process, and the network has better interpretability.
3. We propose to recognize the anchor from the live streaming via the voiceprint deep features, which is a situational application.

The rest of this paper is organized as follows. Section 2 introduces our method in detail. Experimental results with ablation studies are presented and analyzed in Section 3. Conclusions are drawn in Section 4.

2 Method

The overall structure of our anchor voiceprint recognition method is shown in Fig. 3. First, the speech of the anchor is extracted from audio in live streaming by VAD and speech separation. Then, the feature sequence of

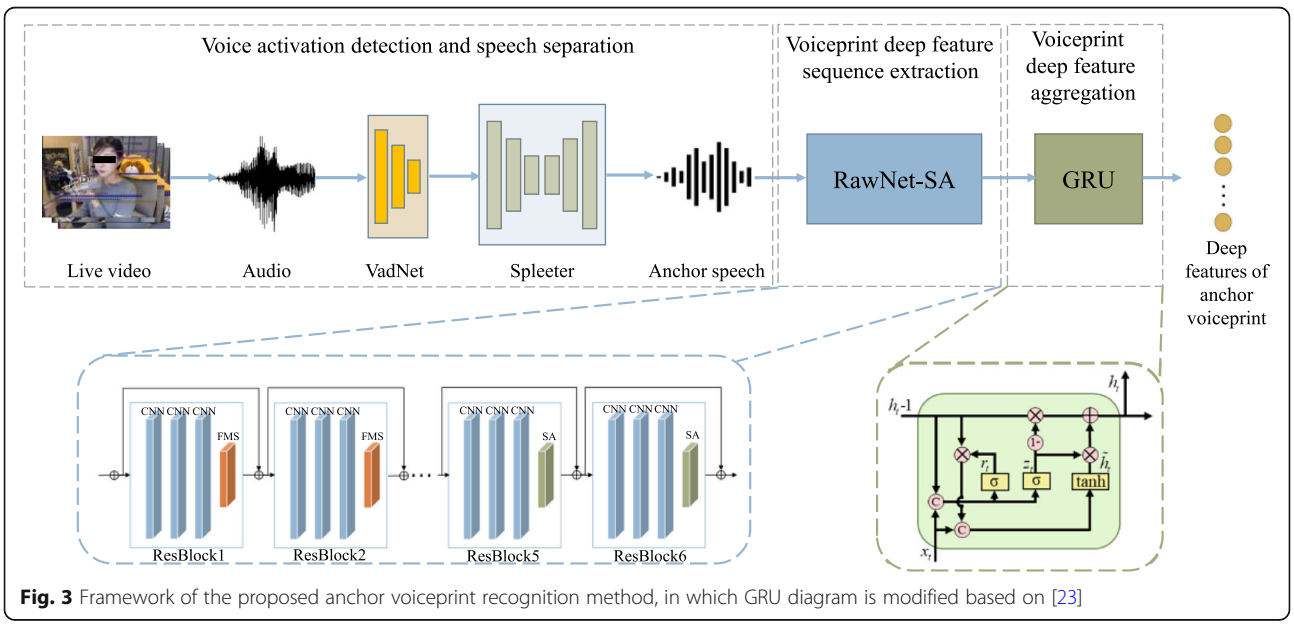

Fig. 3 Framework of the proposed anchor voiceprint recognition method, in which GRU diagram is modified based on [23]

anchor voiceprint is generated from the speech waveform by using the self-attention network RawNet-SA constructed based on RawNet2. Finally, the feature sequence of anchor voiceprint is aggregated by GRU to transform into a deep voiceprint feature vector for anchor recognition.

2.1 Voice activation detection and speech separation

Since the anchor in the live streaming will not be talking all the time, and there will be music, sound effects, outdoor noise, and other information to interfere with voiceprint recognition, it is necessary to remove the silent voice segments of the anchor through VAD before further processing, and then separate the speech. Traditional VAD methods are usually based on energy [24], pitch [25], zero crossing rate [26], and the combination of various features, the key problem of which is to judge whether there is speech in the audio segment.

Since the traditional methods cannot get expected results in complex environments, we adopt the lightweight network VadNet (Fig. 4) proposed by Wagner et al. [27] to realize VAD. Firstly, the feature sequence is generated by a three-layer CNN with the waveform of audio as the input. Then, the feature sequence is aggregated by a two-layer GRU and transformed into feature vector. Finally, the fully connected layer as a classifier is utilized to estimate whether the audio segment contains speech.

After removing the silent voice, we need to extract the anchor speech separately from the remaining audio segments containing background sound. Spleeter [28] is an open-source software developed by Deezer Research, which can separate various sounds including vocals in music, and is mainly applied to music information retrieval, music transcription, and singer identification, etc. We take the characteristics of Spleeter to separate the singer's voice from music to pick up the anchor's speech. Figure 5 describes the structure of U-Net [29] in Demucs (Fig. 5A) and the structure of encoder and decoder in U-Net (Fig. 5B). In Fig. 5A, based on Demucs [30], the soft mask of each source is estimated by a 12-layer U-Net, and separation is then done from the estimated source spectrograms with soft masking or multichannel wiener filtering. The network is composed of 6 encoders and 6 decoders, in which the feature sequence is modeled by two-layer Bi-LSTM between the encoder and decoder. The specific structure of encoder and decoder is shown in Fig. 5B, in which each encoder consists of two 1-D convolution layers, with ReLU and GLU as activation functions respectively. The difference between decoder and encoder is that the convolution layer activated by GLU comes before the convolution layer activated by ReLU, and the convolution layer activated by ReLU is no longer ordinary convolution, but transposal convolution. Since it only needs to separate the speech of the anchor, we use the 2-stem model provided by Spleeter, which only separates the speech from other sounds, rather than out producing four different types of sounds like the original Demucs model, to increase the separation speed.

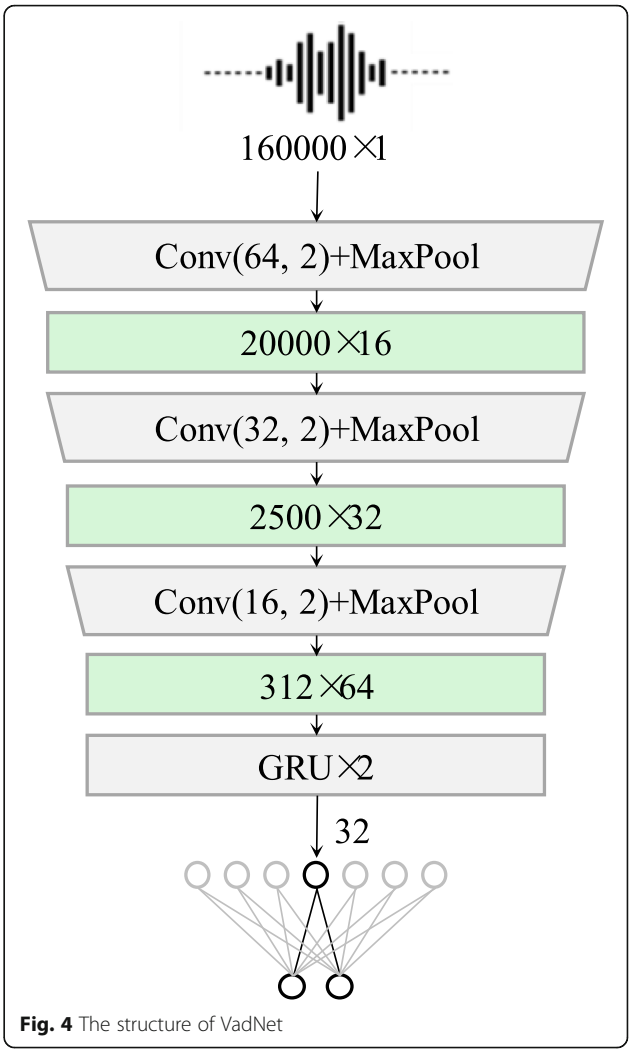

Fig. 4 The structure of VadNet

2.2 Voiceprint deep feature sequence extraction with RawNet-SA

During live streaming, there are normally tons of noise presented, for example, background music or noise and foreground non-human sound events. Even after preprocessing, the input audio will inevitably be mixed with some noise. More, the duration and speed of each speech may vary depending on the content of live streaming. However, the existing voiceprint feature extraction networks usually adopt acoustic features as input. The hyperparameters of the extracted acoustic features will influence the representation ability of voiceprint features. Thus, it is difficult to find the appropriate acoustic feature that can be adapted to the anchor voice

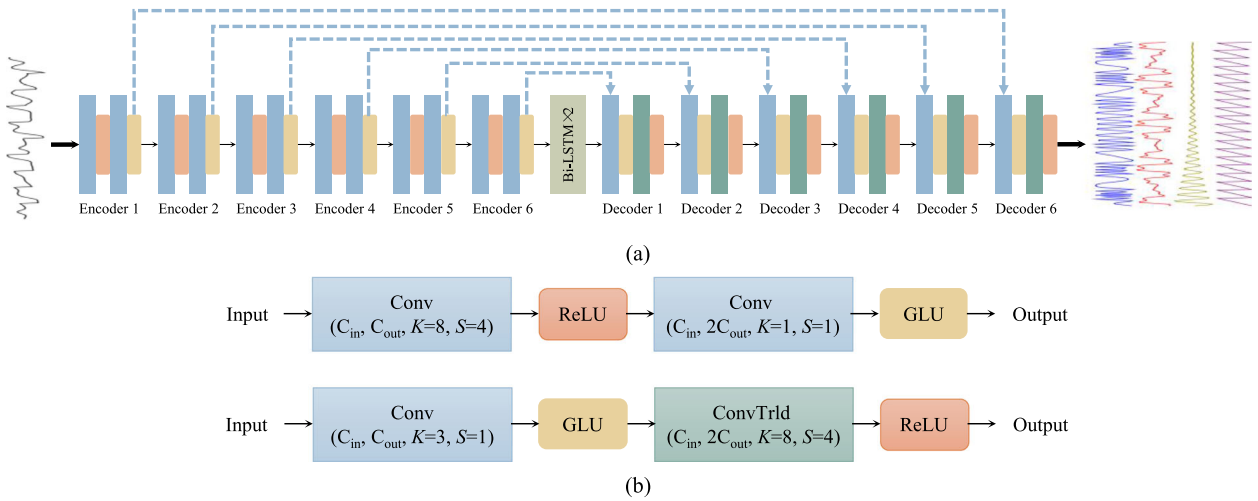

Fig. 5 The structure of U-Net in Demucs and the structure of Encoder and decoder in U-Net. **A** The structure of U-Net. **B** Detailed view of the layers Encoder_i (upper) and Decoder_i (below)

in all cases. Besides that, acoustic feature extraction requires additional computational overhead. By using audio waveform as input, RawNet2 does not need to extract acoustic features, while retaining the temporal relationship of audio, and achieves good performance on VoxCeleb dataset. We know that the self-attention mechanism can effectively strengthen the feature learning ability of neural network by regarding the importance of channels and frames of feature sequences. As a result, to avoid using acoustic features and further enhance the feature extraction ability of the network, we proposed a model combining RawNet2 with self-attention module (RawNet-SA) to generate anchor voiceprint features. The structure of RawNet-SA is shown in Table 1, in which numbers in Conv and Sinc indicate filter length, stride, and number of filters, and the number in Maxpool indicates filter length.

Since the computing cost of the self-attention layer will boost sharply with the increase of the dimension of input feature sequence, and the dimension of feature sequence is relatively high in the front part of RawNet-SA, the Sinc-conv layer and the first three Resblocks of RawNet-SA follow the structure of RawNet2 to accelerate the inference speed, while reducing the training difficulty. In addition, the channel attention layers of the last three Resblocks are replaced with self-attention layers to promote the feature representation ability of the model. To utilize the temporal information in the feature sequence, GRU is used to aggregate the feature sequence and transform it into a fixed-length feature vector.

The Sinc is a convolution layer with interpretable convolutional filters proposed in [31]. Different from the standard convolution layer, the kernel of Sinc is defined as the form of filter-bank composed of rectangular bandpass filters, and the learnable parameters only contain low and high cutoff frequencies. The Sinc can be computed with:

$$y[n] = x[n] * (g[n, f_1, f_2] \cdot w[n]) \tag{1}$$

Table 1 The structure of RawNet-SA

Layer	Input:59049	Output
Sinc	Sinc (251,1,128) MaxPool (3) BN LeakyReLU	(19,683,128)
Resblock×3	BN	(725,128)
	LeakyReLU	
	Conv (3,1,128)	
	BN	
	LeakyReLU	
	Conv (3,1,128)	
	MaxPool (3)	
	FMS	
Resblock × 3	BN	(26,256)
	LeakyReLU	
	Conv (3,1,256)	
	BN	
	LeakyReLU	
	Conv (3,1,256)	
	MaxPool (3)	
	SA	
Aggregation	GRU (1024)	(1024)
Embedding	FC (1024)	(1024)

$$g[n,f_1,f_2] = 2f_2\frac{\sin(2\pi f_2 n)}{2\pi f_2 n} - 2f_1\frac{\sin(2\pi f_1 n)}{2\pi f_1 n} \quad (2)$$

$$w[n] = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{L}\right) \quad (3)$$

where $x[n]$ is a chunk of the speech signal, $g[n, f_1, f_2]$ is the filter of length L, $y[n]$ is the filtered output, f_1 and f_2 represent low and high cutoff frequencies respectively, and $w[n]$ is Hamming window function.

The feature map scaled (FMS) layers in RawNet-SA follows the structure of the channel attention module in RawNet2. Different from the channel attention module commonly used in image processing, the vector generated by FMS is used as the weight and bias of the channels to improve the effect of attention constraint. Let $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_F]$ be the output feature sequence of Resblock, f be the number of channels in the feature sequence, and $\mathbf{c}_f \in \mathbb{R}^T$ (T is the length of feature sequence), then $\mathbf{C} \in \mathbb{R}^{T \times F}$. FMS can be computed with:

$$s_f = \mathrm{sigmoid}(\mathbf{c}_f \cdot \mathbf{w}) \quad (4)$$

$$\mathbf{c}_f^{'} = \mathbf{c}_f \cdot s_f + s_f \quad (5)$$

Because FMS only considers the relationship between channels and ignores the relationship between feature sequence frames, self-attention layer is utilized to enhance the representation ability of the model. In addition, since the computational complexity of self-attention layer increases sharply with the augment of the size of its input feature sequence, we only use/add self-attention layers in the last three Resblocks. The self-attention (SA) in Table 1 above represents the self-attention layer.

The structure of the original self-attention layer [19] for speech recognition is shown in Fig. 6A, where FC-KEY, FC-QUERY, and FC-VALUE represent fully connected layers respectively. The feature sequence is input to FC-KEY and FC-QUERY, and the outputs of FC-KEY and FC-QUERY are multiplied, and then normalized to obtain the weight matrix. The residual of the new feature sequence $\mathbf{A}$ is finally obtained by multiplying the weight matrix by the output of FC-VALUE as follows:

$$\mathbf{A} = \frac{\mathrm{softmax}\left(\mathbf{W}_q\mathbf{X}(\mathbf{W}_k\mathbf{X})^{\mathrm{T}}\right)}{\sqrt{d_k}} + \mathbf{W}_v\mathbf{X} \quad (6)$$

where $\mathbf{X} \in \mathbb{R}^{S \times d}$ is the matrix obtained by input word vectors concatenate; S denotes the number of word vectors; $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v \in \mathbb{R}^{d \times d}$ denote the parameter matrices of FC-QUERY, FC-KEY, and FC-VALUE in Fig. 6A respectively; and d_k represents the dimension of word vectors. To apply the self-attention layer to RawNet-SA, we let the time dimension T of voiceprint feature sequence as the sequence dimension S in word vector matrix as shown in Fig. 6B.

To accelerate the training speed, inspired by non-local neural network [32], the dimension of the feature sequence is compressed by FC-QUERY and FC-KEY, then restored by the fully connected layer FC-Extract before merging the residuals, and the batch normalize (BN) layer is applied to accelerate the training speed of the model. The residual $\mathbf{C}'$ is formally obtained as follows:

$$\mathbf{C}^{'} = \mathbf{W}_E\left(\mathrm{softmax}\left(\mathbf{W}_q\mathbf{C}(\mathbf{W}_k\mathbf{C})^{\mathrm{T}}\right)\right) + \mathbf{W}_v\mathbf{C} \quad (7)$$

where $\mathbf{W}_E \in \mathbb{R}^{c \times c}$ denotes the parameter matrices of FC-Extract and the BN calculation is omitted.

In a nutshell, the feature sequence $\mathbf{V} \in \mathbb{R}^{T \times c}$ is obtained by 3 Resblocks with channel attention layers and 3 Resblocks with self-attention layers with the waveform of speech as input, at which time $T = 26$ and $c = 256$.

2.3 Voiceprint deep feature aggregation by GRU

Most voiceprint feature extraction networks tend to apply the pooling-like methods or learnable dictionary encoding methods, such as global average pooling, global maximum pooling, NetVLAD [33], and GhostVLAD [34], to aggregate voiceprint feature sequences to transform them into deep voiceprint feature vectors. However, these methods do not consider the temporal relationship of feature sequences and lose a lot of information. Therefore, to effectively utilize the temporal relationship of feature sequences, GRU is applied to aggregate feature sequences in RawNet-SA. First, the reset gate vector $\mathbf{r}_t$ is generated to store the relevant information from the past time step in the new memory content. The Hadamard product of $\mathbf{r}_t$ and the previously hidden state $\mathbf{h}_{t-1}$ is then added to the input vector to determine what information is collected from the current memory content. After summing up, the non-linear activation function (tanh) is applied to obtain the $\tilde{\mathbf{h}}_t$. Secondly, the update gate will save the information of the current unit and pass it to the network. The update gate vector $\mathbf{z}_t$ will determine what information is collected from the current memory content and previous time-steps. Finally, the hidden state of the current unit is obtained by applying Hadamard product to $\mathbf{z}_t$ and $\mathbf{h}_{t-1}$, and summing it with the Hadamard product operation between (1- $\mathbf{z}_t$) and $\tilde{\mathbf{h}}_t$.

Let feature sequence $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T]$, $\mathbf{v}_t \in \mathbb{R}^c$ and c is the number of channels, then the aggregation of feature sequence is carried out according to the follows:

$$\mathbf{z}_t = \delta(\mathbf{W}_{xz}\mathbf{v}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z) \quad (8)$$

$$\mathbf{r}_t = \delta(\mathbf{W}_{xr}\mathbf{v}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r) \quad (9)$$

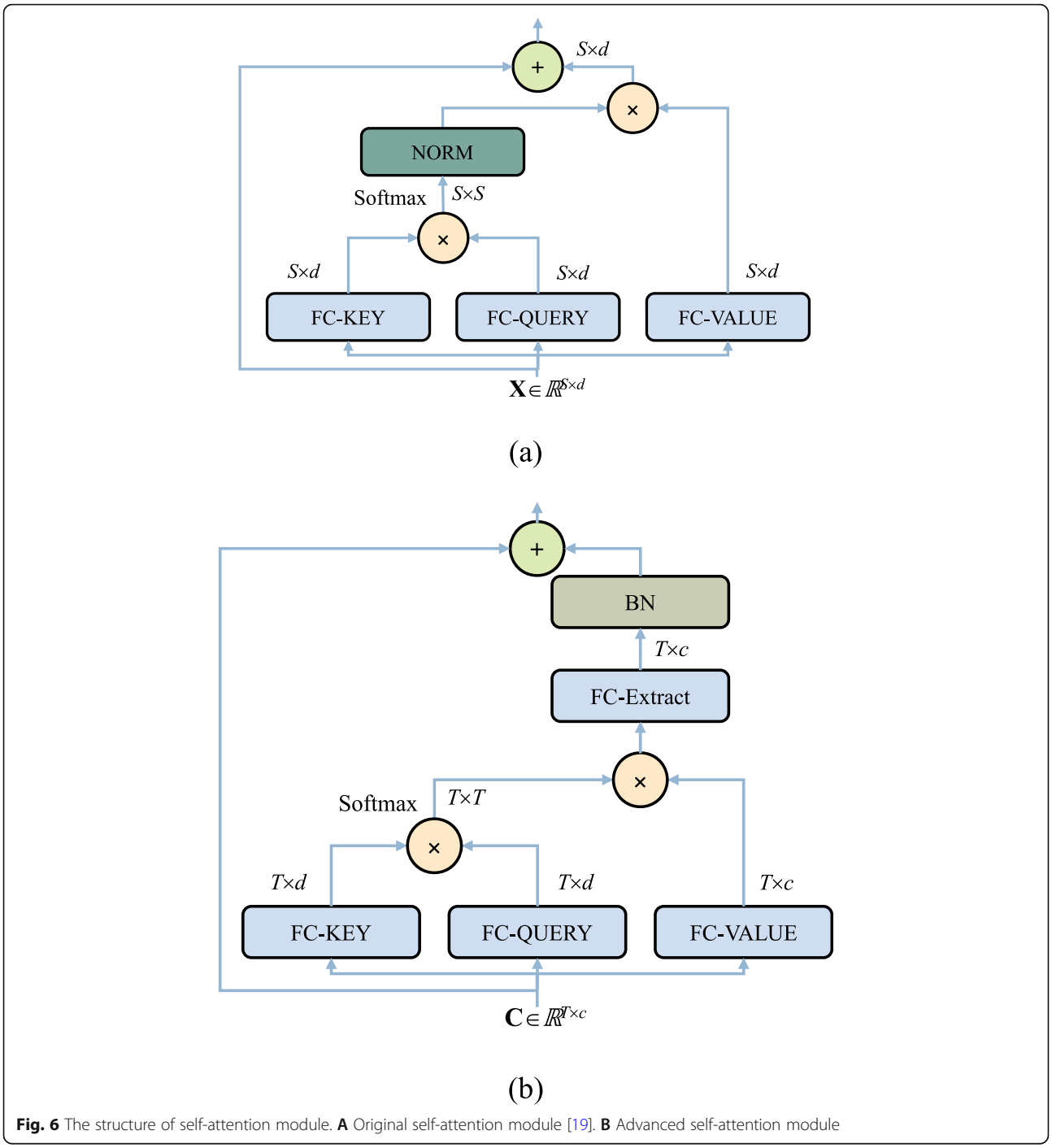

Fig. 6 The structure of self-attention module. **A** Original self-attention module [19]. **B** Advanced self-attention module

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{xh}\mathbf{v}_t + \mathbf{W}_{hh}(\mathbf{r}_t \bullet \mathbf{h}_{t-1}) + \mathbf{b}_h) \quad (10)$$

$$\mathbf{h}_t = \mathbf{z}_t \bullet \tilde{\mathbf{h}}_t + (1 - \mathbf{z}_t) \bullet \mathbf{h}_{t-1} \quad (11)$$

where $\mathbf{v}_t$ is the input, $\mathbf{z}_t$ is the update gate vectors, $\mathbf{r}_t$ is the reset gate vectors, $\mathbf{h}_t$ is the hidden states at time t, $\mathbf{W}$ represents the parameter matrices, $\mathbf{b}$ is the bias vector, and $\bullet$ denotes the element-wise product (Hadamard product). At last, to remove feature redundancy and accelerate the speed of anchor voiceprint recognition, the dimension of feature vector is controlled by the fully connected layer at the end of RawNet-SA.

2.4 Anchor voiceprint recognition with deep features

In this section, RawNet-SA is trained by the softmax loss function on the close dataset, and then the trained RawNet-SA generates the deep voiceprint feature of the anchor. As a result, the identity of the anchor depends

on the similarity of the anchor voiceprint features. The softmax loss function is calculated as:

$$L = -\sum_{i=1}^{m} \log \frac{\exp\left(\mathbf{W}_{y_i}^{\mathrm{T}}\mathbf{x}_i + \mathbf{b}_{y_i}\right)}{\sum_{j=1}^{n} \exp\left(\mathbf{W}_{y_j}^{\mathrm{T}}\mathbf{x}_i + \mathbf{b}_{y_j}\right)} \tag{12}$$

where m represents the size of Mini-Batch, n is the number of speakers in the dataset, $\mathbf{x}_i$ is the ith voiceprint feature vector in Mini-Batch, y_i is the true category of the ith feature vector in Mini-Batch, $\mathbf{W}_{yi}$ is the y_ith column of the parameter matrix of the full connection layer used for classification, and $\mathbf{b}_j$ is the jth row of the bias vector of the full connection layer. By converting $\mathbf{W}_{yi}{}^{\mathrm{T}}\mathbf{x}_i$ and $\mathbf{W}_{yi}{}^{\mathrm{T}}\mathbf{x}_j$ using the cosine function, we obtain:

$$L = -\sum_{i=1}^{m} \log \frac{\exp\left(\left\|\mathbf{W}_{y_i}^{\mathrm{T}}\right\|\|\mathbf{x}_i\|\cos\left(\theta_{i,i}\right) + \mathbf{b}_{y_i}\right)}{\sum_{j=1}^{n} \exp\left(\left\|\mathbf{W}_{y_j}^{\mathrm{T}}\right\|\|\mathbf{x}_i\|\cos\left(\theta_{i,j}\right) + \mathbf{b}_{y_i}\right)} \tag{13}$$

where $\theta_{i,j}$ is the included angle between the ith feature vector in the mini-batch and the jth column of the parameter matrix $\mathbf{W}$. Each column of the parameter matrix $\mathbf{W}$ can be regarded as the central vector of its corresponding category. Therefore, the process of using softmax loss function to train the network can be viewed as guiding the network to find the feature space, which makes the cosine similarity between the feature vector $\mathbf{x}$ and the vector of the corresponding column vector of the parameter matrix as high as possible. Meanwhile, the cosine similarity between the feature vector $\mathbf{x}$ and the vectors of other columns is low enough. Accordingly, in our application, cosine similarity is used as the similarity of voiceprint feature vector:

$$similarity = \frac{\mathbf{x}_1^{\mathrm{T}}\mathbf{x}_2}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|} \tag{14}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively represent the voiceprint feature vectors from different speech signals.

3 Experiments and discussion

In this section, we evaluate the performance of the proposed anchor voiceprint recognition in live streaming method by comparing with other SOTA speaker recognition methods.

We conduct a total of seven experiments as follows:

1. Experiment I: the overall performance comparison with SOTA methods.
2. Experiment II: the role of self-attention mechanism by ablation study.
3. Experiment III: the influence of self-attention module on inference speed.
4. Experiment IV: the effect of different channel squeeze ratios on voiceprint recognition in self-attention layer.
5. Experiment V: the influence of different feature aggregation methods on voiceprint recognition.
6. Experiment VI: the effect of VAD and speech separation on voiceprint recognition.
7. Experiment VII: the influence of different similarity measurement methods on voiceprint recognition.

3.1 Experiment setup

3.1.1 Dataset

We choose VoxCeleb2 [35] dataset, VoxCeleb1 dataset, CN-Celeb [36] dataset, and MUSAN [37] dataset to conduct the experiments. VoxCeleb1 and VoxCeleb2 datasets contain 1251 and 6112 speakers, respectively, without duplication. The speakers cover different ages, genders, and accents. Audio scenes include red carpet catwalks, outdoor venues, indoor video studios, etc. Sound acquisition equipment adopts professional and handheld terminals, and the background noise includes conversation, laughter, and different scenes. Using VoxCeleb2 dataset for training and VoxCeleb1 dataset for testing is a standard procedure for many speaker recognition methods. CN-Celeb dataset is an unconstrained large-scale Chinese speaker recognition dataset. The dataset contains 1000 speakers, each speaker contains at least five different scene recordings, with a total of about 130,000 sentences and a total duration of 274 h. It is collected from entertainment TV shows, singing, vlog, etc. It is very similar to the live streaming environment and contains all kinds of noise, such as background music, audience applauded, etc. MUSAN dataset consists of music from several genres, speech from twelve languages, and a wide assortment of technical and non-technical noises. It is often used to generate the corrupted version of other noiseless datasets. The general statistics of VoxCeleb1, VoxCeleb2, and CN-Celeb are given in Table 2.

All models in experiments were trained by VoxCeleb2 dataset. In experiment I-II, IV-V, and VII, we evaluate the methods on VoxCeleb-E and VoxCeleb-H [35] (two different test protocols of VoxCeleb1). We also use CN-Celeb dataset for test in experiment I-II, IV-V, and VII to see the effectiveness of the proposed method in scenes similar to the live streaming environment.

Table 2 Statistics of different datasets

Dataset	VoxCeleb1	VoxCeleb2	CN-Celeb
# of POIs	1251	6112	1000
# of utterances	153,516	1,128,246	130,000
# of hours	352	2442	274

POI person of interest

Specially, we illustrate the effectiveness of VAD and speech separation on CN-Celeb test set (CN-Celeb-T), VoxCeleb1 test set (Vox1T-O), and corrupted versions of Vox1T-O (Vox1T-N and Vox1T-M) generated using MUSAN.

3.1.2 Implementation details

Our experiment platform is a PC with 16 GB RAM, 2.40 GHz CPU, NVIDIA 2080Ti GPU, and Ubuntu 20.04 LTS operating system. Our framework is implemented by Pytorch, accelerated by CUDA10.1, and cuDNN7.6. RawNet-SA was trained on the VoxCeleb2 dataset. We modify the duration of the input waveforms to 59,049 samples (≈ 3.69 s) in training stage to facilitate mini-batch construction (If the length of the voice is less than 3.69 s, it can be copied to 3.69 s). In testing stage, we apply test time augmentation (TTA) [35] with a 20% overlap. Different parts of speech are intercepted to obtain multiple voiceprint feature vectors, and the average value is taken as the final voiceprint feature vector. During training stage, an Adagrad optimizer is used, and its learning rate starts from 0.01 and decays according to the following:

$$lr_t = \frac{lr_{t-1}}{1 + (d \times t)} \tag{15}$$

where lr_t is the learning rate at the tth iteration, t is the number of iteration steps, and d is the decay rate of the learning rate, which is set as 0.0001. And the batch size of network training is set as 50 and the total number of epochs is 35.

3.1.3 Evaluation indicators

We use the following evaluation indicators to verify the performance:

EER: a method widely used to measure the performance of voiceprint recognition. When the EER is lower, the overall recognition performance is better. Let the threshold value to judge whether the speaker is the same person be t, and the similarity of the two voiceprint feature vectors be s, when $s>t$, it is considered that the two feature vectors come from the speech of the same speaker; otherwise, they come from the speech of different speakers. After traversing the test set, different false rejection rates (FRR) and false acceptance rates (FAR) can be calculated for different thresholds:

$$FAR = \frac{FP}{TP + FP} \tag{16}$$

$$FRR = \frac{FN}{FN + PN} \tag{17}$$

where TP is the true positives, TN is the true negatives, FP denotes the false positives, and FN stands for the false negatives. When the threshold is adjusted to $FAR=FRR$, $ERR=FAR=FRR$.

Minimum detection cost function (minDCF): a method widely used to measure the performance of voiceprint recognition. The lower the minDCF, the better the overall recognition performance. DCF is calculated as follows:

$$DCF = C_{\mathrm{FR}} * FRR * P_{\mathrm{target}} + C_{\mathrm{FA}} * FAR * \left(1 - P_{\mathrm{target}}\right) \tag{18}$$

where C_{FR} and C_{FA} represent the penalty cost of FRR and FAR respectively, and P_{target} is a prior probability, which can be set according to different application environments. To improve the intuitive meaning of DCF, it is normalized by dividing it by the best cost that can be obtained without processing the input data:

$$DCF_{\mathrm{norm}} = \frac{DCF}{\min\left[C_{\mathrm{FR}} * P_{\mathrm{target}}, \quad C_{\mathrm{FA}} * (1 - P_{\mathrm{target}})\right]} \tag{19}$$

When C_{FR}, C_{FA}, and P_{target} are set, a set of values of FRR and FAR minimize DCF. Currently, $\mathrm{DCF}_{\mathrm{norm}}$ is minDCF. Here, we use two different sets of parameters to calculate minDCF:

1) DCF08: C_{FR} = 10, C_{FA} = 1, and P_{target} = 0.01 are set according to the setting of NIST SRE 2008.
2) DCF10: C_{FR} = 1, C_{FA} = 1, and P_{target} = 0.001 are set according to the setting of NIST SRE 2010.

3.2 Experiment I: comparison with state-of-the-art methods

In this experiment, we present the visualization results of the proposed method and SOTA methods on VoxCeleb1 and CN-Celeb dataset. The method proposed by Chung et al. [35] extracts the deep voiceprint feature sequence through ResNet50 and aggregates the feature sequence using time average pooling (TAP). ResNet50 initializes the network weight by using softmax pre-training and then compares the loss training with the offline hard negative mining strategy. The method proposed by Xie et al. [38] extracts the deep voiceprint feature sequence through Thin ResNet35 and uses GhostVLAD to aggregate the feature sequence. The network is trained by cross-entropy loss, and its performance outperforms the method in [35]. The method proposed by Nagrani et al. [39] uses the same network as the method in [38], but the network is pre-trained by cross-entropy loss, and then trained by relation loss. SpeakerNet [40] was proposed by Nvidia that uses statistics pooling (SP) to aggregate the feature sequence and is trained by AAM-Softmax loss [41]. DANet [42] generates the deep voiceprint feature sequence through the

VGG-like model described in [20] and introduces double multi-head attention to aggregate the feature sequence, in which the network is trained using cross-entropy loss.

As mentioned in Section 2.2, the self-attention module in Fig. 6A is an original version designed for speech recognition, which has more parameters and is difficult to train, while the self-attention module in Fig. 6B is an improved version that can enable the fast convergence of the network, and the number of parameters is adjustable. In this experiment, we tested both the original version RawNet-origin-SA* (a model using the original self-attention module of Fig. 6A) and the improved version RawNet-SA (a model using the self-attention module of Fig. 6B). Table 3 presents that our RawNet-origin-SA* reaches the lowest EER compared to the method using acoustic feature as network input and the baseline method RawNet2. RawNet-origin-SA* got 2.37% EER in VoxCeleb-E, a decrease of 0.32% compared with SpeakerNet and 0.20% compared with RawNet2. In VoxCeleb-H, EER of 4.54% can be obtained, which is decreased by 0.07% and 0.35% for DANet and RawNet2, respectively. This is because the self-attention module can make the network focus on the relationship between feature frames, while RawNet2 only uses channel attention to pay attention to the channel dimension of the feature map. RawNet-SA attained 4.52% EER on VoxCeleb-H, 0.37% less than RawNet2, and 2.54% EER on VoxCeleb-E, 0.03% less than RawNet2. RawNet-SA is not as effective as RawNet-origin-SA* because the network is not initialized with the parameters of trained RawNet2, so the actual training iterations of RawNet-SA are less than RawNet-origin-SA*. Although Thin ResNet34 [38] and SpeakerNet perform better than RawNet-SA in CN-Celeb dataset, considering the performance of all datasets, the overall performance of RawNet-SA and RawNet-origin-SA is optimal. It should be noted that the number of training iterations of SpeakerNet is about six times that of RawNet-SA, and AAM-Softmax loss is used in training.

To evaluate and compare their performance at all operating points, we provide the detection error tradeoff (DET) curves (Fig. 7) of the baseline method RawNet2 and the proposed RawNet-origin-SA*, RawNet-SA, as shown in Fig. 7A, B, and C respectively. It can be seen that RawNet-origin-SA* performs best on all operating points of the simple test set VoxCeleb-E. In the complex test set VoxCeleb-H, RawNet-SA approximates RawNet-origin-SA* and exceeds RawNet-origin-SA* in CN-Celeb dataset.

We also include Fig. 8 to exhibit what speech will be considered as the voice of the same person and what speech will be considered as the voice of different people by the RawNet-SA. We randomly selected 4 pairs of speech audios, which were true-positive (TP) pair, true-negative (TN) pair, false-positive (FP) pair, and false-negative (FP) pair respectively. Speech audios in true positive pair come from the same speaker and the similarity between these deep voiceprint features of audios is high enough. The spectrograms in the TP part of Fig. 8 are very similar. Speech audios in true negative pair come from different speakers so that the similarity between these deep voiceprint features of audios is low enough. It can be seen that there are significant differences in the spectrograms in the TN part of Fig. 8. However, the spectrograms in the FP part and FN part are similar so it is hard to judge if they are from the same speaker by the deep voiceprint feature extracted with RawNet-SA.

3.3 Experiment II: ablation study of self-attention mechanism

To demonstrate the role of self-attention mechanism, we conducted ablation study on VoxCeleb1 dataset and CN-Celeb dataset using EER and minDCF, as shown in Table 4, in which RawNet2 was used as the baseline model to study the method. We can see that our RawNet-origin-SA* and RawNet-SA exceed the baseline method. More details of the experiment are described below.

RawNet w/out SA* is based on RawNet2 that removes the channel attention layer of the last 3 Resblocks, which achieve 2.44% EER in VoxCeleb-E, 0.07% higher than

Table 3 Results of comparison to state-of-the-art method on VoxCeleb-E and VoxCeleb-H evaluation protocols

Method	Input	Backbone	Loss	CN-Celeb	VoxCeleb-E	VoxCeleb-H
Chung et al. [35]	S	ResNet50	TAP	/	4.42%	7.33%
Thin ResNet3 4[38]	S	Thin ResNet34	GhostVLAD	20.04%	3.13%	5.06%
Nagrani et al. [39]	S	Thin ResNet34	GhostVLAD	/	2.95%	4.93%
SpeakerNet [40]	S	SpeakerNet-M	SP	**19.33%**	2.69%	4.80%
DANet [42]	S	DANet	Double SA	24.11%	3.18%	4.61%
RawNet2	Raw	RawNet2	GRU	24.27%	2.57%	4.89%
RawNet-origin-SA*	Raw	RawNet-origin-SA	GRU	23.49%	**2.37%**	4.54%
RawNet -SA	Raw	RawNet-SA	GRU	22.24%	2.54%	**4.52%**

"*" denotes that the network is initialized with the trained RawNet2 parameters. Original-SA denotes the self-attention layer in Fig. 6A

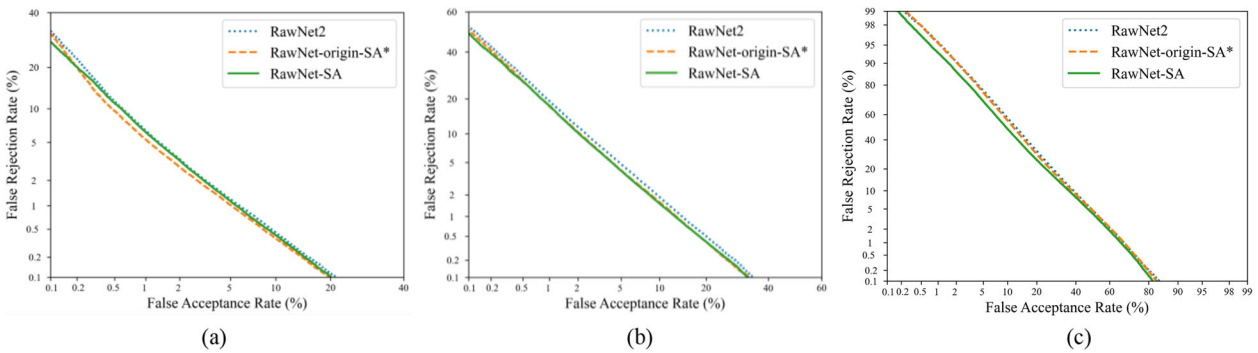

Fig. 7 DET curves of models on different datasets. **A** VoxCeleb-E. **B** VoxCeleb-H. **C** CN-Celeb

RawNet-origin-SA*, and 4.69% EER in VoxCeleb-H, 0.15% higher than RawNet-origin-SA*. The training of RawNet w/out SA* follows the same protocol as RawNet-origin-SA*, but its performance is still inferior to RawNet-origin-SA*, which demonstrates that the model performance is not promoted by the redundancy of channel attention layer in RawNet2, but the addition of self-attention layers effectively enhances the representation ability of the model.

RawNet-MHSA is a model that replaces the self-attention module in RawNet-SA with a multi-head version, and the number of SA heads is set to 4. This means the input feature sequence will be split into 4 chunks in the channel dimension, and then processed separately by the self-attention module, and finally concatenated to the output of the multi-head self-attention module. RawNet-MHSA achieves 2.75% EER in VoxCeleb-E, 0.18% higher than that of RawNet2, and 4.91% EER in VoxCeleb-H, 0.02% higher than that of RawNet2. In CN-Celeb, 22.16% of EER is obtained, 0.08% lower than that of RawNet-SA. Although RawNet-MHSA performed well in the CN-Celeb dataset, its performance on other datasets was even worse than the baseline method.

RawNet-all-SA is based on RawNet2 that all six FMSs are replaced with self-attention modules, which can achieve 3.69% EER in VoxCeleb-E, 1.15% higher than that of RawNet-SA, and 6.61% EER in VoxCeleb-H, 2.09% higher than that of RawNet-SA. As mentioned in Section 2.2, the computing cost and parameter size of RawNet-all-SA are much larger than RawNet-SA, the training time of RawNet-all-SA will take about twice as that of RawNet-SA, and it will not converge like RawNet-SA.

RawNet-origin-SA* is a model using the self-attention module described in Fig. 6A instead of the self-attention module in Fig. 6B. Due to the difficulty in training the original self-attention module, the network is initialized with the trained RawNet2 network parameters, and the

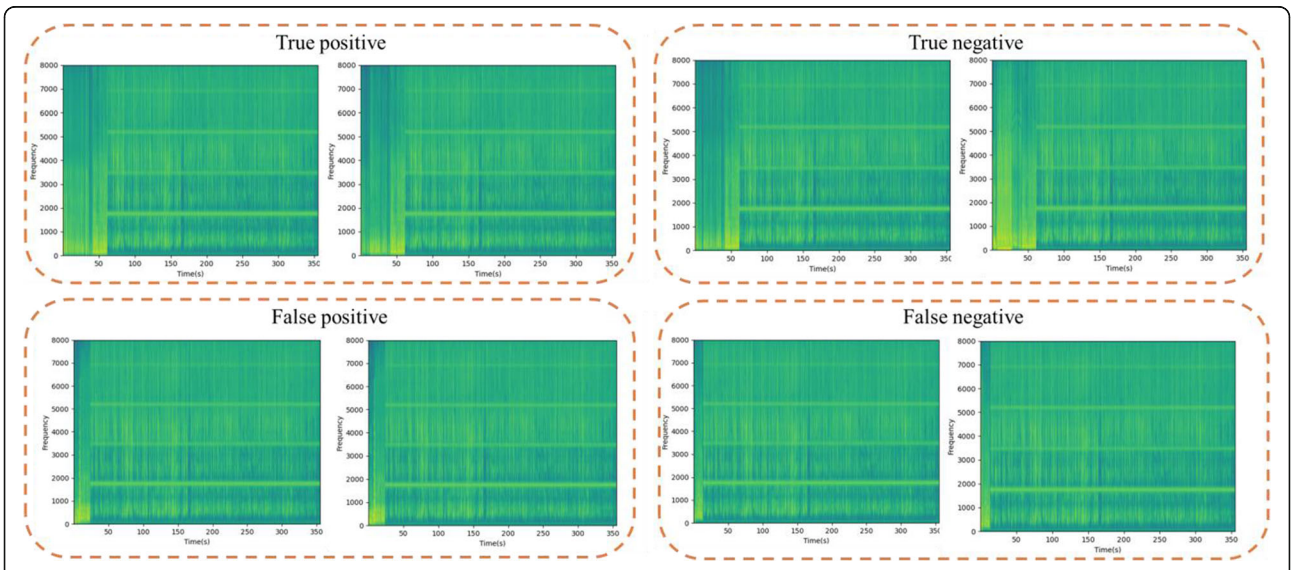

Fig. 8 The visualization results of anchor voiceprint recognition

Table 4 The role of self-attention mechanisms on the recognition performance

Models	VoxCeleb-E			VoxCeleb-H			CN-Celeb		
	EER	DCF08	DCF10	EER	DCF08	DCF10	EER	DCF08	DCF10
RawNet2	2.57%	0.14	0.52	4.89%	0.24	0.64	24.27%	0.78	0.97
RawNet2*	2.43%	**0.13**	0.50	4.60%	0.23	0.64	24.23%	0.78	0.96
RawNet2 w/out SA*	2.44%	0.14	0.48	4.69%	0.23	0.64	23.55%	0.77	0.94
RawNet-MHSA	2.75%	0.15	0.53	4.91%	0.24	0.65	**22.16%**	**0.75**	**0.93**
RawNet-all-SA	3.69%	0.20	0.61	6.61%	0.32	0.73	22.51%	0.77	0.96
RawNet-origin-SA*	**2.37%**	**0.13**	0.50	4.54%	**0.22**	**0.63**	23.49%	0.78	0.94
RawNet-SA	2.54%	0.14	**0.47**	**4.52%**	**0.22**	0.65	22.24%	0.76	0.94

"*" denotes that the network is initialized with the trained RawNet2 parameters

VoxCeleb2 dataset is used to further fine-tune the network. From Table 4, RawNet-origin-SA* achieves 2.37% EER in the VoxCeleb-E, 0.20% lower than RawNet2. In VoxCeleb-H, 4.54% of EER is obtained, 0.35% lower than RawNet2. To ensure the fairness of the comparison, we also trained the original RawNet2 in the same way. The experimental results named RawNet2* achieves 2.43% EER in VoxCeleb-E, 0.06% higher than RawNet-origin-SA*, and 4.60% EER in VoxCeleb-H, 0.06% higher than RawNet-origin-SA*. This indicates that the improvement of RawNet-origin-SA* is not caused by more training iterations.

RawNet-SA improves the structure of self-attention layers so that the network can quickly converge without using the parameters of trained RawNet2 for initialization. Finally, RawNet-SA achieved an EER of 2.54% in VoxCeleb-E, 0.03% lower than RawNet2, and 4.52% in VoxCeleb-H, 0.37% lower than RawNet2. RawNet-SA also achieved 22.24% EER in CN-Celeb dataset, even lower than other networks initialized by parameters such as the trained RawNet2* or RawNet-origin-SA*. This shows that the improved self-attention layer can further lift the robustness of voiceprint features and make the network suitable for different data distributions.

3.4 Experiment III: influence of self-attention module on inference speed

To prove that the inference speed of our proposed network structure is not significantly below that of the RawNet2, we test the time cost of different network structures as shown in Fig. 9.

Since the specific content of the input data does not affect the inference time of the model, we use the randomly generated sequence instead of the real-world audio as the network input and set the length of sequences to 3.69 s to control the length of the input. In the experiment, we randomly generated 1000 speech samples, each 100 into a group, for 10 consecutive tests, and finally taking the shortest time as the result. Figure 9 shows that RawNet-SA only consumes about 15.60 ms, 0.43 ms more than the original RawNet2 for each speech sample, and costs 1.02 ms less than RawNet-origin-SA* for each speech sample, which indicates that the addition of self-attention layers has little influence on the inference speed of

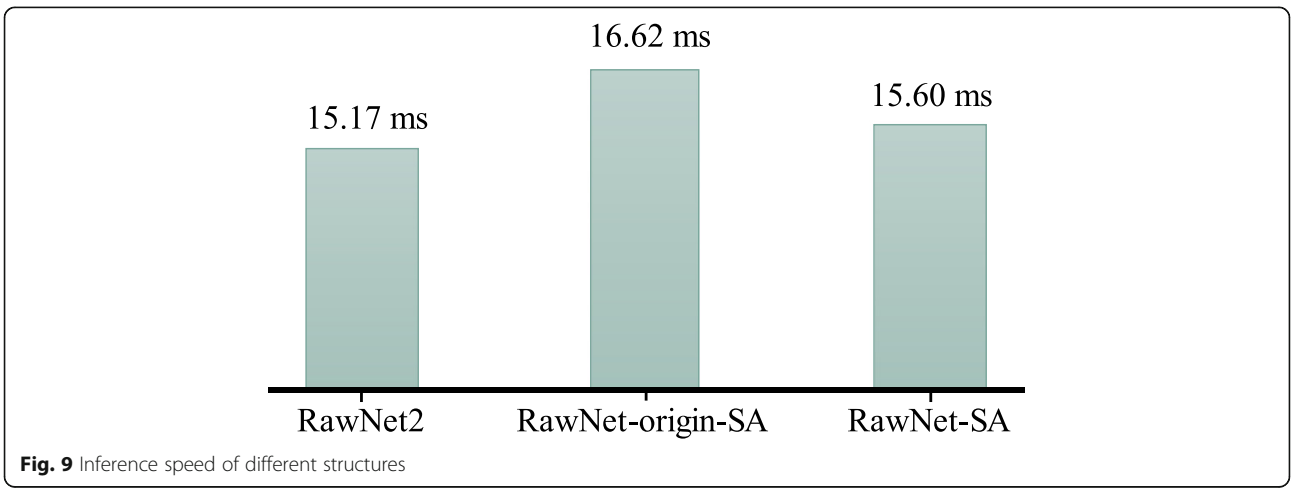

Fig. 9 Inference speed of different structures

Table 5 The effect of different channel squeeze ratios on recognition performance

Squeeze Ratio	VoxCeleb-E			VoxCeleb-H			CN-Celeb		
	EER	DCF08	DCF10	EER	DCF08	DCF10	EER	DCF08	DCF10
1	4.52%	0.24	0.68	7.89%	0.38	0.80	23.12%	0.79	0.97
0.75	2.70%	0.15	0.55	4.88%	0.24	0.65	**21.95%**	**0.76**	**0.93**
0.5	2.72%	0.15	0.52	4.93%	0.24	**0.63**	22.14%	**0.76**	**0.93**
0.25	**2.54%**	**0.14**	**0.47**	**4.52%**	**0.22**	0.65	22.24%	**0.76**	0.94

the network, and the time consumption can be further crop by improved self-attention layers.

3.5 Experiment IV: effect of different channel squeeze ratios on self-attention layer

To investigate the effect of different channel squeeze ratios on self-attention layer, we compare the performance of RawNet-SA under different channel squeeze ratios as illustrated in Table 5. Let channel squeeze ratios $r=d\div c$; here, c is the input channel of self-attention layers and d is the number of output channels of FC-KEY, FC-VALUE, and FC-QUERY. The result shows that r = 0.25 produces the lowest EER in VoxCeleb-E and VoxCeleb-H. The EER in the VoxCeleb-E when r = 0.25 is 2.54%, 0.16% lower than r = 0.75. In the VoxCeleb-H, EER is 4.52%, 0.36% lower than that of r = 0.75. In the CN-Celeb dataset, r = 0.25 is 22.24% EER, only 0.29% higher than r = 0.75 and 0.10% higher than r = 0.5. This is because compressing the number of channels appropriately can remove the redundancy of the model to a certain extent, make the features more robust and the network easier to adapt. In general, the higher the channel squeeze ratio, the better the overall effect of the model, which produce the more the number of model parameters. Unfortunately, because we limit the total number of iterations during network training, the performance of RawNet-SA with a high channel squeeze ratio is worse than that of RawNet-SA with low channel squeeze ratio due to under-fitting. Figure 10 draws the EER changes of RawNet-SA with different channel squeeze ratios during network training. RawNet-SA with lower channel squeeze ratio has faster convergence speed and lower EER. When the channel squeeze ratio is 1, the network is significantly under-fitting.

3.6 Experiment V: influence of different feature aggregation methods

To illustrate the influence of different feature aggregation methods, we compared the performance of RawNet-SA with average pooling, max pooling, self-

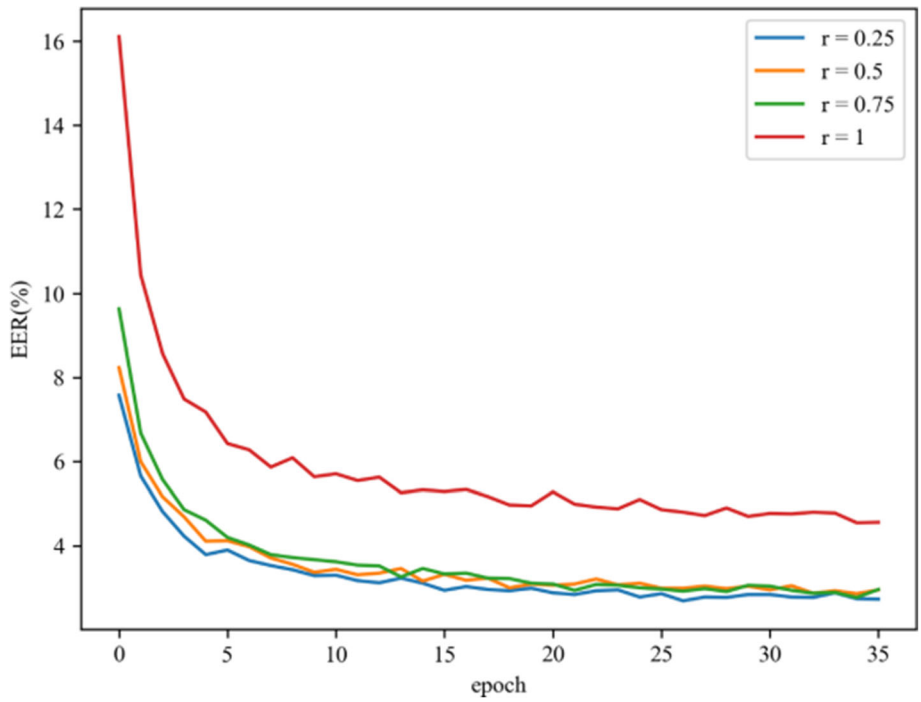

Fig. 10 EER curves during network training

Table 6 The effect of different feature aggregation methods on recognition performance

Aggregation Method	VoxCeleb-E			VoxCeleb-H			CN-Celeb		
	EER	DCF08	DCF10	EER	DCF08	DCF10	EER	DCF08	DCF10
Average Pooling	3.20%	0.16	0.52	5.35%	0.25	0.65	21.85%	**0.73**	**0.93**
Max Pooling	6.66%	0.34	0.80	10.56%	0.47	0.86	23.66%	0.80	0.96
ASP	5.85%	0.31	0.77	9.41%	0.44	0.85	23.50%	0.79	0.96
SAP	4.24%	0.22	0.63	6.97%	0.33	0.78	22.40%	0.76	0.94
Ghost VLAD	3.01%	0.16	0.52	5.01%	0.24	**0.63**	**21.32%**	**0.73**	**0.93**
Bi-GRU	2.80%	0.15	0.52	5.04%	0.25	0.67	22.31%	0.77	0.95
GRU	**2.54%**	**0.14**	**0.47**	**4.52%**	**0.22**	0.65	22.24%	0.76	0.94

attentive pooling (SAP) [43], attentive statistical pooling (ASP) [44], GRU, and Bi-GRU. Table 6 exhibits that GRU has the lowest EER in VoxCeleb-E and VoxCeleb-H. In detail, the EER of GRU in VoxCeleb-E is 2.54% EER, 0.26% lower than that of Bi-GRU. In VoxCeleb-H, EER is 4.52%, which is 0.52% lower than Bi-GRU, indicating that Bi-GRU cannot improve the performance of RawNet-SA, making network convergence more difficult. RawNet-SA GhostVLAD achieves 21.32% EER in CN-Celeb dataset, 0.92% lower than GRU. However, in VoxCeleb-E, EER is 3.01%, 0.47% higher than GRU. In VoxCeleb-H, 5.01% EER is obtained, 0.49% higher than GRU, which indicates that GhostVLAD cannot adapt the network to different data distribution, although GhostVLAD reaches the lowest EER in CN-Celeb dataset. In this experiment, the performance of SAP and ASP is even worse than AP, which means that SAP and ASP are not suitable for the proposed model.

3.7 Experiment VI: effect of VAD and speech separation on voiceprint recognition

To illustrate the effectiveness of VAD and speech separation, we compared the performance of models on CN-Celeb-T. In this experiment, we regard the CN-Celeb-T as a noisy dataset because it inherently contains a lot of noise, such as background music, audience applauded, etc. And CN-Celeb-T-VAD is the dataset processed by VAD and speech separation. Table 7 shows that RawNet-origin-SA* has 16.14% EER in CN-Celeb-T, 0.25% higher than that in CN-Celeb-T-VAD. And RawNet-SA is 15.04% EER in CN-Celeb-T, 0.23% higher than that in CN-Celeb-T-VAD. These results indicate that the effect of network on CN-Celeb-T-VAD generally outperforms that CN-Celeb-T, proving that VAD and speech separation are effective.

We also compared with a speech enhancement + speaker recognition method VoiceID [45] on VoxCeleb1 test set (Vox1T-O) shown in Table 8. In this experiment, like VoiceID, we use the noise and music recordings of MUSAN to generate Vox1T-N and Vox1T-M where Vox1T-N is mixed with noise and Vox1T-M is mixed with music. We also applied the speech separation method on Vox1T-M dataset (Vox1T-M-S) to explore the effectiveness of Spleeter. Experimental results show that the EER of the RawNet-origin-SA* is 8.35% in Vox1T-N, 1.51% less than VoiceID, 5.75% in Vox1-M and 3.38% less than VoiceID. The EER of Vox1T-M-S is 5.52%, which is 0.23% lower than Vox1T-M. While the EER of RawNet-SA in Vox1T-N is 8.90%, 0.96% less than VoiceID, and the EER in Vox1-M is 6.15%, 2.98% less than VoiceID. In Vox1T-M-S, 6.11% of EER is obtained, 0.04% lower than that in Vox1T-M. These results prove that RawNet-origin-SA* and RawNet-SA perform better than VoiceID on corrupted datasets and speech separation is helpful for voiceprint recognition. It can also be seen that compared with VoiceID, RawNet-SA and RawNet-origin-SA* are more sensitive to noise. This is because VoiceID uses data mixed with noise during training, while we do not use any data enhancement trick.

3.8 Experiment VII: the influence of different similarity measurement methods on voiceprint recognition

To illustrate the influence of different similarity measurement methods, we compared the performance of RawNet2, RawNet-origin-SA*, and RawNet-SA using different similarity measurement methods (such as cosine, probabilistic linear discriminant analysis (PLDA) [46], and b-vector [47]). The experiment results are shown in Table 9. We use the PLDA Toolkit[1], which follows the PLDA steps in [46] for our PLDA. Firstly, according to the suggestion of [46], we apply principal component analysis (PCA) to the extracted feature embeddings before PLDA. We use the 128 top principal components of deep voiceprint features to train

[1] https://github.com/RaviSoji/plda

Table 7 The effect of VAD and speech separation

Models	CN-Celeb-T			CN-Celeb-T-VAD		
	EER	DCF08	DCF10	EER	DCF08	DCF10
RawNet2	17.25%	0.58	0.89	16.28%	0.60	0.86
RawNet2*	17.30%	0.60	0.90	16.51%	0.61	0.87
RawNet-MHSA	15.34%	**0.56**	**0.86**	15.16%	**0.57**	**0.85**
RawNet-all-SA	15.51%	0.57	0.91	15.18%	0.59	0.87
RawNet-origin-SA*	16.14%	0.58	0.87	15.89%	0.60	0.87
RawNet-SA	**15.04%**	**0.56**	0.87	**14.81%**	0.58	0.86

"*" denotes that the network is initialized with the trained RawNet2 parameters

the PLDA model. These features are generated from the training set (VoxCeleb2) of the model without normalization or whitening. Then, in the inference stage, the features generated by the test set (VoxCeleb1 and CN-Celeb) are transformed into a latent space, which keeps the same dimensions as the features after PCA. Finally, we calculate the log-likelihood ratio between the two features in latent space as their similarity.

B-vector system regards speaker verification as a binary classification problem, and takes the combination of element-wise addition, subtraction, multiplication, and division of two deep features as the input of binary classification network. Since more combinations will expand the input size of the classifier in the b-vector system and increase the computation overhead, as described in [47], we only use the concatenation of element-wise addition and multiplication in the b-vector system. The input of our b-vector system I is set as follow:

$$\mathbf{I} = [(\mathbf{w}_{\text{query}} \oplus \mathbf{w}_{\text{target}}), (\mathbf{w}_{\text{query}} \otimes \mathbf{w}_{\text{target}})] \tag{20}$$

where $\mathbf{w}_{\text{query}}$ and $\mathbf{w}_{\text{target}}$ denote the deep voiceprint features from the banned anchors and the current anchor respectively, and the symbol $[\cdot,\cdot]$ represents the concatenation of the two vectors. The network of b-vector system is formed by two fully connected layers of a size of [1024, 512] with leaky rectified linear unit (ReLU) activations and dropout of 50%. The similarity of the two voiceprint features is obtained by the output linear layer composed of one neuron. From Table 9, for RawNet2, the cosine similarity in VoxCeleb-E reaches 2.57% EER, 1.21% lower than PLDA and 0.82% lower than b-vector. The cosine similarity of RawNet-origin-SA* in VoxCeleb-H is 4.54% EER, 1.50% lower than PLDA and 1.06% lower than b-vector. As for RawNet-SA, the cosine similarity achieves 22.24% EER in CN-Celeb, 2.43% lower than PLDA and 0.60% lower than b-vector. These results show that the cosine similarity is superior to PLDA and b-vector under all conditions of this experiment. This may be because the models of PLDA and b-vector are trained through the deep voiceprint features extracted from the VoxCeleb2 dataset, and the distribution difference between the training dataset and the test dataset makes the performance of PLDA and b-vector worse than expected.

Table 8 Anti-noise test of different models

Models	Vox1T-O	Vox1T-N	Vox1T-M	Vox1T-M-S
VoiceID [45]	6.79%	9.86%	9.13%	/
RawNet2	2.49%	9.03%	6.18%	6.01%
RawNet2*	**2.25%**	8.48%	5.80%	5.55%
RawNet-origin-SA*	2.31%	**8.35%**	**5.75%**	**5.52%**
RawNet-SA	2.73%	8.90%	6.15%	6.11%

"*" denotes that the network is initialized with the trained RawNet2 parameters

4 Conclusion

With the rapid development of online live streaming industry, we urgently need an intelligent method to identify anchors. Considering that the voiceprint information as one of the important information can represent the identity of the anchor, we propose an anchor voiceprint recognition method in live video streaming using RawNet-SA and GRU. Firstly, the speech of the anchor is extracted from the live streaming by using VAD and speech separation. Then, the feature sequence of anchor voiceprint is generated from the speech waveform with the self-attention network RawNet-SA. Finally, the feature sequence of anchor voiceprint is aggregated by GRU and transformed into deep voiceprint feature vector for anchor recognition. EER is used as the evaluation indicator for the effectiveness of anchor voiceprint recognition. We conducted seven experiments on public datasets. Overall, we verified the effectiveness of self-attention mechanism and GRU, and obtained 22.24% EER on CN-Celeb dataset. Experimental results show that our method obtains good voiceprint recognition performance without abundantly increasing time consumption.

Table 9 The influence of different similarity measurements on the recognition performance

Models	Similarity	VoxCeleb-E			VoxCeleb-H			CN-Celeb		
		EER	DCF08	DCF10	EER	DCF08	DCF10	EER	DCF08	DCF10
RawNet2	Cosine	**2.57%**	**0.14**	**0.52**	**4.89%**	**0.24**	**0.64**	**24.27%**	**0.78**	**0.97**
	PLDA	3.78%	0.19	0.58	6.43%	0.29	0.70	27.76%	0.82	1.00
	B-vector	3.39%	0.19	0.69	5.99%	0.32	0.87	26.16%	0.82	1.00
RawNet-origin-SA*	Cosine	**2.37%**	**0.13**	**0.50**	**4.54%**	**0.22**	**0.63**	**23.49%**	**0.78**	**0.94**
	PLDA	3.51%	0.17	0.59	6.04%	0.28	0.72	27.46%	0.81	0.97
	B-vector	3.17%	0.18	0.69	5.60%	0.29	0.84	26.24%	0.81	1.00
RawNet-SA	Cosine	**2.54%**	**0.14**	**0.47**	**4.52%**	**0.22**	**0.65**	**22.24%**	**0.76**	**0.94**
	PLDA	3.94%	0.19	0.59	6.48%	0.29	0.76	24.67%	0.80	0.96
	B-vector	3.54%	0.21	0.72	6.46%	0.37	0.91	22.84%	0.84	1.00

"*" denotes that the network is initialized with the trained RawNet2 parameters

In the future, we plan to further optimize our model and loss function to improve the representation ability of the model. In recent years, various cross-domain methods based on generative adversarial networks (GAN) have made great progress. In the following work, we will combine GAN to improve the effectiveness of the network for unknown distributed data and make it conveniently applied to practical applications. To meet real-time recognition, the speed promotion will be another important direction of our research. Finally, to better verify the effect of deep features, we will introduce paraconsistent feature engineering to quantify the representation ability of deep features in future work.

Abbreviations

GRU: Gated recurrent unit; VAD: Voice activation detection; GMM-UBM: Gaussian mixture model and universal background model; MFCC: Mel-scale frequency cepstral coefficients; DNN: Deep neural network; TDNN: Time delay neural network; EER: Equal error rate; CNN: Convolutional neural network; ASV: Automatic speaker verification; BLSTM-RNN: Bidirectional long short-term memory recurrent neural network; Bi-GRU: Bidirectional gated recurrent unit; GRCNN: Gated recurrent convolutional neural network; RNN: Recurrent neural network; LC-GRNN: Light convolutional gated recurrent neural network; SOTA: State-of-the-art; LSTM: Long short-term memory; FMS: Feature map scaled; SA: Self-attention; BN: Batch normalize; TAA: Test time augmentation; FRR: False rejection rates; FAR: False acceptance rates; TP: True positive; TN: True negative; FP: False positive; FN: False negative; GAN: Generative adversarial networks

Acknowledgements

The authors thank the associate editor and the anonymous reviewers for their constructive comments and useful suggestions.

Authors' contributions

All the authors made significant contributions to the work. JY, JZ, JL, and LZ proposed the conception of this work and devised the algorithm. JY wrote the draft of this paper and did experiments. JZ checked experiments as well as revised this paper. LZ, JL, and JZ provide instrumentation and computing resources for this study. All authors read and approved the final manuscript.

Funding

This research was supported by the Beijing Municipal Education Commission Cooperation Beijing Natural Science Foundation (No. KZ 201910005007), National Natural Science Foundation of China (No. 61971016).

Availability of data and materials

The datasets generated and/or analyzed during the current study are available in the VoxCeleb repository (https://www.robots.ox.ac.uk/~vgg/data/voxceleb/), CN-Celeb repository (http://www.openslr.org/82/), and MUSAN repository (http://www.openslr.org/17/).

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 4 August 2021 Accepted: 9 December 2021
Published online: 20 December 2021

References

1. D.A. Reynolds, T.F. Quatieri, R.B. Dunn, Speaker verification using adapted Gaussian mixture models. Digital Signal Processing **10**, 19 (2000)
2. N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**, 788 (2011)
3. E. Variani, X. Lei, E. McDermott, I.L. Moreno, J. Gonzalez-Dominguez, *Deep neural networks for small footprint text-dependent speaker verification, in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Florence, Italy, 2014), pp. 4052–4056
4. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, *X-Vectors: Robust DNN embeddings for speaker recognition, in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Calgary, AB, 2018), pp. 5329–5333
5. M. McLaren, L. Ferrer, D. Castan, and A. Lawson, The speakers in the wild (SITW) speaker recognition database, in (2016), pp. 818–822.
6. Y. Qian, N. Chen, H. Dinkel, Z. Wu, Deep feature engineering for noise robust spoofing detection. IEEE/ACM Trans. Audio Speech Lang. Process. **25**, 1942 (2017)
7. G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, Audio replay attack detection with deep learning frameworks, in Interspeech 2017 (ISCA, 2017), pp. 82–86.
8. A. Gomez-Alanis, A.M. Peinado, J.A. Gonzalez, A.M. Gomez, A gated recurrent convolutional neural network for robust spoofing detection. IEEE/ACM Trans. Audio Speech Lang. Process. **27**, 1985 (2019)
9. A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, A light convolutional GRU-RNN deep feature extractor for ASV spoofing detection, in Interspeech 2019 (ISCA, 2019), pp. 1068–1072.
10. A. Gomez-Alanis, J. A. Gonzalez-Lopez, S. P. Dubagunta, A. M. Peinado, and M. Magimai.-Doss, On joint optimization of automatic speaker verification and anti-spoofing in the embedding space, IEEE Trans. Inform. Forensic Secur. 16, 1579 (2021).
11. A. Gomez-Alanis, J.A. Gonzalez-Lopez, A.M. Peinado, A Kernel density estimation based loss function and its application to ASV-Spoofing Detection. IEEE Access **8**, 108530 (2020)

12. A. Nagrani, J. S. Chung, and A. Zisserman, VoxCeleb: a large-scale speaker identification dataset, in Interspeech 2017 (ISCA, 2017), pp. 2616–2620.
13. A. Hajavi and A. Etemad, A deep neural network for short-segment speaker recognition, in Interspeech 2019 (ISCA, 2019), pp. 2878–2882.
14. Y. Jiang, Y. Song, I. McLoughlin, Z. Gao, and L.-R. Dai, An effective deep embedding learning architecture for speaker verification, in Interspeech 2019 (ISCA, 2019), pp. 4040–4044.
15. R.C. Guido, Paraconsistent feature engineering [Lecture Notes], IEEE Signal Process. Mag. **36**, 154 (2019)
16. J. Jung, H.-S. Heo, J. Kim, H. Shim, and H.-J. Yu, RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification, in Interspeech 2019 (ISCA, 2019), pp. 1268–1272.
17. K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, *Learning phrase representations using RNN encoder-decoder for statistical machine translation, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Doha, Qatar, 2014), pp. 1724–1734
18. J. Jung, S. Kim, H. Shim, J. Kim, and H.-J. Yu, Improved RawNet with feature map scaling for text-independent speaker verification using raw waveforms, in Interspeech 2020 (ISCA, 2020), pp. 1496–1500.
19. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, *Attention is all you need, in Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., Long Beach, California, USA, 2017), pp. 6000–6010
20. M. India, P. Safari, and J. Hernando, Self multi-head attention for speaker recognition, in Interspeech 2019 (ISCA, 2019), pp. 4305–4309.
21. P. Safari, M. India, and J. Hernando, Self-attention encoding and pooling for speaker recognition, in Interspeech 2020 (ISCA, 2020), pp. 941–945.
22. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Computation **9**, 1735 (1997)
23. M. Jabreel, A. Moreno, A deep learning-based approach for multi-label emotion classification in Tweets. Applied Sciences **9**, 1123 (2019)
24. K.-H. Woo, T.-Y. Yang, K.-J. Park, C. Lee, Robust voice activity detection algorithm for estimating noise spectrum. Electron. Lett. **36**, 180 (2000)
25. R. Chengalvarayan, Robust energy normalization using speech/nonspeech discriminator for German connected digit recognition, in EUROSPEECH. Vol. **99**, 61–64 (1999)
26. A. Benyassine, E. Shlomot, H.- Su, D. Massaloux, C. Lamblin, and J.- Petit, ITU-T Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications, IEEE Communications Magazine 35, 64 (1997).
27. J. Wagner, D. Schiller, A. Seiderer, and E. André, Deep learning in paralinguistic recognition tasks: are hand-crafted features still relevant? in Interspeech 2018 (ISCA, 2018), pp. 147–151.
28. R. Hennequin, A. Khlif, F. Voituret, M. Moussallam, Spleeter: A fast and efficient music source separation tool with pre-trained models. JOSS **5**, 2154 (2020)
29. O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in medical image computing and computer-assisted intervention—MICCAI 2015, edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Springer International Publishing, Cham, 2015), pp. 234–241.
30. A. Défossez, N. Usunier, L. Bottou, and F. Bach, Music source separation in the waveform domain, ArXiv:1911.13254 [Cs, Eess, Stat] (2019).
31. M. Ravanelli and Y. Bengio, Interpretable convolutional filters with SincNet, ArXiv:1811.09725 [Cs, Eess] (2019).
32. X. Wang, R. Girshick, A. Gupta, and K. He, Non-local neural networks, in 2018 IEEE/CVF Conference on computer vision and pattern recognition (2018), pp. 7794–7803.
33. R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**, 1437 (2018)
34. Y. Zhong, R. Arandjelović, A. Zisserman, in *computer vision—ACCV 2018*, ed. by C. V. Jawahar, H. Li, G. Mori, K. Schindler. GhostVLAD for set-based face recognition (Springer International Publishing, Cham, 2019), pp. 35–50
35. J. S. Chung, A. Nagrani, and A. Zisserman, VoxCeleb2: deep speaker recognition, in Interspeech 2018 (ISCA, 2018), pp. 1086–1090.
36. Y. Fan, J. W. Kang, L. T. Li, K. C. Li, H. L. Chen, S. T. Cheng, P. Y. Zhang, Z. Y. Zhou, Y. Q. Cai, and D. Wang, CN-Celeb: A challenging chinese speaker recognition dataset, in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE, Barcelona, Spain, 2020), pp. 7604–7608.
37. D. Snyder, G. Chen, and D. Povey, MUSAN: a music, speech, and noise corpus, ArXiv:1510.08484 [Cs] (2015).
38. W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, Utterance-level aggregation for speaker recognition in the wild, in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE, Brighton, United Kingdom, 2019), pp. 5791–5795.
39. A. Nagrani, J.S. Chung, W. Xie, A. Zisserman, Voxceleb: large-scale speaker verification in the wild. Computer Speech & Language **60**, 101027 (2020)
40. N. R. Koluguri, J. Li, V. Lavrukhin, and B. Ginsburg, SpeakerNet: 1D depth-wise separable convolutional network for text-independent speaker recognition and verification, ArXiv:2010.12653 [Eess] (2020).
41. J. Deng, J. Guo, J. Yang, N. Xue, I. Cotsia, and S. P. Zafeiriou, ArcFace: additive angular margin loss for deep face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2021).
42. M. India, P. Safari, and J. Hernando, Double multi-head attention for speaker verification, in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE, Toronto, ON, Canada, 2021), pp. 6144–6148.
43. K. Okabe, T. Koshinaka, and K. Shinoda, Attentive statistics pooling for deep speaker embedding, in Interspeech 2018 (ISCA, 2018), pp. 2252–2256.
44. S. Min Kye, Y. Kwon, J. Son Chung, *Cross attentive pooling for speaker verification, in 2021 IEEE Spoken Language Technology Workshop (SLT)* (IEEE, Shenzhen, China, 2021), pp. 294–300
45. S. Shon, H. Tang, and J. Glass, VoiceID Loss: Speech enhancement for speaker verification, in Interspeech 2019 (ISCA, 2019), pp. 2888–2892.
46. S. Ioffe, in *Computer Vision – ECCV 2006*, ed. by A. Leonardis, H. Bischof, A. Pinz. Probabilistic linear discriminant analysis (Heidelberg, Springer, Berlin, 2006), pp. 531–542
47. H.-S. Lee, Y. Tso, Y.-F. Chang, H.-M. Wang, S.-K. Jeng, *Speaker verification using kernel-based binary classifiers with binary operation derived features, in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Florence, Italy, 2014), pp. 1660–1664

Publisher's Note