ESRNN: Effective Residual Self-Attention Recurrent Neural Network for Sound Event Location

Bin Zhang

1Ocean University of China

October 30, 2023

Abstract

Sound event location is a momentous subtask of two-dimensional direction-of-arrival (2D-DOA) estimation to forecast the azimuth and elevation angle from each active sound event class of an audio fragment using multi-label regression in the 3D Cartesian coordinates. The main problem with the traditional multi-signal classification (MUSIC) algorithm and the existing baseline convolution recurrent neural network (BCRNN) are lower precision and huge computation in weaker signal-noise-ratio (SNR) environment. In particular, MUSIC algorithm will cause complete distortion when the SNR is lower than -5 dB. We thus design effective residual self-attention recurrent neural network (ESRNN) not only to overcome the distortion of traditional MUSIC algorithm under lower SNR but also further reduce predicted 2D-DOA error in different SNR reverberation environments. Two different filter structures, ESRNN-L and ESRNN-G, are designed to improve the SNR of our model when the SNR is above 0 dB or below -5 dB in a targeted manner respectively. To verify the efficiency and robustness, we train and test our model using TAU Spatial Sound Events 2019 datasets extracted the phase and magnitude spectrogram with different synthetic SNRs ranging from -10 dB to 30 dB. The experimental results show that the optimal 2D-DOA error of our ESRNN-L is 21% lower than BCRNN when the SNR is less than -5 dB, and ESRNN-G is 15% error lower with 10% parameters reduction when the SNR is higher than 0 dB.
ESRNN: Effective Residual Self-Attention Recurrent Neural Network for Sound Event Location

Bin Zhang, Jiawen He, Peishun Liu, Liang Wang, Hao Zhou and Ningxue Wang, Ruichun Tang Staff, IEEE.

Abstract—Sound event location is a momentous subtask of two-dimensional direction-of-arrival (2D-DOA) estimation to forecast the azimuth and elevation angle from each active sound event class of an audio fragment using multi-label regression in the 3D Cartesian coordinates. The main problem with the traditional multi-signal classification (MUSIC) algorithm and the existing baseline convolution recurrent neural network (BCRNN) are lower precision and huge computation in weaker signal-noise-ratio (SNR) environment. In particular, MUSIC algorithm will cause complete distortion when the SNR is lower than -5 dB. We thus design effective residual self-attention recurrent neural network (ESRNN) not only to overcome the distortion of traditional MUSIC algorithm under lower SNR but also further reduce predicted 2D-DOA error in different SNR reverberation environments. Two different filter structures, ESRNN-L and ESRNN-G, are designed to improve the SNR of our model when the SNR is above 0 dB or below -5 dB in a targeted manner respectively. To verify the efficiency and robustness, we train and test our model using TAU Spatial Sound Events 2019 datasets extracted the phase and magnitude spectrogram with different synthetic SNRs ranging from -10 dB to 30 dB. The experimental results show that the optimal 2D-DOA error of our ESRNN-L is 21% lower than BCRNN when the SNR is less than -5 dB, and ESRNN-G is 15% error lower with 10% parameters reduction when the SNR is higher than 0 dB.

Index Terms—Sound event location, low signal-noise-ratio, 2-D DOA, residual self-attention, recurrent neural network.

I. INTRODUCTION

SOUND event location, which is a fundamental multi-labels task needs to estimate the occurrence time angle along with one or more corresponding spatial locus multi-channel inputs. It is a theoretical basic research area for track indicates event in different acoustic scenes especially in spatial room impulse responses (SRIRs), realistic spatialization and reverberation environments. Sound event location is a submission DOA which including 1D or 2D angles for each active sound event class calculated from the coordinates using the estimation of parameter or time difference of arrival method. The task of sound event location can be applied to many areas such as machine automatic driver [1]–[4], smart automation [5]–[8], ocean positioning technology [9]–[11], quality monitoring [12], medical navigation [13], navigation without visual input or with occluded targets [14], self-localization [15], smart-phone audio assistant [16], acoustic device monitoring [17], scene visualization systems [18], moving source tracking [19], Virtual Reality [20] and etc. The processing flow and applications can be shown in Fig. 1. The most challenged portion of this mission is to locate the sound source with one or more overlapping sound events from rooms which mixed with different spatial ambient noise recorded or collected in real spatial sound scene recordings.

The main idea of DOA estimation is able to be summed up as four categories that are Beam Forming, Space Spectral Estimation, Time Difference of Arrival (TDOA) and Deep Learning methods. DOA estimation includes classification and regression two ways to handle discrete and continuous angle values. With the artificial intelligence developed rapidly, many traditional methods are replaced with the deep learning method which has the ability to extract features automatically for its robustness and adaptability. Butko et al. [21] leverage Hidden Markov Model and Gaussian Mixture Model (HMM-GMM) classifier to recognize fourteen different two overlapping source sound event which adopts distributed microphone array and use Steered-Response Power (SRP) method to locate the source in a meeting room. Chakraborty et al. [22] employ a sound field modeling to replace for SRP which eases the problem of label association. As deep learning technology continues to mature, Adavanne et al. [9] propose that utilize recurrent neural networks which using the azimuth and elevation angle output the space structure information by the way of regression to trace and detect overlapping sound events. Wu et al. [23] based on a spectra recovery algorithm designing an efficient deep convolution network and apply that electromagnetism

Fig. 1. The overall processing flow and applications of sound event location.
DOA estimation which converts the DOA estimation problem into a sparse linear inverse problem by introducing spatially overcomplete formulation. Tang et al. [24] present a novel learning-based approach to estimate the DOA as a regression problem which design a recurrent neural network trained by four-channel FOA format data [25]. Gradually, Adavanne et al. [26] propose a multi-space reverberation overlapping sound database which contains real-life encouraged sound events and adds real ambient sounds with different sign-to-noise ratios to the recording for simulating realistic scenarios based on FOA datasets. Nguyen et al. [27] design a 2D convolutional neural network (CNN) to estimate the number of sources and DOA from a short-time spatial pseudo-spectra which solves the prior necessity of known the number of sources with multi-task processing unseen conditions. It is the first time that spectra pseudo-spectra are used as the input features for DOA processing in different noise, reverberation and multi-overlapping sound event environments. The approach they proposed can predict the predict up to 4 overlapping sound events which offer a new novel thought for DOA estimation using spatial pseudo-spectrum algorithm as a preconditioning. Liu et al. [28] design a deep neural network that realizes a higher resolution which could better generalization to random signal number and SNR. Papageorgiou et al. [29] introduce a convolutional neural network which is trained from multi-channel datasets in low SNR environments and can predict angular directions across different SNR using the sample covariance estimate as a task of classification. Slizovskaia et al. [30] propose a conditioned network including combination of data augmentation with pre-trained class-specific embedding focusing on sound event DOA estimation which provide a fresh thought to design feature-wise modulation layers for locating.

Our work is based on the aforementioned research, with the main purpose of overcoming interference in noisy environments with different SNRs for multiple overlapping sound events, and improving the super-resolution and anti-interference performance of 2D-DOA estimation. We design a novel network with a local or deep global residual attention block for sound event location. The core idea is to exploit the residual information fused with soft thresholding to improve the performance of sound event location task in different noise environments. We introduce a residual-based network which integrate deep residual block with BCRNN and optimize the merged network, we called that effective residual self-attention deep recurrent neural network. We also prove and visual the details of the new module structure why it can improve the performance of sound event location. The main contributions of this paper can be summarized as follows:

1) We converge a novel residual self-attention module using the deep recurrent network, named ESRNN, based on supervision for 2D-DOA estimated to be more suitable for the sound event location.

2) To better improve the 2D-DOA estimation accuracy of ESRNN, we design ESRNN-L and ESRNN-G individually which have excellent generalization performance in different scenarios to enhance the local and global attention capture ability.

3) We sufficiently compared ESRNN-L and ESRNN-G with the BCRNN and the conventional MUSIC algorithm on TUN2019 datasets with synthetic difference SNR datasets which have lower loss and robust performance.

The rest of this paper is shown below: Section II mainly introduces the direction processing of the traditional MUSIC algorithm on DOA (1D and 2D) estimation. Section III mainly describes the structure and data processing of the two models of ESRNN-L and ESRNN-G proposed in details. Section IV introduces how we synthesized audio with different SNR ratios for different reverb audios channels and demonstrated the results of experimental comparative analysis. The final Section V, we provide a summary of our work and an outlook for future work.

II. RELATED WORK

As a classical algorithm for spectrum estimation, the main idea of MUSIC algorithm [31] is to decompose the observation space of incoherent source signals into signal subspace and noise subspace. Music algorithm decomposes its matrix eigenspace by using the orthogonality of its feature space and find the angle corresponding to its peak by constructing a spatial spectral function, which is the estimation of the direction angle of the incoming wave. A linear array with \( M \) elements, at the array at the incidence angle \( \theta \) to receive the incident narrowband signal, the spacing between the elements is \( d \), the signal received by the elements noting as \( Y_i(t) \), \( i \in [1, M] \), representing the signal received by the \( i \) th element at time \( t \). Then the time difference between adjacent elements receiving signals is \( \Delta t = (d \cdot \sin \theta)/c \), \( c \) representing the speed of light and the phase difference between the adjacent two elements is \( \nabla \phi = (2\pi f d \cdot \sin \theta)/\sigma \), where \( \sigma \) represents the wavelength and \( f \) represents the frequency of the signal. The phase difference of the received signal for the entire array is denoted as \( A \) which represents its guide vector. The expression of \( A \) is denoted as follows:

\[
A(\theta) = \begin{bmatrix}
1, e^{\frac{2\pi d \sin \theta}{\sigma}}, e^{\frac{2\pi 2d \sin \theta}{\sigma}}, \ldots, e^{\frac{2\pi (M-1)d \sin \theta}{\sigma}}
\end{bmatrix}
\]

Then the matrix of the received signal of the entire array is denoted as \( Y(t) = A(\theta) \cdot S(t) + N(t) \), where \( S(t) \) represents...
the signal source matrix and \( N(t) \) represents the noise matrix at the corresponding moment. Take single source signal receiving array of uniform linear array (ULA) as an example is shown in Fig. 2:

Next, the received matrix of many transmitted sources can be expressed as \( Y(t) = A(\theta_D) \cdot S_D(t) + N_D(t) \), where \( A = [a(\theta_1), a(\theta_2), \ldots, a(\theta_D)] \), where \( A(\theta_D) \) is the \( M \times D \) dimensional guide vector, \( S_D(t) \) meaning the \( D \times 1 \) dimensional signal source at time \( t \), \( N_D(t) \) representing the noise signal matrix at time \( t \). The covariance matrix of the discrete signal \( Y(t) \) can be obtained from the covariance formula:

\[
R_y = E [Y(t) \cdot Y^H(t)]
\]

where \( R_y = AR_sA^H + R_N \), where \( R_s \) represents the covariance matrix of the signal source, \( R_N \) a covariance matrix representing the noise subspace, and the \( R_y \) decomposition can be obtained by decomposition of eigenvalues:

\[
R_y = U_S \cdot \sum_S U_S^H + U_N \cdot \sum_N U_N^H
\]

where \( \sum_S \) and \( \sum_N \) represent the eigenvalues of the signal subspace and the noise subspace, respectively, and the \( U_S \) and \( U_N \) represent their eigenvectors according to the incoherence between the transmitted signals and orthogonal to the noise signal, then their orthogonality can be used to construct the spatial spectral function:

\[
P(\theta_i) = \frac{1}{a(\theta_i)^H E_n E_n^H a(\theta)}
\]

among them \( E_n \) represents the noise subspace matrix, \( H \) represents the conjugate transpose matrix, and the MUSIC algorithm estimates the wave direction of the incoming wave by searching for the peak of the spatial spectral function corresponding to the angle of arrival.

The MUSIC algorithm uses the wave path difference and time difference between the far-field signal to reach different array elements for passive localization of the incoming wave direction of finite snapshots in a short period of time, and has high resolution under the ideal conditions of incoherent sources and known number of sources.

Compared with 1D-DOA estimation predicting azimuth angle, 2D-DOA [32] estimation predicts the azimuth and elevation angle of the incoming wave of the incident signal at the same time. 1D-DOA usually uses ULA, while 2D-DOA estimation is usually based on uniform planar array (UPA).
B. ESRNN Architecture

1) Effective Residual Self-Attention Recurrent Neural Network: The ESRNN architecture uses the characteristic of 2D convolutional neural network layer first to extract the local shift-invariant features from phase and magnitude spectrum as shown in Fig. 4. Convolution kernels have fewer parameters to capture the local features which benefit from its local perception and parameter sharing. The expression of feature extraction between a convolution kernel and an input production can be described as follows:

\[ y_l = \sum_{i \in M_l} x_i \ast k_{il} + b_l \]  \hspace{1cm} (7)

where \( x_i \) is \( i \)th channel of the input feature map, \( y_l \) is the \( l \)th of the input feature map, \( k \) represents the kernel, \( M_l \) means the contact to all channels for each \( l \)th sub-channel feature map output and \( b_l \) is the bias. Then, \( K \) filters CNN layers that each one has \( 3 \times 3 \times 2C \) dimension with the rectified linear unit activation. Next, using batch normalization (BN), activations and max pooling (MP) layer which will allow the network keeping reducing the dimension but maintain the local feature capacity along the frequency axis of \( T \) frames. This will output \( T \times 4 \times K \) dimension features where 4 is a result of MP layers (details see IV.D.2) Model Details Configuration). The BN accurately the speed of training layer and better converging the module. The BN formulation can be seen as follows:

\[ x = \frac{x_i - \mu}{\sigma^2 + \epsilon} \]  \hspace{1cm} (8)

where \( \mu \) and \( \sigma^2 \) denote the mean and variance. The Relu function is used to change the non-linear relation between the input and output data in every layer. The MP layer will retain the most important features because of the characteristic of its invariance in local patches. We called the joint layer named conv-mp-block which include 2D convolution kernel, BN, Relu activation function and MP layer sequentially. Finally, the number of \( P \) nodes of gated recurrent units (GRU) followed \( Q \) fully connected (FC) layers to discriminate the difference with 2D-DOA coordinate. After the dimension of \( T \times P \) output from \( T \) frames of \( P \) GRU nodes, the final result will contain \( T \times Q \) dimension from FC layer which enjoys parameters between time stages with linear activation.

Thanks to the conv-mp-block, we can easily capture the significant feature representation of spectrogram local information. But we notice that some noise signal will be reserved. This is not a good idea for network designed to locate or detect the key information because those near-zero noise numbers will decrease the SNR after the forward or backward circular propagation and finally increase the generalization error for the whole network. To decrease the power of noise signal, we redesign an efficient network which can easily transform the useful information to discriminate the noise or key information to near-zero features which will increase the feature resolution ability for different senses to enhance the perception of our network. The soft thresholding value can be seen a nonlinear transformation layer based on residual block structure which is used to remove noise-related features. The soft thresholding value module build started with global average pooling (GAP), which is used to absolute the entered feature map \( x \) to 1-D vector. Then, two FC layers are followed to scale the entered features to the range of \((0,1)\). The final thresholding value will be calculated as follows:

\[ \tau_c = \frac{1}{1 + e^{-z_c}} \cdot \text{average} |x_c| \]  \hspace{1cm} (9)

where \( \tau_c \) is the feature map thresholding value of the \( c \) channel and \( z_c \) is the feature at the \( c \)th neuron. This soft thresholding value is viewed as a residual block merged into conv-mp-block(s). The theoretical basis function of the soft thresholding as follows:

\[ y = \begin{cases}  
  x - \tau, & x > \tau \\
  0, & -\tau \leq x \leq \tau \\
  x + \tau, & x < -\tau 
\end{cases} \]  \hspace{1cm} (10)

where \( x \) is the input feature, \( y \) is the output feature, \( \tau \) is threshold respectively. If not set \( \tau \), the output \( y \) maybe preserve near-zero values which are defined noise features. If set \( \tau \), the output \( y \) will filter noise-related features if \( \tau \) is designed properly. We designed two ways to reduce different near-zero features in different channels of the input feature.

2) Local Focus Attention Module: The overall architecture network can be seen in Fig. 4. The first method we design that captures different near-zero features in local channels before they merged into the GRU module, the details can be seen as Fig. 5, called ESRNN-L. This method can filter out noise layer by layer from the bottom floor. Firstly, the model performs temporal convolution calculation on the features of the activated sound events of the input. Then integrates the self-attention mechanism [33] after the convolution of the local conv-mp-block, which exerts the powerful ability to model the correlation relationship between the elements in the input sequence. Further pays attention to the information from different representation subspaces and improves the model’s ability to filter local noise. Since the attention model performs three linear mappings of the input sequences to obtain \( Q, K, V \), using the weights calculated by the dot product of \( Q \) and \( K \) weighted \( V \). The core formula of key-value attention is as follows:
The design of the ESRNN-L improves the cross-dimensional interaction, ESRNN-G effectively reduces the aim of retaining global information to amplify global recursive network features as a global context vector. With G use deep full local feature information extracted by shallow network, called ESRNN-G, which shown in Fig. 7. ESRNN-deeper. Be inspired by this, we design another architecture the global difference can be detected when the layer got greatly increase the complexity of the model. We find that only reduce the convergence speed of the model but also information for high-level input. The amount of parameters extracted locally which retains more key low-level semanticing of the network depth, ESRNN-L recursive loop features are retains the key local speech feature information and improves residual function structure to be learned. ESRNN-L better

\[ \text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \]  

The input of self-attention encoding is the eigenvector I output of conv-mp-block. The local sparse encoding Q, K, V is obtained after linear mapping of the parameter matrix \(W^q, W^k\) and \(W^v\). According to the inner product \(QK^T\) similarity calculation of the vector, the output of the similarity matrix is normalized by \(\sqrt{d_k}\) distance. The weighted attention encoding is output by the softmax function and then the weighted encoded matrix vector of the output is obtained after the inner product of V. Its self-attention process is shown in Fig. 6.

We apply equation (10) to the locally weighted mixed vector filtering at the decoding end of the self-attention sparse encoding to retain more non-noisy feature vectors. These results are mapped by the GRU gating unit on the time series convolution filtering to obtain the final output value. In order to avoid the disappearance of gradients caused by too deep local network depth, the residual module is added to introduce the identity mapping of the output of the local network layer. Then adjust the optimal goal of the current layer and change it to learn the residual of input and output. The residual identity mapping [34] formula is as follows:

\[ H(x) = F(x) + x \]  

Next, two common GRU layers followed by the output of the previous layers which is a gated loop unit to effectively solve long-term dependency problems with fewer parameters and faster speed in our scenarios. Then, one FC layer map the learned distributed signal feature representation of the role of the sample marker space. Finally, after the linear activation output the final 2D-DOA estimation matrix. For better compare the parameter numbers of the design of the ESRNN-L and ESRNN-G. We visual every layer and analysis the relationship between adjacent layers and different blocks. The model layers construction, params and layer output shape details of ESRNN-L and ESRNN-G can be observed in Table I and Table II respectively.

### C. Training Processing

The loss for 2D-DOA estimation of ESRNN we use mean square error (MSE) which \(x, y\) and \(z\) represent the active sound events location of a series frames in 3D Cartesian coordinates. We set the \(x = 0, y = 0\) and \(z = 0\) to discriminate the inactive sound event. In 3D space, we use the Euclidean metric which calculate the distance of two points such as two points \((x_1, y_1, z_1)\) and \((x_2, y_2, z_2)\) are given, the distance between two points is given by \(\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}\). The same points are given by \((x_1 - x_2)^2 + (y_1 - y_2)^2 + (y_1 - y_2)^2)/3\) so that if the estimated position is more close to the true position, the distance will be more reduced. The Cartesian coordinate values of the 2D-DOA regression estimation can better learn the continuity of the network which will better enhance the ability of the network compared with discrete azimuth and elevation values. Because the adaptive learning ability to the neural network can obtain more detailed continuous resolution angle than the rough discrete interval angle.

Further experiments on this are discussed in the next Section. The ESRNN is implemented using Keras and TensorFlow as backend. We train the ESRNN with a weighted combination of MSE loss for 50 epochs using Adam optimizer with default parameters as used in the baseline [9]. We steer the network

---

**Fig. 6.** Self-Attention convolutional calculation processing.

**Fig. 7.** ESRNN-G architecture; ESRNN-G focus more on global near-zero features so that followed after the local feature extraction after all conv-mp-blocks. This design has more strong power to filter out the noise features because the difference will more obviously after the convolution by convolution.
for better convergence using early stopping when the network training loss is not declined than the minimum loss value in order to prevent the network over-fitting.

IV. EXPERIMENTS

A. Deploy Running Environment

The operating system of our experiment environment is Centos7 3.10.0-1160.6.1.el7.x86_64 and Windows11_21H2 which is used to train and evaluate the model on Intel(R) Xeon(R) Platinum 8124M 3.00GHz CPU and 12th Gen Intel(R) Core(TM) i7-12700H 2.30GHz CPU respectively.

B. Datasets

The dataset of our model we used that is TAU Spatial Sound Events 2019. This dataset contains two different sub-datasets including at most two overlapping events which are Ambisonic and Microphone Array coming from an identical scene. The mainly different is that Ambisonic offers four-channel FOA format recordings and Microphone Array offers four-channel MIC format recordings from a tetrahedral array configuration. These recordings are synthesized from the spatial room impulse response (IRs) which collected from five indoor locations.

TAU Spatial Sound Events 2019–Microphone Array dataset use the channels 6,10,22 and 26 respectively coming from the number of microphones in Eigenmike (32 channels) which the number of channels in IRs is equal to that. The channels of Microphone Array dataset that correspond to the position of \((\pm 45, \pm 35)\) and \((\pm 135, \mp 35)\), the distance equaling 42cm of the above positions. Compared with the MIC format recordings, Ambisonic provide FOA format which converts 32 channel microphone signals by encoding filters based on anechoic measurements of Eigenmike (32 channels) array response. Followed by equation(3), equation(4) and the characteristic of FOA format, the steering vector of the FOA is given by azimuth angle \(\varphi\) and elevation angle \(\theta\):

\[
A(\vartheta, \varphi) = \begin{pmatrix}
1 \\
\sqrt{3}\cos\theta\sin\varphi \\
\frac{\sqrt{3}\sin\theta}{\sqrt{3}\cos\varphi}
\end{pmatrix}
\] (13)
The development dataset contains of 400 recordings each format which includes about one minute long sampled at 48000Hz and the average SNR of each sound event is about 30 dB. For avoiding the problems of overfitting, we also use cross-validation way to evaluate and choose the best model. We split 400 items to four parts which include 200(100+100), 100 and 100 to train, validate and test for each cross-validation function. The recording file name including split number, IRs location number, the number of overlapping sound events and recording number per split. Every label file name is equal to each recording file and the label file named as a csv extension which including the sound events, their respective temporal onset-offset times, azimuth and elevation angles. We visualize the distribution of azimuth and elevation angles on each split corresponding to five IRs in Fig. 8 and Fig. 9.

To better verify the effectiveness of our proposed model, we add different SNR that is -10 dB, -5 dB, 0 dB, 5 dB and 10 dB to extend datasets which each dataset is doubled in different SNR for two formats respectively. For example, a total of $54002 = 4000$ pieces of data are generated under the above five different SNRs. The core algorithm for compositing batch datasets with different SRN can be summarized as Algorithm

![Fig. 8. Azimuth angles distribution per impulse response in split 1, split 2, split 3 and split 4 from the top to bottom.](image)

### C. Evaluation

We train our network for 2D-DOA estimation as a regression task. The training, evaluation and test processing we use MSE function to calculate the loss between the ground truth labels and predicted DOA results. For a batch of $M$ samples, calculate the expectation of the sum of squares of the difference between the parameter estimated value and the
Fig. 9. Elevation angles distribution per impulse response in split 1, split 2, split 3 and split 4 from the top to bottom.

**Algorithm 1** Config Signal-Noise Ratio

**Input:** Wav Datasets  
**Output:** Different SNR Datasets

1. Read channels, samplewidth, framerate, nframes of each data and normalized batch frames data.  
2. Reshape batch frames data format to [nframes,nchannels].  
3. Generate the same length Gaussian white random noise.  
4. Then calculate the signal power $P_s$ and generate noise power $P_{n1}$ (assume the total length is N): $P_s = \frac{\sum_{i=1}^{N} (x_i)^2}{N}$, $P_{n1} = \frac{\sum_{i=1}^{N} (n_i)^2}{N}$  
5. Next calculate the power of noise : $P_n = k^2 \cdot P_{n1}$  
6. Transposition $k \rightarrow left$. $k = \sqrt{\frac{P_s}{10^{\frac{SNR}{10}} \cdot P_{n1}}}$  
7. Final Gaussian white random noise: $s = k \cdot Step2$  
8. Retrive wav format files with Gaussian white noise

The parameter true value of the $i$-th sample where $g$ means ground truth label and $g'$ means predicted label. The basic function of MSE can be viewed as follows:

$$MSE \left( g, g' \right) = \frac{\sum_{i=1}^{M} \left( g_i - g'_i \right)^2}{M}$$  \hspace{1cm} (14)

The predicted 2D-DOA coordination $(x', y', z')$ we recompute with the reference $(x, y, z)$ utilizing the central arc angle $\partial \in [0, 180]$. The $\partial$ can be reviewed as follows:

$$\partial = 2 \cdot \arcsin \left( \frac{\sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2}}{2} \right) \cdot \frac{180}{\pi}$$  \hspace{1cm} (15)

The 2D-DOA error can be calculated for the whole dataset as follows:

$$DOA error = \frac{\sum_{n=1}^{N} \partial((x_n, y_n, z_n), (x'_n, y'_n, z'_n))}{N}$$  \hspace{1cm} (16)

where $N$ is the whole number of 2D-DOA estimation for a recording of length $N$ time frames, and $\partial((x_n, y_n, z_n), (x'_n, y'_n, z'_n))$ means the angle between $n$-th predicted and correspondent ground truth DOA. The frame recall is used to explain time frames whether the ground truth equals to the predicted of every DOA estimation. It can be calculated as follows:

$$Frame recall = \frac{\sum_{n=1}^{N} \text{sign}(D_G^n = D_P^n)}{N}$$  \hspace{1cm} (17)

where $D_G^n$ and $D_P^n$ is the number of references and predicted DOAs in the list of all DOAs for a recording of length $T$ time frames at $n$-th frame respectively, $\text{sign}()$ indicating function returning one if $D_G^n = D_P^n$, else zero.

**D. Results**

1) **Data Preparation Processing:** We first use the short-time Fourier transform (STFT) to extract the spectral characteristics of the input audio to obtain a complex matrix containing amplitude and phase in which the window function uses the Hanning window to obtain better resolution and less spectral leakage and the frame shift duration is set to 0.02 seconds. The sample rate is set to 48000 and the length of the frame shift is the frame shift duration multiplier sample rate(48000 × 0.02 = 960). The window length is twice the frame shift that is $M = 1920$ and the number of STFT points is set to the window length. The final feature shape obtained after STFT is $(3000,1024,4)$ where $C = 4$ represents the number of used channels in which 1024 represents bins and 3000 represents the number of frames. To facilitate data processing, we combine the data of four channels to obtain the final feature characteristics of $(3000,4096)$. To eliminate dimensional-level differences generated by the data, we use a standardized method to map the data to the [0,1] interval scaling by calculating the mean and variance of the data. At the same time, we set the timing label corresponding to the positioning event category in the audio, and set the start and end time of the sampling to the number of seconds corresponding to the frame shift multiplier corresponding to the start and end of the duration, that is, the start and end frames corresponding to the duration. Finally, the obtained activation category event annotation shape is $[3000,11]$ and standardized to obtain the corresponding azimuth and elevation angle matrix $[3000,22]$ where 11 represents the category corresponding to 11 binary labels and 22 represents the corresponding normalized azimuth and elevation angle information. Finally, the ultimate label feature length is obtained by stitching $[3000,33]$.

2) **Model Details Configuration:** After the data preprocessing in the previous step, we obtained the standardized complex characteristic matrix with dimension $(3000,4096)$ and the standardized label data with dimension $(3000,33)$. The details of ESRNN-L and ESRNN-G is displayed in Table I and Table II. We can see that the ESRNN-L contains convmp-block$_i$ and followed by merged_residual_block$_i$ which $i$ from one to three means the number of each layer. The MP layer size of every convmp-block layer is $(8,8,4)$ separately, setting $K = 64$ of 2D convolution core size to $3 \times 3$ of every convmp-block layer and the spectrogram feature sequence length we set $T = 128$ frames.

For merged_residual_block, we first input the local or global characteristics of the output of convmp-block into the BN layer to further balance the data, and use the Relu activation function to activate the differing characteristics after scaling.
Next, we set the 2D convolution core size to $3 \times 3$ and set the data format for channel. First is initialized to he normal. In the residual structure branch of block, we use the sigmoid function to calculate the optimal threshold of the feature channel with the dimension of $C \ast W \ast H$ ($C$ is the number of channels, $W$ is the width of the feature, and $H$ is the height of the feature) after the FC output, and finally use the short connection to output the filtered feature value.

The hidden unit of the whole GRU nodes of the model is set to $P = 128$ and the number of unit layers is 1. The final FC layer we set the number of units $Q = 128$. Of course, we did not add a dropout layer to accelerate the model convergence for any layer in the whole model design process. Instead, we set that in each round of loss iterative optimization. If the optimal value of loss remains the minimum within five epochs, the model convergence will be determined, that is, the best model training result. We believe that this method ensures the parameter integrity of the model to a certain extent, provides a certain generalization ability for the convergence of the model, and also ensures the consistency of the parameters.

3) Trainable Parametric Analysis: Compared with ESRNN-L, ESRNN-G collects global signal spectrum features after conv-mp-blocks which can be tracked from Fig. 5 and Fig. 7. The first layer output of the merged_residual_block structure of EGRNN-L is $(\text{batch}_\text{size}, 64, 64, 64)$, the output of the second layer structure is $(\text{batch}_\text{size}, 64, 32, 4)$, and the output of the third layer structure shape is $(\text{batch}_\text{size}, 64, 32, 1)$. In contrast, we can see that the output of the global merged_residual_block layer structure of ESRNN-G is $(\text{batch}_\text{size}, 64, 64, 2)$ and the sum of the total parameters corresponding to the two layers is 248116 and 82448 respectively. In contrast to ESRNN-L, ESRNN-G uses a global hop connection to input directly to the GRU, while the GRU consists of updating the gating unit and resetting the gating unit. The parameter amount of the GRU is calculated as follows:

$$W_u = 3 \ast ((X_{dim} + H_{dim}) \ast H_{dim} + H_{dim}) \quad (18)$$

where $W_u$ the total number of parameters at the current moment in GUR, $X_{dim}$ represents the dimension of the current input vector and $H_{dim}$ means the dimension of output which is equal to the number of hidden layer unit. Due to the design of the global hop connection of ESRNN-G, the input dimension of its $W_u$ is reduced, so the number of parameters is also reduced as a whole. Therefore, the total trainable parameters of the ESRNN-G model are 530534 compared with the total trainable parameter of the ESRNN-L model of 652954 and the total number of parameters of ESRNN-G is reduced by 122420.

Compared to the total trainable parameter of BCRNN 611201, the global connection consistency of ESRNN-G reduces the total trainable parameter by 557667. The final model of ESRNN-G is only 6.43MB, a decrease of about 10% compared to BCRNN's 7.11MB.

4) Experimental Results Analysis: We used the TAU Spatial Sound Events 2019-Microphone Array dataset training model. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. Experimental Results Analysis: We used the TAU Spatial Sound Events 2019-Microphone Array dataset training model. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$. To obtain more characteristic information and reduce the risk of model overfitting, we use cross-validation to divide the data into four parts $[[1,2], [2,3], [3,4], [4,1], [1,4], [2,3], [3,4], [4,1]]$.

Fig. 10. Training loss change process of BCRNN, ESRNN-L, ESRNN-G in MIC. The ordinate represents 2D-DOA error and the abscissa represents epoch. From top to bottom, left to right is $[[1,2], [2,3], [3,4], [4,1]]$, $[[2,3], [3,4], [4,1], [1,4]]$, $[[3,4], [4,1], [1,4]]$, and $[[4,1], [1,4]]$.

Fig. 11. Visualize the 2D-DOA input and outputs for simulated datasets.
TABLE III
VALIDATION AND TEST RESULTS OF BCRNN, ESRNN-L, ESRNN-G VALIDATION LOSS IN MIC. AVERAGE MEANS THE AVERAGE RESULT FOR CROSS-VALIDATION. BEST MEANS THE BEST RESULT FOR CROSS-VALIDATION. THE BEST RESULTS ARE SHOWN IN BOLD BLACK

<table>
<thead>
<tr>
<th>Validation Results</th>
<th>DOA Error</th>
<th>Frame Recall</th>
<th>DOA Error</th>
<th>Frame Recall</th>
<th>DOA Error</th>
<th>Frame Recall</th>
<th>DOA Error</th>
<th>Frame Recall</th>
<th>DOA Error</th>
<th>Frame Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCRNN [9]</td>
<td>38</td>
<td>0.85</td>
<td>41.5</td>
<td>0.84</td>
<td>36</td>
<td>0.82</td>
<td>35.6</td>
<td>0.85</td>
<td>37.7</td>
<td>0.84</td>
</tr>
<tr>
<td>ESRNN-L</td>
<td>29.4</td>
<td>0.83</td>
<td>28.2</td>
<td>0.83</td>
<td>32.1</td>
<td>0.79</td>
<td>33.9</td>
<td>0.82</td>
<td>30.9</td>
<td>0.82</td>
</tr>
<tr>
<td>ESRNN-R</td>
<td>30.8</td>
<td>0.85</td>
<td>32.6</td>
<td>0.84</td>
<td>32.3</td>
<td>0.82</td>
<td>27.4</td>
<td>0.87</td>
<td>30.7</td>
<td>0.85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Results</th>
<th>DOA Error</th>
<th>Frame Recall</th>
<th>DOA Error</th>
<th>Frame Recall</th>
<th>DOA Error</th>
<th>Frame Recall</th>
<th>DOA Error</th>
<th>Frame Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCRNN [9]</td>
<td>36.7</td>
<td>0.85</td>
<td>39</td>
<td>0.81</td>
<td>38.7</td>
<td>0.83</td>
<td>34.3</td>
<td>0.81</td>
</tr>
<tr>
<td>ESRNN-L</td>
<td>29.5</td>
<td>0.82</td>
<td>26.6</td>
<td>0.8</td>
<td>33.8</td>
<td>0.81</td>
<td>30.9</td>
<td>0.77</td>
</tr>
<tr>
<td>ESRNN-R</td>
<td>30.8</td>
<td>0.84</td>
<td>29.2</td>
<td>0.82</td>
<td>33.4</td>
<td>0.84</td>
<td>25.8</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Fig. 12. Test results range from -10dB to 10dB of synthetic data under different SNRs. The top line chart is 2D-DOA error. The middle histogram means frame recall and the bottom is spectrum diagram.

the algorithm shown in Algorithm I. We use the best model filter from Table III to compare the unseen behavior performance of our model on MIC format under different SNR ratios. The experimental results are shown in Fig. 12.

As shown the bottom line in Fig. 12, with the gradual reduction of the SNR of the synthesized MIC data, the amplitude and phase characteristics of the visualization are gradually submerged by Gaussian white noise and the spectral characteristics of the audio data are gradually blurred. It can be found that when the SNR is above 0 dB, ESRNN-G can extract better spectral characteristic information from the whole world because of the light noise submersion. Therefore, ESRNN-G has good effect. The loss with the lowest experimental effect is 32.1 and 35.7 when SNR = 10dB and SNR = 5dB, respectively, which is nearly 19% and 15% lower than 39.9 and 42.0 of BCRNN. However, when SNR is lower than 0 dB, ESRNN-G cannot effectively filter out the noise mixing information at the lower level from the upper level because the noise almost submerges the spectrum information such as amplitude and phase, while ESRNN-L can produce better experimental results because it can effectively retain more spectrum feature information layer by layer using soft threshold from the low-order feature extraction. The 2D-DOA error with the lowest effective in the ESRNN-L experiment was 34.2, 35.5 and 37.1 when SNR = 0dB, SNR = −5dB and SNR = −10dB, respectively, with downgrade about 21%, 15% and 11% compared with 43.4, 42.0 and 41.7 of BCRNN.

To verify the robustness of the proposed model, we still compare the traditional MUSIC algorithm with deep learning based BCRNN in FOA datasets (from 0 dB to 30 dB) using the best saved model of FOA to test. The best experimental results are shown in Table IV.

The traditional MUSIC algorithm will completely lose resolution when the SNR is lower than -5 dB [35]. Therefore, we focus on comparing the test results with different models when the SNR is higher than 0 dB in Table IV. It can be seen from Table IV that the optimal effect of ESRNN-G can reach 23.2 in the FOA dataset (30 dB) test which is nearly 9% lower than that of BCRNN. It also can be seen from Table IV that the DOA error will increase as the SNR of the FOA data decreasing gradually but the robustness of ESRNN-L and ESRNN-G is still optimal compared with MUSIC and BCRNN.

V. CONCLUSIONS AND FUTURE WORK

In this work, we proposed two soft thresholding filter structures based on BCRNN which named ESRNN-L and ESRNN-G in different SNR scenarios to evaluate the elevation and azimuth angle of 2D-DOA estimation. To enhance the robustness and generalization of our model, ESRNN-L and ESRNN-G were designed for filtering low-level local noise and high-level global noise respectively, which used the incoherent fusion adaptive function between noise and the key characteristic spectrum. ESRNN-L filtered noise information layer by layer between shallow local signals and noises while ESRNN-G was designed for global noise capture. ESRNN-G enlarged noise information layer by layer through progressive convolution feature extraction from non-coherence of low-level signal-noise information and finally set zero separation noise information by global soft threshold adaptive filter. We made full use of the self-attention focusing to design adaptive soft threshold functions based on local and global to reduce noise at different SNR ratios.

To verify the validity of the experiment, we compared ESRNN with BCRNN and traditional MUSIC algorithm on TAU Spatial Sound Events 2019 datasets. The experimental results shown that the 2D-DOA error of the optimal result was reduced by nearly 24% compared with BCRNN. To verify the generalization performance of the proposed model, we stillled doing experiments on different synthetic reverberant SNR (-10 dB, -5 dB, 0 dB, 5 dB and 10 dB) datasets of TAU Spatial Sound Events 2019 datasets mixed with Gaussian white noise. The average 2D-DOA error of ESRNN-L/ESRNN-G had 10%-13%/3%-9% downward in different
TABLE IV
THE COMPARE OF BCRNN, ESRNN-L, ESRNN-G AND MUSIC ALGORITHM IN FOA DATASET (0 dB, 5 dB 10 dB, 30 dB). THE BEST RESULTS ARE SHOWN IN BOLD BLACK

<table>
<thead>
<tr>
<th></th>
<th>FOA 30 dB</th>
<th>FOA 10 dB</th>
<th>FOA 5 dB</th>
<th>FOA 0 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOA Error</td>
<td>Frame Recall</td>
<td>DOA Error</td>
<td>Frame Recall</td>
<td>DOA Error</td>
</tr>
<tr>
<td>BCRNN [9]</td>
<td>25.7</td>
<td>0.86</td>
<td>30</td>
<td>0.5</td>
</tr>
<tr>
<td>ESRNN-L</td>
<td>26.2</td>
<td>0.79</td>
<td>29.1</td>
<td>0.48</td>
</tr>
<tr>
<td>ESRNN-G</td>
<td>23.2</td>
<td>0.86</td>
<td><strong>28.6</strong></td>
<td>0.45</td>
</tr>
<tr>
<td>MUSIC [32]</td>
<td>-</td>
<td>-</td>
<td>42.9</td>
<td>-</td>
</tr>
</tbody>
</table>

synthetic reverberant SNR datasets (-10 dB, -5 dB, 0 dB, 5 dB and 10 dB) respectively. The best 2D-DOA error of ESRNN-L/ESRNN-G have 5%-19%/7%-21% downward in different synthetic reverberant SNR datasets (-10 dB, -5 dB, 0 dB, 5 dB and 10 dB) respectively. The experiment powerfully proved that ESRNN-G had the strong global consistency and relatively excellent resolution when the SNR is greater than 5 dB. And ESRNN-L had a strong local recursion when the SNR was lower than 0 dB so it had a good performance with the ability of low-level SNR characteristic discrimination. The experimental results also fully demonstrated the generalization of our proposed model at low SNR. The above trial proved the assumption that ESRNN could catch cleaner spectrum for preferably feature representations. The ESRNN we proposed can infer the azimuth and elevation angle of the active sound source event with 2D-DOA estimation, and this network is not only used to acoustic microphones but also to wireless array sensors or sonars for real-time conditions.

For feature work, DOA estimation can not only be applied to aeroacoustic direction finding using microphone but also has a great quantity of non-Gaussian mixed noise (such as impulse noise interference) in many practical application scenarios (such as underwater DOA estimation) and sparse signal collected in small snapshot scenarios. Therefore, the study of real-time estimation of sparse DOA estimation in non-Gaussian white noise scenarios [36] is one of the main directions we want to explore in the future. In addition, we only design ESRNN-L/ESRNN-G structure for noise filtering of DOA estimation in this work but sound event location is only a subbranch of sound event detection and location [37]. When we adding another output applies the model to the task of sound event detection and location, the accuracy of the model for sound event detection decreases by an average of 3%. So how to balance the overall accuracy between the two is another major direction for our future research.

REFERENCES


Bin Zhang (Ph.D. in progress) is pursuing a doctor’s degree in computing intelligence and applications at Ocean University of China. In 2022, he successfully obtained a master’s degree in computer science from Inner Mongolia University and studied for a further doctorate from Ocean University of China in the same year. During his doctoral studies, his main research interests were artificial intelligence, sound source localization, speech signal processing, and passive localization of sound field calculations.

Jiawen He (Ph.D. in progress) is pursuing a master’s degree in computer software and theory at Ocean University of China in 2019. In 2021, she obtained the qualification for continuous study of a master and doctoral degrees and started a Ph.D. in artificial intelligence. Her research interests include passive target localization using hydroacoustic array signal data, array signal processing, physically constrained neural networks, and multi-task learning.

Peishun Liu is an associate professor at Ocean University of China, with a graduate degree and a doctoral degree. He is also the deputy director of the Production Teaching Integration Center of the School of Information Science and Engineering. The main research direction is network and information security, and the research field are cloud computing security, artificial intelligence. The National Information Security Assessment is a registered information security instructor.

Liang Wang completed his doctoral degree in marine information exploration and processing from the Department of Ocean Technology, School of Information Science and Engineering, China Ocean University in 2009, and served as a lecturer in the Department of Ocean Technology, School of Information Science and Engineering, China Ocean University in the same year. He is mainly engaged in the field of underwater acoustic signal processing, including underwater target positioning, underwater acoustic communication, and other fields.

Hao Zhou graduated from Shandong Jianzhu University with a Bachelor’s degree in Network Engineering. Since 2021, he has been studying for a master’s degree in electronic information at the Department of Information Science and Engineering at Ocean University of China. His research interests include audio signal processing, underwater acoustic target recognition and location, and self-supervised learning.

Xuening Wang received the bachelor’s degree in Electronic Information Engineering from Ocean University of China, Qingdao, China, in 2022. And currently studying for the master’s degree in Computer technology at Ocean University of China, Qingdao, China. Her research interest is underwater sound source localization based deep learning, including distance estimation and arrival direction estimation.
Ruichun Tang is professor, doctoral supervisor worked at Ocean University of China. Her main research interests include intelligent network streaming media perception, underwater image application technology, smart home applications, and medical big data. Since joining Ocean University of China in 2002, she has presided over national key research and development projects for many times. During her teaching period, she has published over 50 academic papers and over 20 soft works on invention patents.