EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH

Open Access

Multi-task deep cross-attention networks for far-field speaker verification and keyword spotting

Xingwei Liang[1,2], Zehua Zhang[3] and Ruifeng Xu[2*]

Abstract

Personalized voice triggering is a key technology in voice assistants and serves as the first step for users to activate the voice assistant. Personalized voice triggering involves keyword spotting (KWS) and speaker verification (SV). Conventional approaches to this task include developing KWS and SV systems separately. This paper proposes a single system called the multi-task deep cross-attention network (MTCANet) that simultaneously performs KWS and SV, while effectively utilizing information relevant to both tasks. The proposed framework integrates a KWS sub-network and an SV sub-network to enhance performance in challenging conditions such as noisy environments, short-duration speech, and model generalization. At the core of MTCANet are three modules: a novel deep cross-attention (DCA) module to integrate KWS and SV tasks, a multi-layer stacked shared encoder (SE) to reduce the impact of noise on the recognition rate, and soft attention (SA) modules to allow the model to focus on pertinent information in the middle layer while preventing gradient vanishing. Our proposed model demonstrates outstanding performance in the well-off test set, improving by 0.2%, 0.023, and 2.28% over the well-known SV model emphasized channel attention, propagation, and aggregation in time delay neural network (ECAPA-TDNN) and the advanced KWS model Convmixer in terms of equal error rate (EER), minimum detection cost function (minDCF), and accuracy (Acc), respectively.

Keywords Speaker verification, Keyword spotting, Personalized voice trigger, Flow attention

1 Introduction

In an ever-growing array of devices, such as mobile phones, smart homes, and automobiles, personal speech assistants facilitate user interaction with their devices through voice commands. Typically, to initiate a human-computer voice interaction, users must pronounce a specific activation phrase, signaling the commencement of the interaction. Accurate detection of the trigger phrase is crucial, as human-computer voice interaction can only proceed if the phrase is correctly identified. Moreover, many personal speech assistants now incorporate speaker verification systems to thwart unauthorized activation attempts by malicious individuals. This necessitates the device's ability to confirm that the speaker's voice print corresponds to the stored voice print in the device's library. Consequently, activating a speech assistant encompasses two phases: keyword spotting (KWS) to detect the prefixed keywords and speaker verification (SV) to conform to the speaker's identity.

Keyword spotting is detecting specific words or phrases within continuous speech or text. Although it is a subfield of automatic speech recognition (ASR) and natural language processing (NLP), KWS is distinct from ASR and NLP. Its primary goal is to accurately and efficiently recognize pre-set keywords, and it achieves this by minimizing the use of computational resources and

*Correspondence:
Ruifeng Xu
xuruifeng@hit.edu.cn
[1] Konka Group Co., Ltd, Shenzhen, China
[2] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
[3] School of Electronics and Information Engineering, Harbin Institute of Technology, Shenzhen, China

hyperparameters. As such, it can run on low-resource devices. Recently, deep neural network (DNN)-based methods have significantly improved over conventional methods in small-footprint KWS. For instance, DeepKWS [1] is noted for its simpler implementation, higher accuracy, and low computational cost. Depthwise separable convolution [2] separates the channel and spatial domains, reducing the number of the parameters in standard convolution without significantly degrading performance. This technique has been successfully applied to KWS [3]. Noise is a significant factor affecting KWS, and researchers are actively working to reduce its impact. Yu et al. [4] develop a long short-term memory (LSTM) [5] to improve the robustness of small-footprint Keyword Spotting tasks in noisy and far-field environments. Huang et al. [6] propose an adaptive noise cancelation method based on the short-time Fourier transform (STFT) with deferred filter coefficients to retrieve keywords from noisy signals. Ng et al. [7] proposed a lightweight feature interactive convolutional model, called Convmixer, to handle the noisy far-field condition. They use curriculum-based multi-condition training to attain better noise robustness. To build upon their work, we have incorporated fast speech speed conditions into the model's robustness tests and added a speaker verification model to detect KWS and SV simultaneously.

Speaker verification (SV) is the task of confirming whether the current speaker is indeed the valid user they claim to be, based on their unique voice characteristics. If an unconfirmed identity is detected, it may suggest a fraudulent attempt, known as a spoofing attack, designed to deceive the SV system. Such attacks can involve tactics like impersonation, voice conversion, text-to-speech synthesis, or replay attacks, resulting in "malicious speech." In this context, the speech serves as the tool of deception, not necessarily implying a malicious speaker. To counter these threats, an anti-spoofing system is employed. Its role is to differentiate between authentic and counterfeit voice inputs, detecting attempts at malicious speech. This ability is of vital importance in preventing unauthorized access, especially in security-sensitive applications of SV such as voice biometrics [8] or voice-controlled systems. SV models can be broadly classified into two categories: staged and end-to-end. A staged system typically comprises three modules: a speaker feature extraction module, speaker embedding, and a similarity score calculator. The speaker feature extraction module converts the input speech signal into relevant features such as Mel-frequency cepstral coefficients (MFCCs) or other suitable representations that capture speaker-specific characteristics. The extracted features are then passed through a neural network or other machine learning models to generate a fixed-size vector called a speaker embedding, commonly using x-vectors [9], d-vectors [10], or deep embedded features [11–15]. This embedding represents the unique characteristics of the speaker's voice. The similarity score calculator measures the distance between the new embedded feature and the registered embedded feature in the device. A decision is made based on a predefined threshold to determine whether the two speech segments belong to the same speaker. Probabilistic linear discriminant analysis (PLDA) [16], cosine similarity [17], and distance metric learning [18] are popular similarity score calculation techniques. Thresholds can be derived empirically from the development process or calculated using Bayesian decision theory [19, 20].

In contrast, end-to-end approaches aim to perform speaker verification in a single step using a unified model. It inputs raw or pre-processed speech signals and directly outputs a similarity score about the speaker's identity. End-to-end models are usually based on deep neural networks, which are trained to directly optimize the speaker verification task. An example of such a system is the b-vector system [21], which uses kernel-based binary classifiers with binary operation derived features for speaker verification. This system has demonstrated its effectiveness in capturing the unique characteristics of individual speakers. Such end-to-end systems are becoming increasingly popular due to their ability to handle complex patterns and variations in speech, offering promising results in terms of both performance and computational efficiency.

The main difference between the staged and end-to-end SV models lies in their respective loss functions. Early deep neural network-based SV models employed Softmax as the loss function, which maximizes the differences between different speakers in the training set. However, Softmax cannot effectively reduce intra-speaker variance, prompting the proposal of variants such as A-Softmax [22], AM-Softmax [23–26], and AAM-Softmax [27], which can constrain the embedding space to reduce intra-speaker variance. In contrast, end-to-end SV systems generally use a loss function based on metric learning, which learns similarity directly from the training dataset. The metric-based learning loss function depends on the amount of data used to construct independent elements, which can be prototype-based, quadruples, triples, or pairwise losses, depending on the number of corpora used.

Recently, the ECAPA-TDNN [28] has achieved state-of-the-art performance in speaker verification. To build upon their work, we employ Soft Attention instead of residual connections, which helps prevent gradient vanishing and emphasizes important features.

Furthermore, we utilize a multi-task framework and integrate our improved KWS convmixer model to perform KWS and SV tasks jointly.

The acoustic information for KWS and SV are complementary features with relatively little overlap. Those two tasks are usually processed separately and independently to ensure robustness and accuracy. However, in an ideal scenario, both tasks can be performed simultaneously using the same input and outputting results, known as personalized voice triggers. A multi-task learning framework can be used to achieve such an objective while improving both tasks' performance. Sigtia et al. [29] propose a supervised multi-task learning framework to perform speech transcription and speaker spotting tasks simultaneously. The speech transcription branch of the network is trained to minimize the connectionist temporal classification (CTC) loss for speech. In contrast, the speaker spotting branch is trained to assign the correct speaker labels to the input sequences. Yang et al. [30] apply multi-task learning to both tasks by incorporating user information into the keyword spotting (KWS) system. However, to the best of our knowledge, none of these previous works have created a dataset that contains solely a specific, predefined keyword that serves both as a wake word and for speaker verification purposes. We created such a dataset; a detailed description of our dataset can be found in Section 4, "Experiment setup."

The main contributions of this paper are as follows:

- We propose a shared encoder (SE) capable of effectively extracting shared speech features across two tasks while reducing noise's impact on model robustness.
- We introduce a deep cross-attention (DCA) module that enables efficient information flow between the two tasks, further improving the model performance.
- We develop a soft attention (SA) module to replace the residual connection. This soft attention module can better filter important frequency features based on the context information at each frequency point, outperforming the residual connection in performance.

The reset of this paper is structured as follows: Section 2 presents related works in keyword spotting, speaker verification, attention mechanisms, and the multi-task learning framework. Section 3 offers a comprehensive description of the multi-task deep cross-attention network model. Section 4 introduces the dataset and experimental environment. We discuss ablation experiments and robustness tests in Section 5. Finally, we draw conclusions in Section 6.

2 Related works

We review related studies on our proposed multi-task deep cross-attention networks, mainly focusing on keyword spotting, speaker verification, attention mechanism, and multi-task framework used in KWS and SV.

2.1 Keyword spotting

Keyword spotting (KWS) identifies specific, predefined spoken terms in the input utterance. A common approach for KWS is template-based methods, which create a template for each keyword and compare the incoming speech signal against these templates using similarity metrics, such as dynamic time warping (DTW). Another approach is model-based methods, which train a machine learning model, such as hidden Markov model (HMM), Gaussian mixture model (GMM), or a deep learning model, to recognize and classify the keywords. KWS systems have been used in many practical applications, especially for voice-activated devices. Therefore, designing KWS systems with small memory and computational footprint is essential for deployment on resource-constrained devices. Deep-KWS trains a deep neural network (DNN) to directly predict the keyword(s) with a small footprint and low computation cost. Sainath et al. [31] proposed CNN to replace DNN, which resulted in better performance because DNN does not consider speech's local temporal and spectral correlation. To further reduce the memory footprint, a more recent work applied time delay neural network (TDNN), attention mechanism, and temporal convolutional network (TCN) [32–34] to KWS. Xu et al. [3] proposed a model with a stack of depthwise separable convolution layers with residual connections, improving the performance and resulting in a smaller memory footprint.

Furthermore, the Convmixer model comprises three blocks: a pre-convolutional block, a convolutional mixer block, and a post-convolutional block. Convmixer uses deep separable convolutions with large convolutional kernels in the pre-and post-convolutional blocks to capture long-term contextual information with fewer model parameters. The convolution mixer introduces a novel module, the Convmixer block, to extract speech internal embedding features in both the time and frequency domains. The module employs a two-dimensional convolution for frequency domain feature extraction to capture rich information from the frequency domain. Pointwise convolution then compresses the feature information from 3-D to 2-D. The time-frequency domain features are then extracted using depthwise separable convolution [2]. In time-domain feature extraction, depthwise separable convolution is used to extract time-domain features.

2.2 Speaker verification

In our work, we focus on a form of SV known as text-dependent SV. This refers to scenarios where the lexical content of the utterances is fixed to a specific phrase. If the content is not fixed to a particular phrase, the task is termed as text-independent SV. Early statistical methods for speaker verification tasks utilized GMMs to create a probabilistic model of a speaker's voice characteristics by fitting a mixture of Gaussian distributions to the extracted features from the speech signal [35]. The GMM-UBM approach extends GMM by incorporating a universal background model (UBM), which captures the general characteristics of human speech, and each speaker's model is adapted from the UBM. The most recent SV technique employs the i-vector [36] approach, extracting a low-dimensional fixed-length representation for each speech segment [16]. With the rise of deep learning, high-level representations from the input speech signal are learned and utilized for speaker verification. Recent advancements in deep learning have led to the development of embedding-based methods for speaker verification, such as x-vectors and d-vectors. These methods involve training deep neural networks to extract speaker embeddings, which are then compared using probabilistic linear discriminant analysis (PLDA) to determine if the speech samples belong to the same speaker.

ECAPA-TDNN is a widely recognized and superior SV network architecture that improves upon the traditional time-delay neural network (TDNN) [37] using statistical pooling to map variable-length speech to fixed speaker embedding features. The initial frame layer is reorganized into one-dimensional Res2Net modules with influential jump connections, similar to SE-ResNet. Squeeze-and-excitation [38] blocks are introduced in these modules to model the interdependence of channels, which extends the temporal context of the frame layer by rescaling the channels based on the global properties of the recordings. Additionally, to exploit complementary information learned at different levels of complexity in different network layers, features are aggregated and propagated from different layers. Finally, the statistical pooling module is improved to have channel-dependent frame attention, allowing the network to focus on different subsets of frames during the statistical estimation of each channel.

2.3 Attention mechanism

Self-attention, multi-head attention, transformers [39], and other models of deep learning shine. The attention mechanism [33, 40] plays a key role in image processing, natural speech processing, and audio signal processing. This mechanism enables the model to focus on potential feature information. Although the attention mechanism is proven very effective, it still needs to improve its algorithm complexity. When the complexity of the self-attention model reaches $O(n^2)$, especially when processing sequential signals such as speech, the computation amount of the model will increase exponentially with the increase of sequence length. This makes computing devices take a lot of multiplication and addition, which can be fatal for low-power or portable devices. Many researchers have undertaken studies to decrease the algorithmic complexity of the attention mechanism in models. Model structures such as Linformer [41], Performer [42], Nystromformer [43], and Flowformer [44, 45] have been developed to address this issue. These models significantly reduce the algorithmic complexity to linear time, or $O(n)$, contributing to more efficient computations.

2.4 Multi-task framework

Multi-task learning is an approach inspired by human behavior, wherein multiple tasks are handled simultaneously. This method leverages the domain information in the training signals of related tasks as an inductive bias to enhance generalization [46]. This approach enables the model to extract features from associated tasks concurrently, allowing different tasks to share pertinent knowledge during the learning process. Consequently, the correlation between tasks leads to improved performance on each task [47].

Recently, multi-task frameworks have been employed to integrate speech and speaker information to enhance the performance of speaker verification (SV) systems [48, 49]. Studies have demonstrated that incorporating frame-level phonetic information, learned through multi-tasking prior to the pooling layer, can facilitate the SV system in distinguishing speaker-specific information more effectively. Sigtia et al. [29] and Jung et al. [50] explored the joint optimization of keyword spotting (KWS) and SV within a single network. In the speaker-dependent voice trigger task, the SV system aims to verify the speaker's identity in utterances containing prefixed keywords.

3 Multi-task deep cross-attention networks (MTCANet)

In this section, we depict the architecture of the proposed MTCANet that performs KWS and SV tasks simultaneously. It contains three core modules: a deep cross-attention (DCA) module, a multi-layer stacked shared encoder (SE), and a soft attention (SA) module.

3.1 Overall structure of multi-task deep cross-attention network

Based on Convmixer and ECAPA-TDNN, we have made several improvements to make the model applicable to both SV and KWS tasks. The overall structure of the proposed MTCANet is shown in Fig. 1. The input

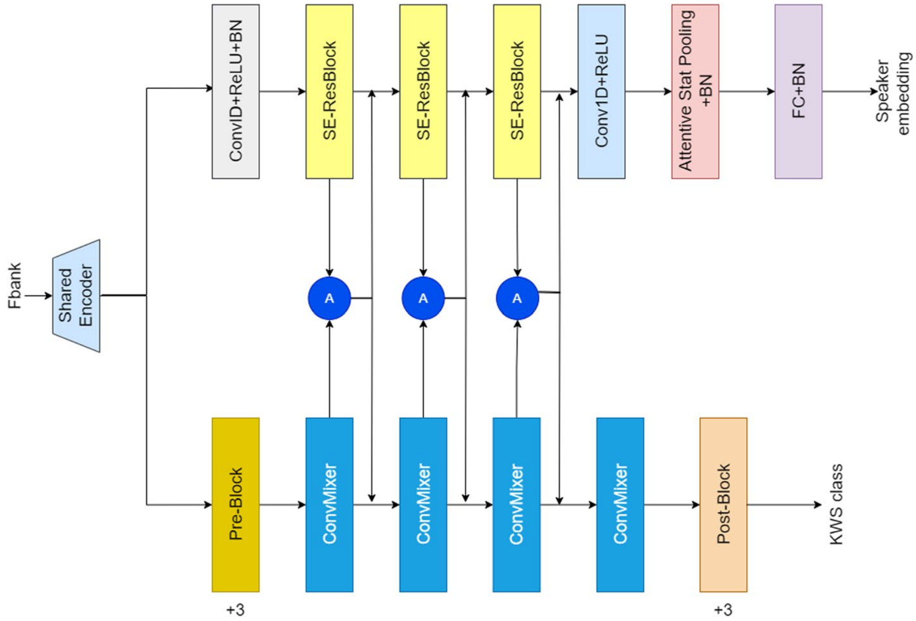

Fig. 1 Model architecture of the proposed multi-task deep cross-attention network (MTCANet). In this instance, it uses the KWS branch as query and the speaker verification branch as key and value, effectively enhancing the utilization efficiency of intermediate embedded features extracted by the two tasks

features of the model are 80-dimensional FBank features, which are further extracted by a shared encoder to reduce the impact of noise [50, 51]. The shared encoder involves convolution operations that distill the most useful information from the FBank features while reducing the impact of unwanted noise. The noise reduction effect comes from the learning process, in which the model is trained to focus on the parts of the input that are most useful for its tasks (such as keyword spotting or speaker verification). Because the noise is usually less informative than the speech signal, the model learns to downweight or ignore it, effectively reducing its impact. By "noise," we refer to any irrelevant or unwanted data in the audio signal, such as background sounds, interference...etc. It helps enhancing the robustness of the model, making it more resistant to variations in the input data. The shared features are then fed into two branches to extract speaker embedding features and perform keyword classification. We add a deep cross-attention module between the two tasks, which uses one branch as a query and the other as a key and value. This can effectively improve the utilization efficiency of intermediate embedded features extracted by the two tasks. Finally, SV generates speaker feature vectors by attentive stat pooling and a fully connected layer. The KWS branch generates the probability of recognition through three post-processing modules, a fully connected layer, and the Sigmoid activation function.

3.2 Shared encoder (SE)

Shared encoders are widely used in language models and speech recognition [52, 53]. Shared encoders provide an effective way to handle multiple related tasks, enabling more robust and accurate models: (1) reusing the same parameters across different tasks leads to less computational and memory resources; (2) help to reduce the impact of noise by learning a common representation that emphasizes the relevant speech signals and downplays the noise; (3) shared encoders help maintain consistency in the features extracted across different tasks, such multiple tasks are learned simultaneously; additionally, it can efficiently learn rich representation of the input data that are useful across multiple tasks; (4) it can be pre-trained on one task and then fine-tuned on another, enabling transfer learning. Such speed up training time and improving performance. Inspired by this, we added a shared encoder to the speaker validation model and the keyword spotting model to further extract the generated FBank features, as shown in Fig. 2. The shared encoder consists of *n* stacked in separate blocks. Each block consists of a separable convolution of convolution kernel *k*, pointwise convolution, batch normalization, and Swish activation function. Separable convolution can capture the features between the upper and lower frames at each frequency point and reduce the parameters and computation of the model. "Frequency Points" correspond to distinct frequencies in each frame. Here, the frame is typically defined as short segments of the audio signal. The

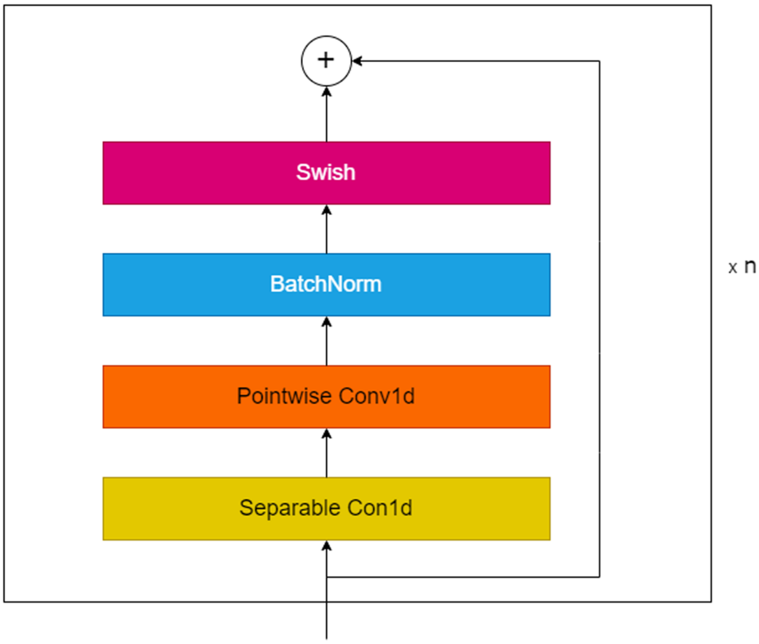

Fig. 2 Shared encoder: a multi-layer stacked shared encoder is utilized across both tasks to reduce the impact of noise on the recognition rate

frequency points are obtained by transforming the raw audio signal into the frequency domain, where different frequencies correspond to different "points." Separable convolution processes each frequency point separately, capturing the changes of that frequency over time (between the "upper and lower frames", i.e., successive time windows). Subsequently, the pointwise convolution combines the information from all frequencies to capture the interactions between them, allows characteristic information to flow between frequency points. The number of separated convolution blocks stacked in this paper is 6, and the convolution kernel sizes are 3, 5, 7, 3, 5, 7.

3.3 Keyword spotting branch

The structure of the keyword spotting network is shown in Fig. 1, which consists of four parts: pre-blocks, ConvMixer blocks, post-blocks, and deep cross attention blocks. Pre-blocks and post-processing blocks comprise depthwise separable convolution, batch normalization, and Swish. The convolution kernel sizes of the pre-blocks are 5,7,1, while the convolution kernel sizes of the post-processed blocks are 17, 19, 1. The structure of the Convmixer block is shown in Fig. 3.

Temporal-frequency coding module, temporal coding module and mixing module constitute the Convmixer block. The temporal-frequency coding module utilizing 2-D convolution can be represented by the following formula:

$$\begin{aligned} x_1 &= \varrho \circ F_1(x_{in}) \\ x_2 &= \varrho \circ (\boldsymbol{F_1}(x_1)) \\ x_3 &= \varrho \circ \mathrm{BatchNorm}(F_2(x_2)) \end{aligned} \tag{1}$$

where ϱ represents the Swish activation function. 2-D convolution $F_1(\cdot)$ extends the features of a single channel to multiple channels. The channel dimension effectively represents the rich information inherent in the time-frequency domain. 2-D depthwise separable convolution $\boldsymbol{F_1}(\cdot)$ is used to extract the time-frequency features of each channel. Pointwise 2-D convolution $F_2(\cdot)$ is used to compress the channel dimension into a single channel again. Subsequently, depthwise separable 1-D convolution $f_1(\cdot)$ facilitates temporal domain feature extraction, as shown in the following formula:

$$x_4 = \varrho \circ \mathrm{BatchNorm}(\boldsymbol{f_1}(x_3)) \tag{2}$$

The convolution across time-frequency and time-domain spaces yields frequency- and time-enriched embeddings. Mixing layers are then incorporated to enable the flow of information throughout the global feature channel, as follows:

$$\begin{aligned} u_{*,i} &= (x_4)_{*,i} + W_2 \cdot \delta\big(W_1 \cdot \mathrm{LayerNorm}(x_4)_{*,i}\big) \\ (x_5)_{j,*} &= u_{j,*} + W_4 \cdot \delta\big(W_3 \cdot \mathrm{LayerNorm}(u)_{j,*}\big) \end{aligned} \tag{3}$$

W_1 and W_2 represent the learnable weights associated with the linear layers of the temporal channel, which are shared across all frequency points (i). These weights are

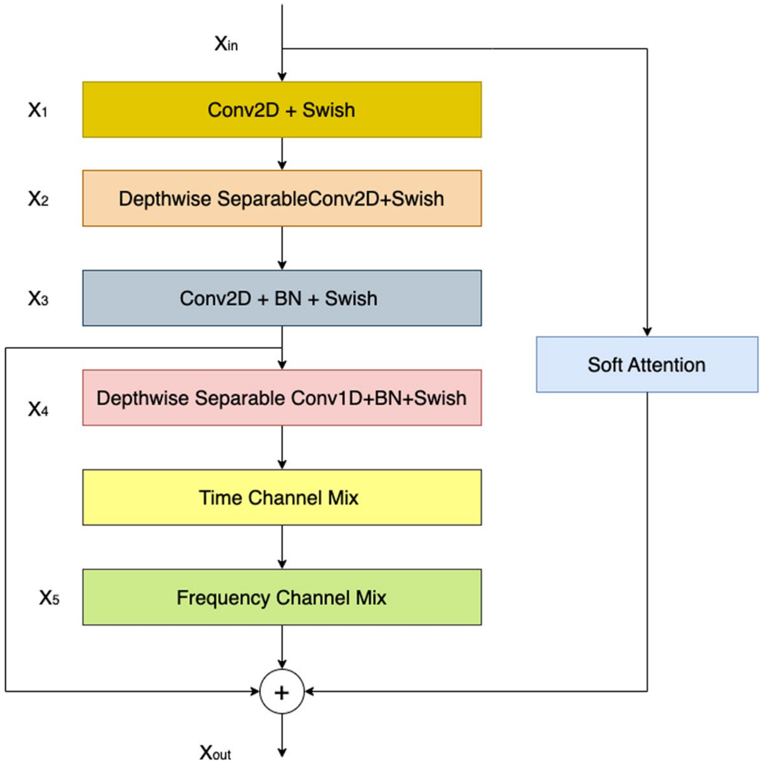

Fig. 3 Keyword spotting branch: the structure diagram of Convmixer block. The convolution across both time-frequency and time-domain spaces yields frequency- and time-enriched embeddings. Soft attention from the previous output and 2D features connected to the block output

adjusted during the training process to capture the temporal dynamics of the input signal. On the other hand, W_3 and W_4 denote the learnable weights of the linear layers within the frequency channels, which are shared across all temporal instances (j). These weights learn the unique characteristics of different frequencies within the input signal.

Finally we added soft attention from the previous output and 2D features connected to the block output.

$$x_{out} = x_3 + f_1(\boldsymbol{f_2}(x_{in})) + x_5 \tag{4}$$

where $\boldsymbol{f_2}(\cdot)$ and $f_1(\cdot)$ represent depthwise separable 1-D convolution and pointwise 1-D convolution in soft attention.

3.4 Speaker verification branch

In the SV network, the output of the shared encoder is processed by three SE-Res2Blocks, as illustrated in Fig. 4. These consist of 1-D convolutions, ReLU activation functions, and batch normalization. The Res2 Dilated Conv1D in each SE-Res2Block has a kernel size of 3 and a dilation rate of 2, 3, and 4, respectively. In the Res2 Dilated Conv1D, since the input to each dilated convolution contains the residuals of all the past convolution, it is called Res2. The above process can be represented by Eq. 5, where κ represents the ReLU activation function, $f_2(\cdot)$ and $f_3(\cdot)$ represent 1-D convolution, and $\boldsymbol{f_3}(\cdot)$ represents Res2 dilated 1-D convolution.

$$\begin{aligned} y_1 &= \text{BatchNorm}((\kappa \circ f_2(y_{in}))) \\ y_2 &= \text{BatchNorm}(\kappa \circ \boldsymbol{f_3}(y_1)) \\ y_3 &= \text{BatchNorm}((\kappa \circ f_3(y_2))) \end{aligned} \tag{5}$$

The SE-blocks enable the calculation of attention weights based on the inter-channel correlation, which enhances important features and attenuates unimportant ones, as shown in Fig. 5. The SE-block can be expressed as Eq. 6.

$$y_4 = y_3 \times \sigma \circ f_5(\kappa \circ f_4(\boldsymbol{p}(y_3))) \tag{6}$$

First, the mean value of features is calculated using the adaptive average pooling layer $\boldsymbol{p}(\cdot)$, then the mean value is adjusted using 1-D convolution $f_4(\cdot)$ and ReLU activation function. Then, the weight is generated by 1-D convolution $f_5(\cdot)$ and Sigmoid activation function. The resulting weights are multiplied by the input of the SE-block, which further reinforces the important characteristic information.

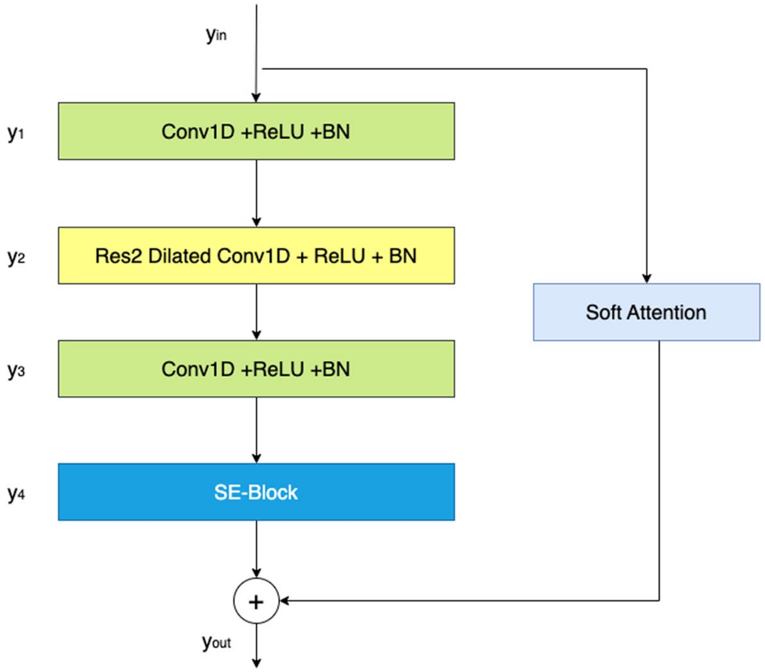

Fig. 4 Speaker verification branch: the structure diagram of an SE-Res2Block

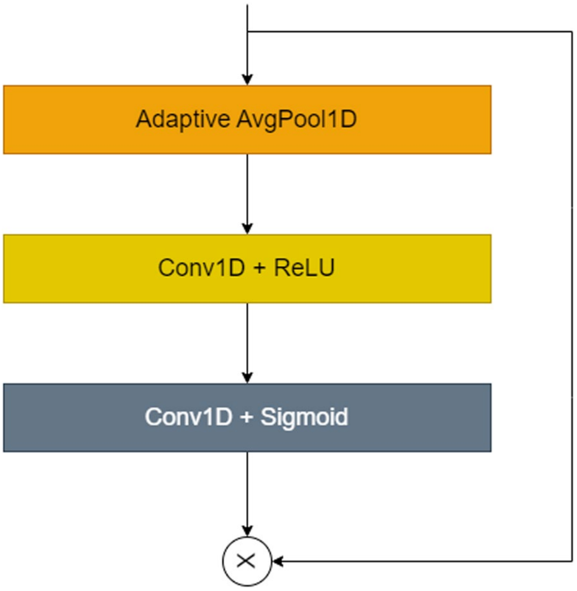

Fig. 5 The structure diagram of SE-Block

Soft attention instead of residual connections are employed to prevent gradient vanishing and emphasize significant features. The output of the SE-Res2 block is the sum of SE block and soft attention as shown below:

$$y_{out} = y_4 + f_6(\boldsymbol{f}_4(y_{in})) \tag{7}$$

where $\boldsymbol{f}_4(\cdot)$ and $f_6(\cdot)$ represent depthwise separable 1-D convolution and pointwise 1-D convolution in soft attention.

3.5 Deep cross attention module (DCA)

To enable embedded information within the speech to be shared by SV and KWS branches, we add a deep cross attention (DCA) module between the two tasks. In classical attention algorithms, the Softmax [54, 55] activation function and the multiplication of matrices make attention inevitably introduce quadratic complexity. The squared complexity causes the computational cost to increase significantly as the speech duration grows. We developed DCA for two branches crossing attention as shown in Fig. 6.

To utilize the conservation property of the flow network theory, we realize the conservation of sources outflow and sinks inflow through normalization.

$$\hat{I} = \phi(Q)\sum_{j=1}^{m}\frac{\phi(K_j)^T}{O_j}, \widehat{O} = \phi(K)\sum_{i=1}^{n}\frac{\phi(Q_i)^T}{I_i} \tag{8}$$

where $\hat{I} \in \mathbb{R}^{n\times1}$ indicates the amount of information obtained by each sink after the competition when the amount of source outflow information is fixed. $\widehat{O} \in \mathbb{R}^{m\times1}$ indicates the amount of the information supplied by each source when the amount of inflow information is fixed. Finally, the attention mechanism based on flow network theory can be expressed as:

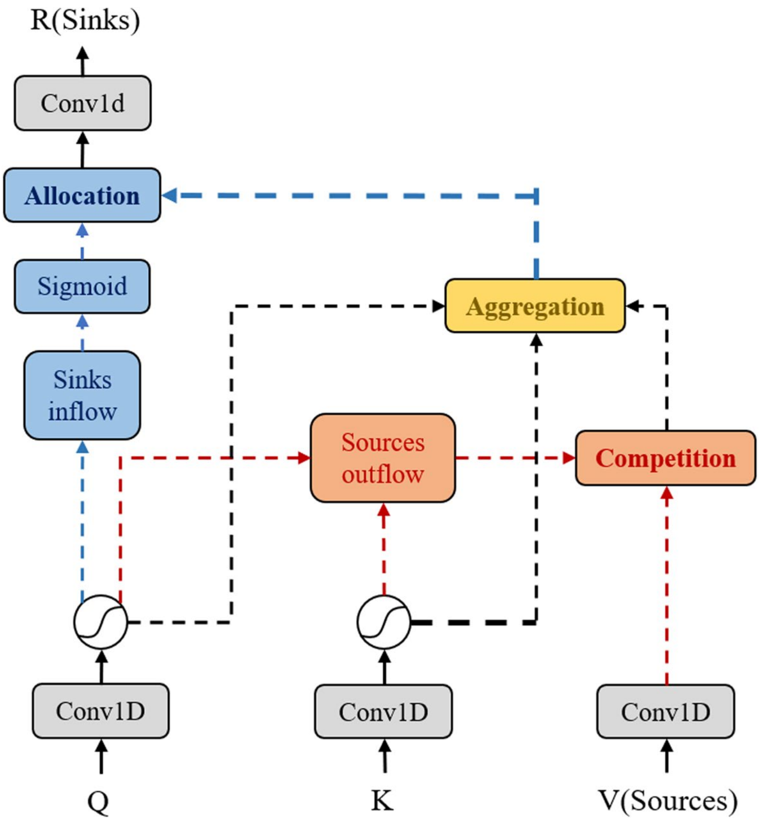

Fig. 6 Diagram of deep cross-attention module

$$\begin{aligned} \text{Competition}: \widehat{V} &= \text{Softmax}(\widehat{\text{O}}) \odot V \\ \text{Aggregation}: A &= \frac{\phi(Q)}{I}\left(\phi(K)^{T}\hat{V}\right) \\ \text{Allocation}: R &= \text{Sigmoid}(\hat{I}) \odot A \end{aligned} \tag{9}$$

where $\odot$ indicates the element-wise multiplication. Q represents the current feature being considered. K represents all other features in the input. V is associated with each Key and gets summed up to produce the final output based on the weight determined by the interaction of query and key. $\hat{V} \in \mathbb{R}^{m\times d}$ represents the source after the competition, which is reweighted according to the inflow conservation. $A \in \mathbb{R}^{m\times d}$ is the aggregated information source, which is calculated by the associativity of matrix multiplication to reduce the computational complexity. $R \in \mathbb{R}^{n\times d}$ represents the results of this new attention mechanism. The whole process of the DCA based on flow network theory is shown in Fig. 6. The outflow of sources competes with each other as the amount of information supplied has been predetermined. Conserving the incoming flow of sinks, the source competes for non-trivial information aggregation; the sink thereby allocates and filters aggregated information from the outflow of the source. In calculating the SV to KWS attention, the SV feature is used as Q, the low-frequency KWS feature is used as K and V, and vice versa. DCA effectively improves the efficiency of the use of information between the two tasks and effectively improves the model's performance.

3.6 Multi-task weighted loss function

Additive angular margin loss (AAM-Softmax) [27] was first used in face and speech recognition. This loss function can be used to increase further the intra-class compactness and inter-class differentiation of extracted features and edges, play a crucial role in discriminant embedded learning, and lead to significant performance improvements, as shown below:

$$L_{\text{AAM-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i,i}+m\right)\right)}}{Z} \tag{10}$$

where Z is defined as: $e^{s\left(\cos\left(\theta_{y_i,i}+m\right)\right)} + \sum_{j=1,j\neq i}^{c} e^{s\left(\cos\left(\theta_{j,i}\right)\right)}$. The AAM-Softmax loss function is used for speaker verification tasks. The KWS task uses the BCEWithLogitsLoss function, as shown below:

$$L_{BCELogits} = -\omega_n \left[y_n \cdot \log_{\sigma}\left(x_n\right) + \left(1 - y_n\right) \cdot \log\left(1 - \sigma(x_n)\right)\right] \tag{11}$$

To balance the order of magnitude of the loss function, AAM-Softmax and BCEWithLogitsLoss are weighted, as shown in the following equation:

$$L = \alpha L_{BCELogits} + L_{AAM-Softmax} \tag{12}$$

4 Experiment setup

We evaluate our proposed MTCANet architecture on a meticulously curated set of datasets. In this section, we provide an overview of the datasets used in our study and describe the experimental setup employed for the evaluation.

4.1 Text-dependent corpora

For both KWS and SV tasks, we have created a custom dataset named Xi-aokangKWS, specially designed for wake word recognition and speaker recognition tasks. The dataset consists of 240,000 near- and far-field utterances from 2025 participants, recorded in three different environmental setups with an SNR ranging from − 5 to 15 dB at two speech speeds.

4.2 Recording environment setup

The audio is recorded in a soundproof room with background noise levels of less than 45 dB and without reverberation. Only stationary background noise is present, and non-stationary noise is absent. Near-field speech is recorded using a single-channel microphone at a distance of 0.1m in a quiet environment. In comparison, far-field speech is recorded at distances of 1 m, 3 m, and 5 m using a multi-channel microphone in three simulated environments: quiet, TV noise, and TV noise plus ambient noise. TV noise is created by playing a documentary on the recording TV, while ambient noise includes keyboard typing, knocking, footsteps, talking, and drinking water. At the recording time, TV noise is set to different volumes. All audio is saved in the WAVE format with a 16 kHz sample rate and 16-bit bit width. The recorded audio is of moderate volume and free from spectral distortion, amplitude truncation, and frame loss.

4.3 Recording participants and recording approaches

A total of 2025 participants joined Xi-aoKangKWS dataset creation. Two thousand are involved in near-field audio recording, while 500 participants are involved in far-field audio recording. The gender ratio is 50% male and 50% female. Age-wise, 20%, 60%, and 20% belonged to the age groups of 8 to 18, 18 to 45, and 46 to 60, respectively. The geographic distribution of the recorders is balanced, with half from Northern China and a half from Southern China. The recorders speak Mandarin without any discernible dialect. All information other than gender and age is kept confidential.

The target keyword for the recording is "Xiaokang Xiaokang." Ten similar-sounding confusing keywords, such as "Xiaoguang Xiaoguang" and "Xiaogang Xiaogang," are also used. Near-field speech is recorded in a quiet setting, with each participant uttering the target keywords 15 times at normal and fast speeds and the confusing keywords twice at normal speed and once at fast speed. Far-field speech is recorded in simulated environments. In both scenarios, participants read the target keyword ten times at microphone distances of 1 m, 3 m, and 5 m at normal and fast speech rates. In total, 120,000 near-field clips and 12,000 far-field clips are recorded.

The captured audio must have proper gaps between command words, consistent volume, and accurate pronunciation without heavy accents to ensure recording quality. During the data annotation, annotators must verify that the target speech pronunciation is clear and complete, with minimal interference from background noise. Audio samples with issues like clipping, data loss, or non-human readings should be considered unusable and discarded.

Ultimately, 2025 participants took part in the recording process, of which 1822 recorders' data were used for the training, validation, and seen test sets. Data from 203 participants were reserved for the unseen test set. The seen test set implies that the speaker is present in the training set, but the seen test set and the training set do not overlap. The unseen test set indicates that the speaker does not appear in the training set.

4.4 Training details

The frame length is 20 ms and frame shift is 10 ms. The number of frames is 200, which is determined by the data set that we collect, and for speech clips that are less than 200 frames we pad them. FBank has a dimension of 80. The Adam method is the optimizer, and all model hyperparameters are randomly initialized. The initial learning rate is 1×10^{-3}, and the learning rate decays exponentially by 0.9. Training is stopped when the learning rate decreases to 5×10^{-5}. The weight α of the loss function is 2.

4.5 Evaluation metrics

In speaker verification system, there are four cases of system recognition: true positive (TP), false negative (FN), false positive (FP), true negative (TN). The above four cases in turn indicate that the speaker is the target speaker and is identified, the speaker is the target speaker but is not identified, the speaker is not the target speaker but is identified, and the speaker is not the target speaker and is not identified. Thus, the false acceptance rate (FAR) and false rejection rate (FRR) can be calculated, as shown in the following formula:

$$\begin{aligned} FAR &= \frac{FP}{FP + TN} \\ FRR &= \frac{FN}{TP + FN} \end{aligned} \tag{13}$$

When the detection threshold becomes small, FRR will decrease and FAR will increase. When the detection threshold increases, the FRR will increase and the FAR will decrease. Equal error rate (EER) refers to the values of FAR and FRR when FAR and FRR are equal. The lower the EER, the better the speaker verification system.

The EER can reflect the performance of the system to a certain extent, but it cannot accurately represent the cost of system error recognition. Therefore, we use the minimum decision cost function (minDCF) as an evaluation metric to represent the cost caused by error recognition. In the speaker verification system, two kinds of alarms will occur: one is that the test speech is designated to the speaker but the system does not recognize; the other is to test that the speech is not a designated speaker and the system recognizes that speaker. These two conditions are defined as false-reject (Miss) and false-accept (False-Alarm) respectively. They have different influences on the detection cost. Different weights can be used to evaluate the detection cost of speaker recognition system in different use environments. The detection cost function (DCF) is shown as follows:

$$DCF = C_{Miss} \cdot P_{Target} \cdot FRR + C_{FA} \cdot (1 - P_{Target}) \cdot FAR \tag{14}$$

where the parameters C_{Miss} and C_{FA} are the cost of Miss and cost of False-Alarm, respectively, P_{Target} is the a priori probability that the test segment speaker is the target speaker. In this paper, we use NIST 2016 [56] as a reference, with C_{Miss} =C_{FA}=1, P_{Target}=0.005. minDCF is the value that minimizes DCF in the process of changing the decision threshold. minDCF can better reflect the cost of misrecognition by speaker verification systems.

Accuracy (Acc) is a commonly used evaluation metric of keyword spotting model. It is obtained by dividing the number of correctly classified samples by the total number of samples. The calculation formula is as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

TP,TN,FP and FN are similar to those above, respectively indicating that they are target keywords and recognized; they are not target keywords and not recognized; they are not target keywords and recognized; they are target keywords and not recognized. The higher the Acc, the better the keyword spotting system

5 Result and discussion

In this section, we conduct comprehensive experiments to assess the performance of the proposed MTCANet. We will delve into the experimental results and provide a detailed discussion.

5.1 Ablation study

Table 1 presents the results of the ablation study, with the equal error rate (EER) and minimum decision cost function (minDCF) to assess the performance of the SV model and the accuracy rate to evaluate KWS model performance. The term "seen" refers to cases where the prosodic speaker was part of the training set, but the speech

Table 1 The ablation study of the effect of different modules on the model performance

	Model	Seen			Unseen		
		EER (%)	minDCF	Acc (%)	EER (%)	minDCF	Acc (%)
Without noise	Convmixer	-	-	96.25	-	-	94.44
	ECAPA-TDNN	0.17	0.009	-	1.88	0.149	-
	Baseline	0.18	0.010	96.16	1.83	0.125	94.77
	+SA	0.18	0.007	96.45	1.69	0.113	95.15
	+SE	0.14	0.006	96.56	1.68	0.112	95.06
	+DCA	0.11	0.006	98.37	1.55	0.110	96.57
With noise	Convmixer	-	-	91.96	-	-	90.36
	ECAPA-TDNN	1.62	0.077	-	4.18	0.223	-
	Baseline	1.64	0.078	92.08	4.16	0.214	90.61
	+SA	1.59	0.079	92.21	4.10	0.209	90.91
	+SE	1.58	0.069	92.89	4.07	0.208	91.11
	+DCA	1.27	0.059	95.61	3.98	0.200	93.08

segment was not; "not seen" denotes cases where the prosodic speaker was absent from the training set. Test segments encompassed both near-field and far-field scenarios and normal and fast speech rates. The baseline model involved directly combining ECAPA-TDNN and Convmixer. Building upon this, the soft attention (SA), shared encoder (SE), and deep cross-attention (DCA) modules were added in turn to verify the impact of the modules on performance. The training set loss after adding different modules to the baseline is shown in Fig. 7. After adding SA and SE, the loss of the training set is slightly lower than the baseline model. The most obvious decrease in loss is DCA, which indicates that our proposed DCA has the most significant impact on loss. Loss does not intuitively reflect the performance of each module, and the results of objective evaluation metrics are described below.

From the results presented in Table 1, it can be observed that the performance of the baseline model across various environments is comparable to that of separately trained ECAPA-TDNN and Convmixer models. The SA module contributes to a notable improvement in model performance at the expense of 0.1 million model parameters, and altering the model structure proves to be more advantageous than incorporating a skip connection. The SE module demonstrates the capability to pre-extract speech-related embeddings, effectively mitigating the impact of noise on the model. Consequently, performance enhancements in noisy conditions surpass those observed in noise-free environments. Furthermore, the DCA module emerges as a highly effective method for fusing the two models, exhibiting the most significant performance improvement among the tested modules. In noise-free unseen test sets, the DCA enhances EER, minDCF, and ACC by 0.13%, 1.79%, and 1.51%, respectively. In noisy unseen test sets, those improvements amount to 0.09%, 3.85%, and 1.97% for EER, minDCF, and ACC, respectively. The DCA module successfully optimizes the utilization of SV and KWS features, substantiating its efficacy in model fusion for improved performance.

5.2 Robustness experiment

This section evaluates the model performance under different environments to evaluate the robustness of the model.

5.2.1 *Performance of model under different SNR environments*

As illustrated in Table 2, our evaluation considers musical and environmental noise, with signal-to-noise ratios (SNRs) ranging from − 5 dB to 15 dB. The test speech samples comprise both near-field and far-field speech recordings. In a low SNR environment of − 5 dB, the noise signal severely masks the speech signal, causing the equal error rate of the speaker recognition model to rise to 9.54% in unseen musical environments. However, considering the minDCF value of up to 0.453, the system remains functional. The accuracy of keyword spotting declines to 86.97%. Although it is lower than clean speech, the recognition accuracy remains acceptable. As the SNR increases to 0 dB, the EER and minDCF values decrease considerably while the ACC value significantly increases. When the SNR reaches 10 dB, the noise impact diminishes substantially, and the performance metrics become comparable to those obtained

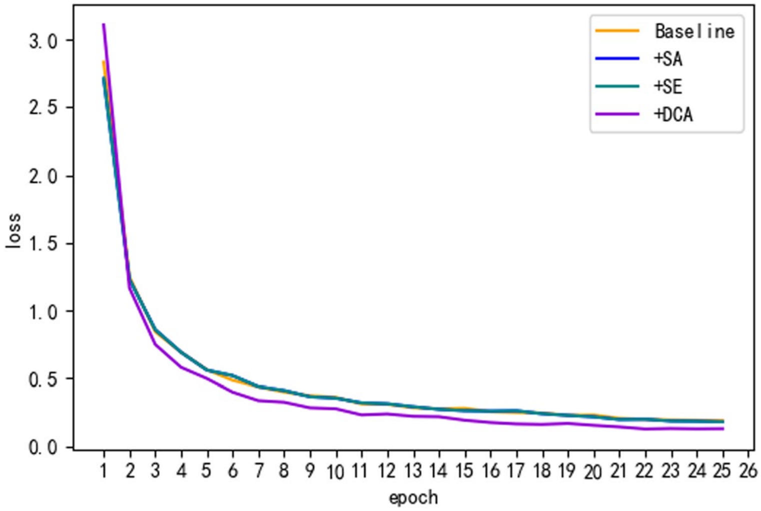

Fig. 7 Loss curve with the number of training epoch. On the basis of the baseline model, soft attention (SA), shared encoder (SE), and deep cross attention (DCA) are added successively

Table 2 Test the effect of noise on model performance

	SNR	Seen			Unseen		
		EER (%)	minDCF	Acc (%)	EER (%)	minDCF	Acc (%)
Noise	− 5 dB	4.11	0.193	90.72	8.07	0.406	87.85
	0 dB	1.34	0.947	94.70	4.92	0.266	91.95
	5 dB	0.63	0.037	96.49	3.68	0.206	93.75
	10 dB	0.27	0.019	97.45	2.81	0.175	94.65
	15 dB	0.20	0.014	97.85	2.44	0.158	95.17
Music	− 5 dB	5.13	0.251	89.83	9.54	0.453	86.97
	0 dB	1.49	0.082	94.68	5.14	0.266	91.75
	5 dB	0.53	0.032	96.68	3.55	0.201	94.03
	10 dB	0.26	0.014	97.76	2.53	0.162	94.87
	15 dB	0.17	0.007	98.04	2.24	0.143	95.35

without noise. In terms of noise categories, the influence of musical noise is found to be more pronounced than environmental noise.

5.2.2 *Performance of model under near-and far-field conditions*

The results are presented in Table 3. For near-field speech, the distance between the sound source and the microphone is 0.1m; for far-field speech, the distance between the sound source and the microphone is 1 m, 3 m, and 5 m. The test set with noise exhibits an SNR ranging from − 5 dB to 15 dB. The far and near field EER values for unseen speakers are comparable to minDCF in the noiseless test set, although Acc near field speech is higher. In the case of unknown speakers, the test set with noise reveals that near-field speech maintains a higher Acc and lower EER and minDCF, while far-field speech displays significantly higher EER and minDCF values than near-field speech. Additionally, Acc decreases as the distance increases. The superior performance of near-field speech may be attributed to the larger number of near-field speech samples in the constructed dataset. In conclusion, the distance between the sound source and the microphone has a tolerable impact on the model performance, and the performance of far-field speech remains relatively consistent within the 1 m to 5 m.

5.2.3 *Performance of model under different speech speed*

The results are presented in Table 4, where we categorize speech speed into normal and fast speed. The test set comprises both near-field and far-field speech, with the SNR range of noisy speech ranging from − 5 dB to 15 dB. For unknown speakers in noise-free conditions, speech speed has minimal impact on speaker verification performance; however, fast speech lowers the accuracy of keyword recognition by 2.55%. In the case of unknown speakers in noisy environments, fast speech results in a 0.93% increase in EER, a 0.024 increase in minDCF, and a 2.36% decrease in overall Acc. Consequently, fast speech speed has a more pronounced effect on keyword detection, while its impact on speaker validation models is relatively smaller.

Table 3 Test the effect of distance on model performance

	Distance	Seen			Unseen		
		EER (%)	minDCF	Acc (%)	EER (%)	minDCF	Acc (%)
Without noise	0.1 m	0.03	0.004	98.73	2.41	0.186	96.40
	1 m	0.09	0.006	98.39	2.33	0.186	94.97
	3 m	0.18	0.013	97.74	2.23	0.146	94.56
	5 m	0.08	0.010	97.81	2.65	0.134	94.24
With noise	0.1 m	0.96	0.046	97.20	2.30	0.123	95.15
	1 m	1.53	0.072	95.29	5.07	0.319	92.25
	3 m	2.16	0.072	94.95	4.85	0.283	91.06
	5 m	2.14	0.094	94.56	5.13	0.298	90.63

Table 4 Test the effect of speech speed on model performance

	Speed	Seen			Unseen		
		EER (%)	minDCF	Acc (%)	EER (%)	minDCF	Acc (%)
Without noise	Normal	0.03	0.004	99.00	1.36	0.093	96.87
	Fast	0.13	0.002	97.69	1.49	0.096	94.32
With noise	Normal	0.76	0.038	96.49	3.37	0.180	94.03
	Fast	1.16	0.055	94.83	4.30	0.216	91.67

6 Conclusion

In conclusion, this paper presents a novel multi-tasking model that jointly addresses speaker verification (SV) and keyword spotting (KWS) tasks. By utilizing a shared encoder (SE) to extract FBank features, our approach efficiently feeds the derived data into speaker verification and keyword spotting branches. We employ soft attention (SA) connections as an alternative to residual connections, which maintain the model's depth and amplify its focus on relevant features. Furthermore, we incorporate a deep cross-attention (DCA) module that effectively fuses the two models with linear complexity. Our experimental results show that this model is better than the baseline model, with EER, minDCF, and Acc increasing by 0.18%, 0.014, and 2.47% respectively, in the unseen noisy test set. Future research will concentrate on minimizing the computational complexity and model parameters and exploring more efficient connection strategies.

Abbreviations

SV	Speaker verification
KWS	Keyword spotting
DCA	Deep cross-attention
EER	Equal error rate
minDCF	Minimum detection cost function
ECAPA-TDNN	Emphasized channel attention, propagation, and aggregation in time delay neural network
MTCANet	Multi-task deep cross-attention network
DNN	Deep neural network
NLP	Natural language processing
ASR	Automatic speech recognition
LSTM	Long short-term memory
STFT	Short-time Fourier transform
MFCC	Mel-frequency cepstral coeffients
PLDA	Probabilistic linear discriminant analysis
CTC	Connectionist temporal classification
SE	Shared encoder
SA	Soft attention
DTW	Dynamic time warping
GMM	Gaussian mixture model
HMM	Hidden Markov model
TCN	Temporal convolutional network
TDNN	Time delay neural network
UBM	Universal background model
AAM-Softmax	Additive angular margin loss
TP	True positive
FN	False negative
FP	False positive
TN	True negative
FAR	False acceptance rate
FRR	False rejection rate
ACC	Accuracy
SNR	Signal-to-noise ratios

Acknowledgements
Not applicable.

Author' contributions
All authors listed have contributed significantly to this work. The authors read and approved the final manuscript.

Authors' information
Not applicable.

Funding
This work was partially supported by the National Natural Science Foundation of China (62006062, 62176076), Shenzhen Foundational Research Funding JCYJ20210324115614039.

Availability of data and materials
The data that support the finding of this study are not publicly available. Data are however available from the authors upon reasonable request and with permission of Konka Corporation.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 3 May 2023 Accepted: 19 June 2023
Published online: 01 July 2023

References

1. G. Chen, C. Parada, G. Heigold, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Small-footprint keyword spotting using deep neural networks (IEEE, Florence, Italy, 2014), pp. 4087–4091
2. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
3. M. Xu, X.L. Zhang, in *Interspeech*. Depthwise separable convolutional resnet with squeeze-and-excitation blocks for small-footprint keyword spotting (ISCA, Shanghai, China, 2020), pp. 2547–2551
4. M. Yu, X. Ji, Y. Gao, L. Chen, J. Chen, J. Zheng, D. Su, D. Yu, in *Interspeech*. Text-dependent speech enhancement for small-footprint robust keyword detection. (2018), pp. 2613–2617
5. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
6. Y. Huang, T.Z. Shabestary, A. Gruenstein, L. Wan, in *Interspeech*. Multi-microphone adaptive noise cancellation for robust hotword detection (ISCA, Graz, Austria, 2019), pp. 1233–1237

7. D. Ng, Y. Chen, B. Tian, Q. Fu, E.S. Chng, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Convmixer: feature interactive convolution with curriculum learning for small footprint and noisy far-field keyword spotting (2022), pp. 3603–3607
8. A. Gomez-Alanis, J.A. Gonzalez-Lopez, S.P. Dubagunta, A.M. Peinado, M.M. Doss, On joint optimization of automatic speaker verification and anti-spoofing in the embedding space. IEEE Trans. Inf. Forensic. Secur. **16**, 1579–1593 (2020)
9. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. X-vectors: robust DNN embeddings for speaker recognition (2018), pp. 5329–5333
10. E. Variani, X. Lei, E. McDermott, I.L. Moreno, J. Gonzalez-Dominguez, in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Deep neural networks for small footprint text-dependent speaker verification (2014), pp. 4052–4056
11. W. Cai, J. Chen, J. Zhang, M. Li, On-the-fly data loader and utterance-level aggregation for speaker and language recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 1038–1051 (2020)
12. J. Zhou, T. Jiang, Z. Li, L. Li, Q. Hong, in *Interspeech*. Deep speaker embedding extraction with channel-wise feature responses and additive supervision softmax loss function. (ISCA, Graz, Austria, 2019), pp. 2883–2887
13. S. Wang, J. Rohdin, O. Plchot, L. Burget, K. Yu, J. Černockỳ, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Investigation of specaugment for deep speaker embedding learning (IEEE, Barcelona, Spain, 2020), pp. 7139–7143
14. B. Gu, W. Guo, L. Dai, J. Du, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. An improved deep neural network for modeling speaker characteristics at different temporal scales (IEEE, Barcelona, Spain, 2020), pp. 6814–6818
15. W. Lin, M.W. Mak, Mixture representation learning for deep speaker embedding. IEEE/ACM Trans. Audio Speech Lang. Process. **30**, 968–978 (2022)
16. D. Garcia-Romero, C.Y. Espy-Wilson, in *Twelfth annual conference of the international speech communication association*. Analysis of i-vector length normalization in speaker recognition systems (ISCA, Florence, Italy, 2011)
17. Z. Bai, X.L. Zhang, J. Chen, Cosine metric learning based speaker verification. Speech Commun. **118**, 10–20 (2020)
18. Z. Bai, X.L. Zhang, J. Chen, Speaker verification by partial auc optimization with mahalanobis distance metric learning. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 1533–1548 (2020)
19. N. Brümmer, E. De Villiers, The bosaris toolkit: Theory, algorithms and code for surviving the new dcf. arXiv preprint arXiv:1304.2865 (2013)
20. L. Ferrer, M.K. Nandwana, M. McLaren, D. Castan, A. Lawson, Toward fail-safe speaker recognition: trial-based calibration with a reject option. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(1), 140–153 (2018)
21. H.S. Lee, Y. Tso, Y.F. Chang, H.M. Wang, S.K. Jeng, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Speaker verification using kernel-based binary classifiers with binary operation derived features (IEEE, Florence, Italy, 2014), pp. 1660–1664
22. W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, in *IEEE conference on computer vision and pattern recognition (CVPR)*. Sphereface: Deep hypersphere embedding for face recognition (IEEE, Honolulu, HI, USA, 2017), pp. 212–220
23. Y. Li, F. Gao, Z. Ou, J. Sun, in *11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. Angular softmax loss for end-to-end speaker verification (ISCA, Taibei, Taiwan, 2018), pp. 190–194
24. Z. Huang, S. Wang, K. Yu, in *Interspeech*. Angular softmax for short-duration text-independent speaker verification (ISCA, Hyderabad, India, 2018), pp. 3623–3627
25. S. Wang, Z. Huang, Y. Qian, K. Yu, Discriminative neural embedding learning for short-duration text-independent speaker verification. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(11), 1686–1696 (2019)
26. F. Wang, J. Cheng, W. Liu, H. Liu, Additive margin softmax for face verification. IEEE Signal Process. Lett. **25**(7), 926–930 (2018)
27. J. Deng, J. Guo, N. Xue, S. Zafeiriou, in *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. Arcface: additive angular margin loss for deep face recognition (IEEE, Long Beach, CA, USA, 2019), pp. 4690–4699
28. B. Desplanques, J. Thienpondt, K. Demuynck, in *Interspeech*. ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification (ISCA, Shanghai, China, 2020), pp. 3830–3834
29. S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, J. Bridle, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Multi-task learning for speaker verification and voice trigger detection (IEEE, Barcelona, Spain, 2020), pp. 6844–6848
30. S. Yang, B. Kim, I. Chung, S. Chang, in *Interspeech*, Personalized keyword spotting through multi-task learning (2022), pp. 1881–1885
31. T. Sainath, C. Parada, Convolutional neural networks for small-footprint keyword spotting (2015)
32. M. Sun, D. Snyder, Y. Gao, V. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Ström, S. Matsoukas, S. Vitaladevuni, Compressed time delay neural network for small-footprint keyword spotting (ISCA, Stockholm, Sweden, 2017)
33. C. Shan, J. Zhang, Y. Wang, L. Xie, in *Interspeech*. Attention-based end-to-end models for small-footprint keyword spotting (ISCA, Hyderabad, India, 2018), pp. 2037–2041
34. S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, S. Ha, in *Interspeech*. Temporal convolution for real-time keyword spotting on mobile devices (ISCA, Graz, Austria, 2019), pp. 3372–3376
35. S.S. Jagtap, D. Bhalke, in *International Conference on Pervasive Computing (ICPC)*. Speaker verification using gaussian mixture model (IEEE, Pune, India, 2015), pp. 1–5
36. 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**(4), 788–798 (2010)
37. V. Peddinti, D. Povey, S. Khudanpur, in *Sixteenth annual conference of the international speech communication association*. A time delay neural network architecture for efficient modeling of long temporal contexts (ISCA, Dresden, Germany, 2015)
38. J. Hu, L. Shen, G. Sun, in *IEEE conference on computer vision and pattern recognition (CVPR)*. Squeeze-and-excitation networks (IEEE, Salt Lake City, Utah, USA, 2018), pp. 7132–7141
39. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
40. X. Wang, S. Sun, C. Shan, J. Hou, L. Xie, S. Li, X. Lei, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting (IEEE, Brighton, UK, 2019), pp. 6366–6370
41. S. Wang, B.Z. Li, M. Khabsa, H. Fang, H. Ma, Linformer: self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020)
42. K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, et al., Rethinking attention with performers. arXiv preprint arXiv:2009.14794 (2020)
43. Y. Xiong, Z. Zeng, R. Chakraborty, M. Tan, G. Fung, Y. Li, V. Singh, in *AAAI Conference on Artificial Intelligence*. Nyströmformer: a nyström-based algorithm for approximating self-attention, vol. 35 (AAAI, Vancouver, CA, 2021), pp. 14138–14148
44. Z. Huang, X. Shi, C. Zhang, Q. Wang, K.C. Cheung, H. Qin, J. Dai, H. Li, in *Computer Vision-ECCV 2022: 17th European Conference*. Flowformer: a transformer architecture for optical flow (Springer, Tel Aviv, Israel, 2022), pp.668–685
45. H. Wu, J. Wu, J. Xu, J. Wang, M. Long, Flowformer: linearizing transformers with conservation flows. arXiv preprint arXiv:2202.06258 (2022)
46. R. Caruana, Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
47. X. Liang, Y. Zou, X. Zhuang, J. Yang, T. Niu, R. Xu, Mmateric: Multi-task learning and multi-fusion for audiotext emotion recognition in conversation. Electronics **12**(7), 1534 (2023)
48. S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, J. Cernockỳ, in *Interspeech*. On the usage of phonetic information for text-independent speaker embedding extraction. (ISCA, Graz, Austria, 2019), pp. 1148–1152
49. M. Zhao, R. Li, S. Yan, Z. Li, H. Lu, S. Xia, Q. Hong, L. Li, in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Phone-aware multi-task learning and length expanding for short-duration language recognition (IEEE, Lanzhou, China, 2019), pp. 433–437
50. M. Jung, Y. Jung, J. Goo, H. Kim, in *Interspeech*. Multi-task network for noise-robust keyword spotting and speaker verification using CTC-based soft VAD and global query attention (ISCA, Shanghai, China, 2020), pp. 931–935

51. S. Pascual, A. Bonafonte, S. Joan, An end-to-end speech recognition system based on deep neural networks. arXiv preprint arXiv:1703.09452 (2017)
52. Y. Xia, T. He, X. Tan, F. Tian, D. He, T. Qin, in *AAAI conference on artificial intelligence*. Tied transformers: neural machine translation with shared encoder and decoder, vol. 33 (AAAI, Honolulu, Hawaii, USA, 2019), pp. 5466–5473
53. Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, X. Lei, in *Interspeech*. Wenet: production oriented streaming and non-streaming end-to-end speech recognition toolkit (ISCA, Brno, Czech, 2021), pp. 4054–4058
54. Y. Sun, X. Wang, X. Tang, in *IEEE conference on computer vision and pattern recognition (CVPR)*. Deep learning face representation from predicting 10,000 classes (2014), pp. 1891–1898
55. Y. Taigman, M. Yang, M. Ranzato, L. Wolf, in *IEEE conference on computer vision and pattern recognition (CVPR)*. Deepface: closing the gap to human-level performance in face verification (IEEE, Columbus, Ohio, USA, 2014), pp. 1701–1708
56. D. Reynolds, E. Singer, S.O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, L. Mason, J. Hernandez-Cordero, *The 2016 nist speaker recognition evaluation*. Technical report, MIT Lincoln Laboratory Lexington United States (ISCA, Stockholm, Sweden, 2017)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.