Chen *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2021) 2021:35
https://doi.org/10.1186/s13636-021-00221-8

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

# Nonlinear residual echo suppression based on dual-stream DPRNN

Hongsheng Chen[1,2,3], Guoliang Chen[1,2,3], Kai Chen[1,2,3] and Jing Lu[1,2,3*]

**Abstract**

The acoustic echo cannot be entirely removed by linear adaptive filters due to the nonlinear relationship between the echo and the far-end signal. Usually, a post-processing module is required to further suppress the echo. In this paper, we propose a residual echo suppression method based on the modification of dual-path recurrent neural network (DPRNN) to improve the quality of speech communication. Both the residual signal and the auxiliary signal, the far-end signal or the output of the adaptive filter, obtained from the linear acoustic echo cancelation are adopted to form a dual-stream for the DPRNN. We validate the efficacy of the proposed method in the notoriously difficult double-talk situations and discuss the impact of different auxiliary signals on performance. We also compare the performance of the time domain and the time-frequency domain processing. Furthermore, we propose an efficient and applicable way to deploy our method to off-the-shelf loudspeakers by fine-tuning the pre-trained model with little recorded-echo data.

**Keywords:** Residual echo suppression, Dual-path recurrent neural network, Dual-stream

## 1 Introduction

The acoustic echo is generated from the coupling between the loudspeaker and the microphone in full-duplex hands-free telecommunication systems or smart speakers. It severely deteriorates the quality of speech communication and significantly degrades the performance of automatic speech recognition (ASR) within the smart speakers. Typical linear acoustic echo cancelation (LAEC) methods use adaptive algorithms to identify the impulse response between the loudspeaker and the microphone [1]. Time-domain least mean square (LMS) algorithms [2, 3] are often employed in delay-sensitive situations. Frequency-domain LMS algorithms are often utilized to guarantee both fast convergence speed and low computational load [2]. The frequency-domain adaptive Kalman filter (FDKF) [4] is also a commonly used method with several efficient variations proposed recently [5, 6].

The performance of LAEC methods severely degrades when nonlinear distortion is non-negligible in the acoustic echo path [7]. Usually, a residual echo suppression (RES) module is required to further suppress the echo. The RES is usually conducted by estimating the spectrum of the residual echo based on the far-end signal, filter coefficients, and the residual signal of LAEC [8–13]. However, it is difficult for the signal-processing-based RES to balance well between the residual echo attenuation and the near-end speech distortion.

Recently, deep neural network (DNN) has been introduced into RES due to its powerful capability of modeling nonlinear systems, including the time domain and time-frequency (TF) domain methods. TF-domain methods adopt the short-time Fourier transform (STFT) to extract spectral features. The fully connected network (FCN) was employed to exploit multiple-input signals in RES [14]. The bidirectional or unidirectional recurrent neutral network (RNN) was also introduced to RES [15–17]. These methods ignore the coupling between magnitude and phase and are unable to recover the phase information, leading to limited performance [18]. Inspired by

*Correspondence: lujing@nju.edu.cn
[1]Key Laboratory of Modern Acoustics, Nanjing University, 210093 Nanjing, China
[2]NJU-Horizon Intelligent Audio Lab, Horizon Robotics, 100094 Beijing, China
Full list of author information is available at the end of the article

the fully convolutional time-domain audio separation network (Conv-TasNet) [18], we proposed a RES method based on the multi-stream Conv-TasNet, where both the residual signal of the LAEC system and the output of the adaptive filter are adopted to form multiple streams [19]. The benefit of introducing the auxiliary signals into the network was validated by simulations. However, the model employs a complicated network structure and is not efficient enough to exploit the information of multiple streams, resulting in large number of parameters which restricts its practical application. Moreover, the benefit of multi-streams is yet to be validated by experiments on off-the-shelf loudspeakers.

Dual-path recurrent neural network (DPRNN) [20] was recently proposed for speech separation task and achieves the state-of-the-art (SOTA) performance on WSJ0-2mix dataset. It utilizes an encoder module for feature extraction and employs RNNs for time series modeling. To overcome the inefficiency of RNN in modeling long sequences, DPRNN splits the long sequential input into smaller chunks and applies intra- and inter-chunk operations iteratively. Compared with Conv-TasNet, DPRNN shows superiority in both performance and parameter number [20]. Moreover, its RNN-based structure has advantages over Conv-TasNet in memory consumption when processing online.

In this paper, we extend our previous work on multi-stream Conv-TasNet. We adopt the residual signal of LAEC and the auxiliary signal to create two streams, and propose two DPRNN-structure networks in the time domain and TF domain respectively to effectively exploit their information. To validate the efficacy of our proposed RES methods, we compare them with several typical methods on both artificial-echo dataset and recorded-echo dataset. Furthermore, we regard the well-trained model on artificial-echo dataset as a pre-trained model and fine-tune it on recorded-echo dataset. Different fine-tuning strategies are investigated to achieve a balance between the performance and the training cost.

## 2 Model description

### 2.1 Problem formulation

The AEC system with RES post-filter is depicted in Fig. 1, where $x(n)$ is the far-end signal, $\hat{y}(n)$ is the output of the adaptive filter, and $H(z)$ represents the echo path transfer function. The microphone signal $d(n)$ consisting of the echo $y(n)$, the near-end speech $s(n)$, and background noise $v(n)$ can be expressed as

$$d(n) = s(n) + y(n) + v(n) \tag{1}$$

The signal of the LAEC $s_{\mathrm{AEC}}(n)$ is given by subtracting the output of the adaptive filter $\hat{y}(n)$ from the microphone signal $d(n)$, with

$$\hat{y}(n) = \hat{h}(n) * x(n) \tag{2}$$

$$s_{\mathrm{AEC}}(n) = d(n) - \hat{y}(n) \tag{3}$$

where $\hat{h}(n)$ denotes the adaptive filter and $*$ represents convolution operation. Due to the inevitable nonlinear feature in the echo path, the LAEC cannot perfectly attenuate the echo, and $s_{\mathrm{AEC}}(n)$ can be regarded as the mixture of the residual echo, background noise, and the near-end signal. The RES can be designed from the viewpoint of speech separation, but unlike the standard speech separation problem, the auxiliary information extracted from the adaptive filter can be exploited to improve the performance. In this paper, we employ $s_{\mathrm{AEC}}(n)$ together with an auxiliary signal, $x(n)$ or $\hat{y}(n)$, to construct a dual-stream DPRNN (DSDPRNN).

### 2.2 Model design

Figure 2 outlines the structure of our proposed DPRNN-based RES method, which consists of two encoder modules, a suppression module, and a decoder module. The two encoder modules are used to extract features from

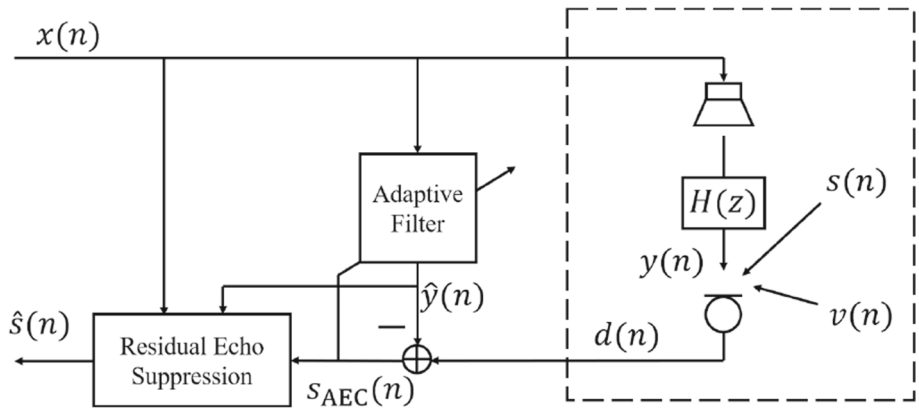


**Fig. 1** The diagram of AEC system with RES post-filter

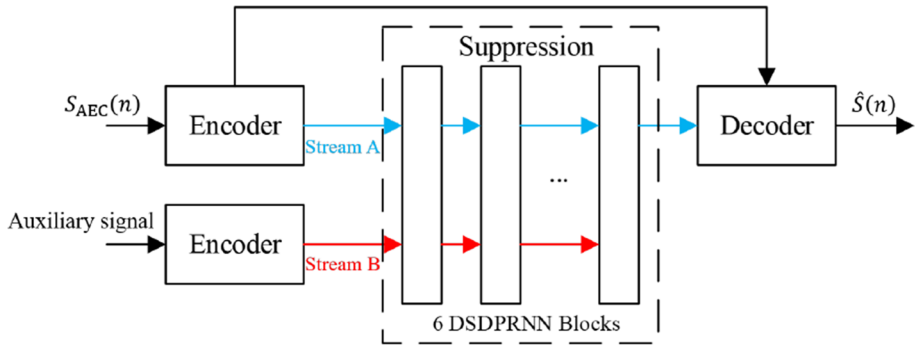


**Fig. 2** The structure of our proposed DPRNN-based RES method. The blue line and the red line represent stream A and stream B respectively

$s_{\mathrm{AEC}}(n)$ and the auxiliary signal to form two streams, streams A and B, respectively. The suppression module suppresses the residual echo and recovers the near-end signal by exploiting the information of streams A and B. The decoder transforms the output of the suppression module into masks and converts the masked feature back to the waveform. The difference between the time-domain and the TF-domain methods mainly lies in the encoder and the decoder, while the structure of the suppression module is the same.

Figure 3 shows the structure of the encoder and the decoder in the time-domain method. The encoder takes a time-domain waveform $\boldsymbol{u}$ as input and converts it into a time series of $N$-dimensional representations using a 1-D convolutional layer with a kernel size $L$ and 50% overlap, followed by a ReLU activation function

$$\boldsymbol{W} = \mathrm{ReLU}(\mathrm{Conv1d}(\boldsymbol{u})) \tag{4}$$

where $\boldsymbol{W} \in \mathbb{R}^{G\times N}$ with length $G$ is the output of the operation. Then, $\boldsymbol{W}$ is transformed into $C$-dimensional representations by a fully connected layer and divided into $T = 2G/K - 1$ chunks of length $K$, where the overlap between chunks is 50%. All chunks are then stacked together to form a 3-D tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{T\times K\times C}$. The decoder applies overlap-add operation to the output of suppression module $\boldsymbol{\mathcal{Y}}_s \in \mathbb{R}^{T\times K\times C}$, followed by a PReLU activation [21], to form the output $\boldsymbol{Q} \in \mathbb{R}^{G\times C}$. Then, an $N$-dimensional fully connected layer with a ReLU activation is applied to $\boldsymbol{Q}$ to

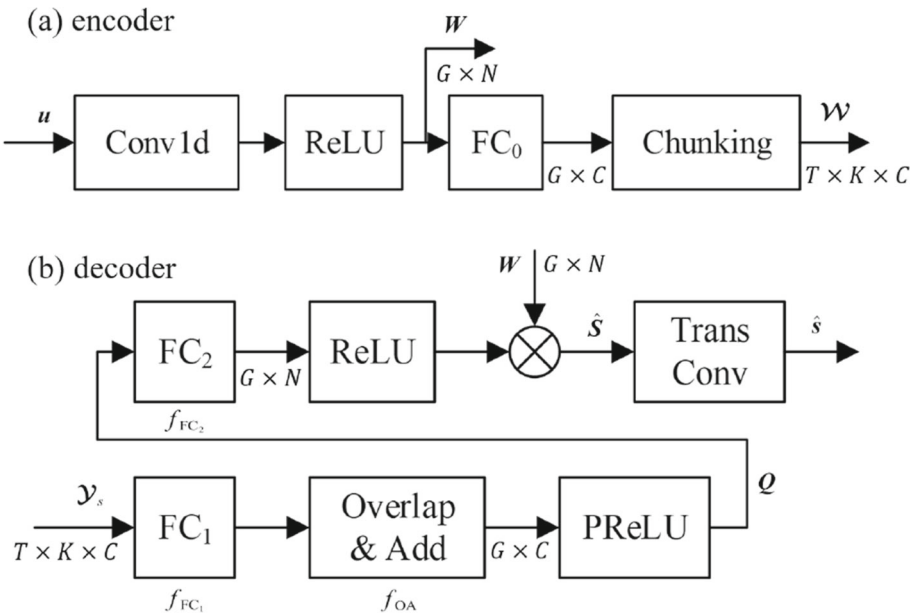


**Fig. 3** The structure of the encoder and the decoder in the time-domain method

obtain the mask of $\boldsymbol{W}$, and the estimation of clean speech's representation $\hat{\boldsymbol{S}}$ is obtained by

$$\hat{\boldsymbol{S}} = \mathrm{ReLU}\left(f_{\mathrm{FC}_2}(\boldsymbol{Q})\right) \odot \boldsymbol{W} \tag{5}$$

$$\boldsymbol{Q} = \mathrm{PReLU}\left(f_{\mathrm{OA}}\left(f_{\mathrm{FC}_1}(\boldsymbol{\mathcal{Y}}_s)\right)\right) \tag{6}$$

where $f_{\mathrm{FC}_i}$, $i = 1, 2$ represents the fully connected layer, $f_{\mathrm{OA}}$ represents the overlap-add operation, and $\odot$ denotes the element-wise multiplication. A 1-D transposed convolution layer is utilized to convert the masked representation back to waveform signal $\hat{\boldsymbol{s}}$.

The intra-chunk operation of DPRNN can also be applied in the frequency domain. Figure 4 shows the structure of the encoder and the decoder in the TF-domain method. We first obtain the TF representation $\boldsymbol{Z} \in \mathbb{C}^{T' \times F}$ by the STFT operation with a $Q$-point Hamming window and 50% overlap, where $F = Q/2 + 1$ is the number of effective frequency bins. We concatenate the real and imaginary component of $\boldsymbol{Z}$ to form a 3-D tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{T' \times F \times 2}$. The 3-D representation $\boldsymbol{\mathcal{W}}' \in \mathbb{R}^{T' \times K' \times C'}$ is then obtained by a 2-D convolutional layer with $C'$ output channel. The kernel size is $5 \times 5$ and the stride is $1 \times 2$, where $K' = \frac{F-3}{2}$ is the number of down-sampled frequency bins. The frame length, the chunk size, and the feature dimension $T', K', C'$ correspond to $T, K, C$ in the time-domain encoder respectively, and the output is further processed by the same suppression module. The decoder takes the output of suppression module $\boldsymbol{\mathcal{Y}}_s'$ as input and successively applies two fully connected layers, followed by a PReLU and a ReLU activation respectively, to form the output $\boldsymbol{\mathcal{Q}}' \in \mathbb{R}^{T' \times K' \times C'}$. Then, $\boldsymbol{\mathcal{Q}}'$ is processed by two independent 2-D transposed convolutional layers, called Trans Conv_A and Trans Conv_P, with the kernel size $5 \times 5$ and the stride $1 \times 2$. Trans Conv_A with a ReLU activation function is utilized to estimate the mask of TF bins. Trans Conv_P followed by a normalization operation for each TF bin is employed to estimate the real part and imaginary part of the phase information. Finally, the spectrogram of the output signal $\hat{\boldsymbol{\mathcal{S}}}'$ is estimated by

$$\hat{\boldsymbol{\mathcal{S}}}' = \left(abs(\boldsymbol{Z}) \circ \mathbb{1}^2\right) \odot \left(\boldsymbol{\mathcal{A}} \times_3 \mathbb{1}^{1\times 2}\right) \odot \boldsymbol{\mathcal{P}} \tag{7}$$

$$\boldsymbol{\mathcal{A}} = \mathrm{ReLU}\left(f_{\mathrm{TC}}^{A}(\boldsymbol{\mathcal{Q}}')\right) \in \mathbb{R}^{T' \times F \times 1} \tag{8}$$

$$\boldsymbol{\mathcal{P}} = \mathrm{Norm}\left(f_{\mathrm{TC}}^{P}(\boldsymbol{\mathcal{Q}}')\right) \in \mathbb{R}^{T' \times F \times 2} \tag{9}$$

where $f_{\mathrm{TC}}^{A}$, $f_{\mathrm{TC}}^{P}$, and Norm represent the functions of Trans Conv_A, Trans Conv_P, and the normalization operation for each TF bin respectively. We use $\mathbb{1}^{I_1 \times I_2 \times \ldots \times I_M}$, $\circ$, and $\times_i$ to denote an all-ones tensor, the outer product, and the

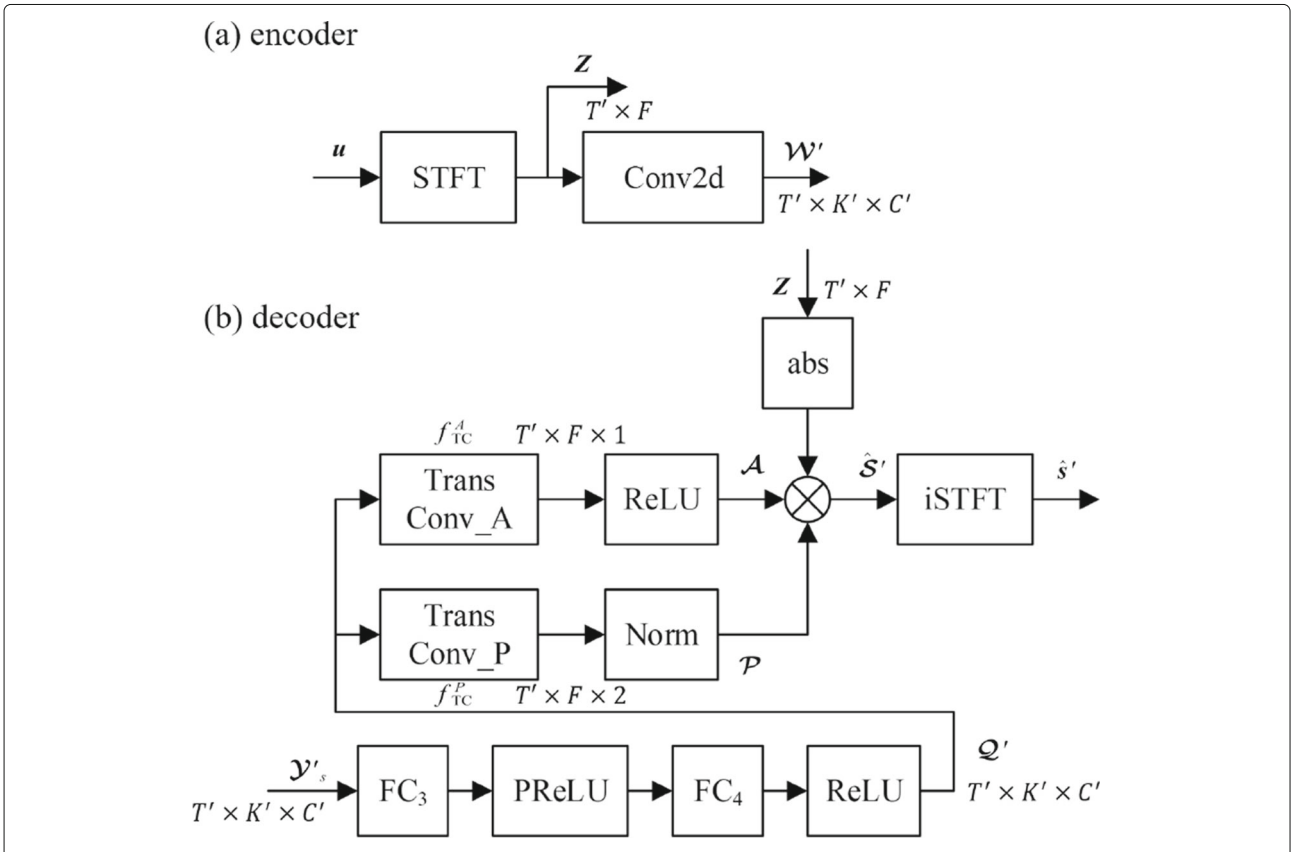


**Fig. 4** The structure of the encoder and the decoder in the TF-domain method

mode-$i$ product [22]. The outer product between the tensor $\mathcal{H} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ and the vector $\boldsymbol{g} \in \mathbb{R}^J$ is defined as

$$\mathcal{R} = \mathcal{H} \circ \boldsymbol{g} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M \times J} \tag{10}$$

$$\mathcal{R}_{i_1, i_2, \ldots, i_M, j} = \mathcal{H}_{i_1, i_2, \ldots, i_M} \cdot \boldsymbol{g}_j \tag{11}$$

The mode-$i$ product between the tensor $\mathcal{H}$ and the matrix $\boldsymbol{D} \in \mathbb{R}^{I_M \times J}$ is defined as

$$\mathcal{R} = \mathcal{H} \times_M \boldsymbol{D} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{M-1} \times J} \tag{12}$$

$$\mathcal{R}_{i_1, i_2, \ldots, i_{M-1}, j} = \sum_{i_M=1}^{I_M} \mathcal{H}_{i_1, i_2, \ldots, i_M} \cdot \boldsymbol{D}_{i_M, j} \tag{13}$$

Similar to the operation in [23], $\mathcal{A}$ and $\mathcal{P}$ in Eqs. 8 and 9 act as the amplitude mask and the phase prediction result respectively. After that, an inverse STFT operation is applied to convert $\hat{\mathcal{S}}'$ back to the waveform signal $\hat{\boldsymbol{s}}'$.

The suppression module consists of six DSDPRNN blocks, each of which contains two dual-stream RNN (DSRNN) blocks corresponding to intra-chunk and inter-chunk processing respectively. Figure 5 presents the structure of the proposed DSRNN block, where each stream is successively processed by an RNN layer, a fully connected layer, and a normalization layer. The RNN layer in each intra-chunk block is a bidirectional RNN layer applied along the chunk dimension with $C/2$ output channels for each direction, while the RNN layer in each inter-chunk block is a unidirectional RNN layer with $C$ output channels and is applied along the frame dimension. Let $\mathcal{V}_i^0 \in \mathbb{R}^{T \times K \times C}$ denote the input tensors of stream $i$, then the output of the RNN layer $\mathcal{V}_i^1$ can be expressed as

$$\mathcal{V}_i^1 = f_{\mathrm{RNN}_i}\left(\mathcal{V}_i^0\right), i = A \text{ or } B \tag{14}$$

where $f_{\mathrm{RNN}_i}$ represents the function of the RNN layer. The feature in $\mathcal{V}_A^1$ and $\mathcal{V}_B^1$ is then mixed by

$$\mathcal{V}_A^2 = \mathcal{V}_A^1 + \left(\mathbb{1}^T \circ \mathbb{1}^K \circ \boldsymbol{\alpha}\right) \odot \mathcal{V}_B^1 \tag{15}$$

$$\mathcal{V}_B^2 = \mathcal{V}_B^1 + \left(\mathbb{1}^T \circ \mathbb{1}^K \circ \boldsymbol{\beta}\right) \odot \mathcal{V}_A^1 \tag{16}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^C$ are trainable parameters. The output $\mathcal{V}_i^2$ is concatenated to the corresponding raw input $\mathcal{V}_i^0$ and then processed by a fully connected layer with $C$ output channels. $\mathcal{V}_i^3$ is obtained with a residual connection and can be formulated as

$$\mathcal{V}_i^3 = f_{\mathrm{FC}_i}\left(\left[\mathcal{V}_i^2, \mathcal{V}_i^0\right]\right) + \mathcal{V}_i^0, i = A \text{ or } B \tag{17}$$

where $[\cdot, \cdot]$ represents the concatenation operation. The concatenation and projection are applied along the chunk dimension in intra-chunk blocks, and these operations are also applied along the feature dimension in inter-chunk blocks. The output $\mathcal{V}_i^4$ of the DSRNN block is then obtained by a normalization layer to $\mathcal{V}_i^3$, except for those in the last DSDPRNN block where $\mathcal{V}_i^3$ serves as the output

$$\mathcal{V}_i^4 = f_{\mathrm{Norm}_i}\left(\mathcal{V}_i^3\right), i = A \text{ or } B \tag{18}$$

where $f_{\mathrm{Norm}_i}$ denotes the function of the normalization layer. The features of streams A and B are processed iteratively by the intra-chunk and the inter-chunk DSRNN blocks, and the output of stream A in the last DSDPRNN block is regarded as the output of the suppression module. We use Group Normalization [24] with a group number of 2. The input feature of the normalization layer $\mathcal{X} \in \mathbb{R}^{T \times K \times C}$ is first divided into two groups as

$$\mathcal{X} = \left[\hat{\mathcal{X}}^1, \hat{\mathcal{X}}^2\right], \ \hat{\mathcal{X}}^1, \hat{\mathcal{X}}^2 \in \mathbb{R}^{T \times K \times \frac{C}{2}}, \tag{19}$$

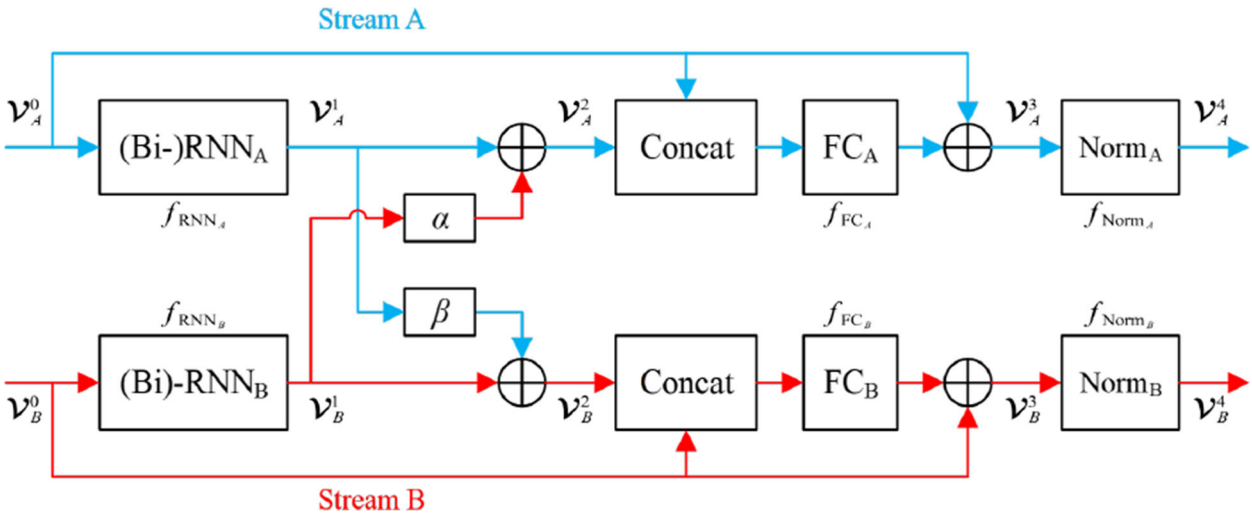


**Fig. 5** The structure of the DSRNN block. The blue line and the red line represent stream A and stream B respectively

and the output is formulated as

$$f_{\text{Norm}}(\boldsymbol{\mathcal{X}}) = \left[\hat{\boldsymbol{\mathcal{Y}}}^1, \hat{\boldsymbol{\mathcal{Y}}}^2\right] \tag{20}$$

with

$$\hat{\boldsymbol{\mathcal{Y}}}^i_{l,k,c} = \frac{\hat{\boldsymbol{\mathcal{X}}}^i_{l,k,c} - \mu\left(\hat{\boldsymbol{\mathcal{X}}}^i_l\right)}{\sqrt{\sigma\left(\hat{\boldsymbol{\mathcal{X}}}^i_l\right) + \epsilon}} \cdot \boldsymbol{\gamma}^i_c + \boldsymbol{\beta}^i_c,\ i = 1, 2 \tag{21}$$

and

$$\mu\left(\hat{\boldsymbol{\mathcal{X}}}^i_l\right) = \frac{2}{CK}\sum_{k=1}^{K}\sum_{c=1}^{C/2}\hat{\boldsymbol{\mathcal{X}}}^i_{l,k,c},\ i = 1, 2 \tag{22}$$

$$\sigma\left(\hat{\boldsymbol{\mathcal{X}}}^i_l\right) = \frac{2}{CK}\sum_{k=1}^{K}\sum_{c=1}^{C/2}\left[\hat{\boldsymbol{\mathcal{X}}}^i_{l,k,c} - \mu\left(\hat{\boldsymbol{\mathcal{X}}}^i_l\right)\right]^2,\ i = 1, 2 \tag{23}$$

where the subscripts $l, k, c$ denote the index of the 3-D tensor, $\boldsymbol{\gamma}^i, \boldsymbol{\beta}^i \in \mathbb{R}^{C/2}$ are trainable parameters, and $\epsilon$ is a small constant for numerical stability.

### 2.3 Training target

We choose the maximization of the scale-invariant source-to-noise ratio (SISNR) [18] as the training target

$$\boldsymbol{s}_{\text{target}} = \frac{| < \hat{\boldsymbol{s}}, \boldsymbol{s} > |\boldsymbol{s}}{||\boldsymbol{s}||^2} \tag{24}$$

$$\boldsymbol{e}_{\text{noise}} = \hat{\boldsymbol{s}} - \boldsymbol{s}_{\text{target}} \tag{25}$$

$$\text{SISNR} = 10\log_{10}\frac{||\boldsymbol{s}_{\text{target}}||^2}{||\boldsymbol{e}_{\text{noise}}||^2} \tag{26}$$

where $\hat{\boldsymbol{s}}, \boldsymbol{s}$ are the estimated and the target clean sources respectively, $< \cdot, \cdot >$ represents the dot product of vectors, and $||\boldsymbol{s}||$ denotes the $l_2$ norm of $\boldsymbol{s}$.

## 3 Experiments

### 3.1 Dataset

Unlike telecommunication systems, where the far-end signal is usually speech, music often acts as the "far-end" signal for smart loudspeakers. Therefore, we use both speech and music as the far-end signal, and the near-end signal is speech. We choose LibriSpeech [25] as the speech dataset and MUSAN [26] as the music dataset. We randomly choose 225, 25, and 40 different speakers from LibriSpeech, and 497, 48, and 115 pieces of music from MUSAN for training, validation, and test respectively. The audio data is sampled at 16 kHz and split into 4-s segments. Totally, we use 26,556, 1083, and 920 segments of 4-s speech and 101,956, 1083, and 920 segments of 4-s music for training, validation, and test respectively.

The clipping function and the sigmoidal function, although not precise models for the actual nonlinearity of the loudspeakers, are commonly utilized numerical models in many previous works on nonlinear acoustic suppression [15, 17]. Thus, the clipping function, sigmoidal function, and convolution operation are successively applied to the far-end signal to generate the simulated echo. The clipping function is either soft-clipping or hard-clipping function [27]

$$Clip_{\text{soft}}(x(n)) = \frac{x_{\max}x(n)}{\sqrt{|x_{max}|^2 + |x(n)|^2}} \tag{27}$$

$$Clip_{\text{hard}}(x(n)) = \begin{cases} x_{\max}, & \text{if } x(n) > x_{\max}, \\ -x_{\max}, & \text{if } x(n) < -x_{\max}, \\ x(n), & \text{otherwise.} \end{cases} \tag{28}$$

where $x_{\max} = \Theta \cdot \max(abs(x(n)))$ determines the maximum value of the clipping function. Three types of soft-clipping and three types of hard-clipping functions are utilized with the parameter $\Theta$ set to 0.6, 0.8, and 0.9.

We also use the sigmoidal function [28] to approximate the nonlinearity of a loudspeaker

$$NL(x(n)) = \frac{1}{1 + e^{[-a \cdot b(n)]}} - \frac{1}{2} \tag{29}$$

$$b(n) = \frac{3}{2}x(n) - \frac{3}{10}x^2(n) \tag{30}$$

$$a = \begin{cases} a_p, & b(n) > 0 \\ a_n, & b(n) \le 0 \end{cases} \tag{31}$$

where the parameter $(a_p, a_n)$ is chosen from {(4,3), (4,1), (2,3), (1,3), (3,3), (1,1)}.

For the convolution operation, we construct 40, 3, and 7 simulated rooms for training, validation, and test respectively. The length and width of these rooms are randomly chosen from [3, 8] m and the height is randomly chosen from [2.5, 4.5] m. The reverberation time $T_{60}$ is randomly chosen from [200, 400] ms. Image method [29] is employed to generate 10 room impulse responses (RIRs) for each room, resulting in 400, 30, and 70 RIRs for training, validation, and test respectively.

The frequency-domain Kalman filter [4] acts as the LAEC to generate the residual echo, and the mean of its echo attenuation on the artificial-echo dataset is about 17.0 dB. To obtain the simulated $s_{\text{AEC}}(n)$, we add both the clean speech signal and the colored noise to the residual echo. The inverse-frequency-power of the colored noise [30] is randomly chosen between 0 and 2. For the training and validation set, the signal-to-echo ratio (SER) (before processing of LAEC) is randomly chosen from $\{-14.2, -16.2, -18.2, -20.2\}$ dB and the colored noise is added with the signal-to-noise ratio (SNR) randomly chosen from {30, 20, 10} dB. For the test set, the SER is $-18.2$ dB and the SNR is 20 dB.

In total, we generate 106,224 segments of speech residual echo and 101,956 segments of music residual echo for training, 1083 segments of speech residual echo and 1083

segments of music residual echo for validation, and 920 segments of speech residual echo and 920 segments of music residual echo for test.

The approach to generate the artificial nonlinear echo is only a rough approximation for simulating the nonlinearity of the loudspeaker. To evaluate the performance of our model in practical applications, we also record echo signals from off-the-shelf loudspeakers using the microphone, AcousticSensing CHZ-221. A pair of EDIFIER R12U (ER) loudspeakers and a pair of LOYFUN LF-501 (LL) loudspeakers are used to record the echo signals in a office with room size 6 m × 6 m × 3.2 m. The recording environment is shown in Fig. 6. For each loudspeaker model, we obtain 10,800 segments of 4-s recorded-echo signals (5400 segments of speech and music respectively) from one loudspeaker for training and 1840 segments (920 segments of speech and music respectively) from another loudspeaker of the same kind for test. The mean of the LAEC's echo attenuation on the recorded-echo dataset is about 24.3 dB. For the training set, the SER of ER echo is randomly chosen from $\{-18.2, -20.2, -22.2, -24.2\}$ dB, the SER of LL echo is randomly chosen from $\{-22.2, -24.2, -26.2, -28.2\}$ dB, and the colored noise is added with the SNR randomly chosen from {30, 20, 10} dB. For the test set, the SER of ER echo is −22.2 dB, the SER of LL echo is −26.2 dB, and the SNR is 20 dB. It should be noted that the recorded-echo training set is only used in the fine-tuning stage.

### 3.2 Experiment configuration

We control the parameter number and processing delay in the time-domain and the TF-domain methods for a fair comparison. For the time-domain method, the number of filters $N$, kernel size $L$, chunk size $K$, and feature dimension $B$ in the encoder are 256, 8, 100, and 128 respectively. For the TF-domain method, the frame length $Q$, the number of down-sampled frequency bins $K'$, and feature dimension $B'$ in the encoder are 400, 99, and 128 respectively. Thus, the tensor of the encoder in the time-domain and the TF-domain methods are of the dimension $T \times 100 \times 128$ and $T \times 99 \times 128$ respectively. The gated recurrent unit (GRU) [31] is used as the RNN layer.

The model is trained by the Adam optimizer [32] for 80 epochs, with each epoch containing 26,556 pairs of training data and each batch containing 8 pairs. The initial learning rate is set to 0.001 and is halved every time the validation loss is not improved in two successive epochs. We apply $l_2$ norm gradient clipping with a maximum of 5. Pytorch is employed for model implementation and four Nvidia GeForce GTX 1080Ti are used for training.

### 3.3 Evaluation metrics

We use three metrics for performance evaluation: the perceptual evaluation of speech quality (PESQ) [33], the signal-to-distortion ratio (SDR) [34, 35], and the short-time objective intelligibility (STOI) [36]. The echo return loss enhancement (ERLE) of the DNN-based methods in single-talk situations has been shown to be of a sufficiently high number in the previous work [19]. In this paper, we pay particular attention to RES performance in the most difficult low-SER double-talk situations, and the PESQ, SDR, and STOI are regarded to be better choices than the ERLE since they can more effectively evaluate the processed near-end speech quality. Furthermore, the

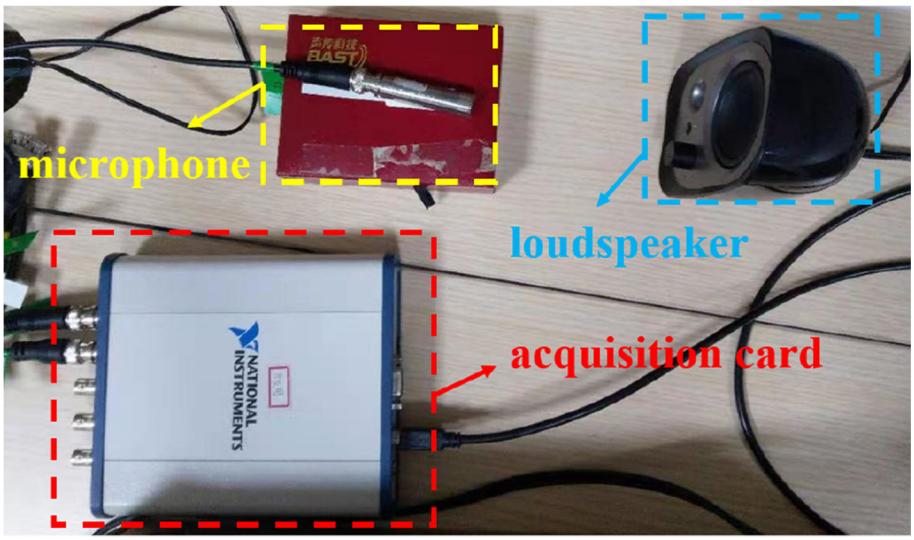


**Fig. 6** The photo of the experiment site where we record the echo signal

desired signal is the near-end speech in most AEC scenarios, while the interference might be either speech from the far end, as in common communication applications, or music played by the smart speakers. Therefore, we use the PESQ instead of the perceptual evaluation of audio quality as an objective metric to measure the quality of the processed near-end speech.

## 4 Results and discussions

### 4.1 Performance comparison

We compare the proposed methods with some typical DNN-based RES methods to validate the efficiency of our model. In the following comparison, we name our proposed methods as DSDPRNN. We further use the suffix "t" and "f" to represent the time-domain and the TF-domain methods respectively and use the suffix "x" and "y" to distinguish between the models in which $x(n)$ or $\hat{y}(n)$ is used as the auxiliary signal. The LSTM-based model (LSTM) [17] and the multi-stream Conv-TasNet model (MSTasNet) are utilized for comparison. The models [14, 15] which have shown significantly inferior performance in our previous work [19] are ignored in this comparison.

The total number of trainable parameters and the multiply-accumulate operations per second (MACCPs) of these models is shown in Table 1. The model size of our proposed methods is only 1/5 of the model size of MSTasNet, and the computation cost is also slightly lower.

The time latency of MSTasNet is set to 410 samples for a fair comparison. The performance in terms of PESQ, SDR, and STOI is shown in Table 2. The DSDPRNN methods outperform the LSTM and the MSTasNet in all artificial-echo conditions, validating that our proposed methods provide an efficient way to exploit the information of dual-stream. For recorded echo, the advantage of the DSDPRNN methods over MSTasNet is less obvious, but their generalization capability in practical applications is still validated. The comparison between the time-domain and the TF-domain methods shows that the former tends to achieve slightly better SDR scores, while the latter has slightly better performance in terms of PESQ and STOI. Furthermore, we observe that the methods with the auxiliary signal $\hat{y}(n)$ achieve better performance in the attenuation of recorded echo, implying their better generalization capability compared with the methods using $x(n)$ as the auxiliary signal.

**Table 1** The total number of trainable parameters and MACCPs of our proposed methods and several typical DNN-based RES methods

| Model | Model size | MACCPs |
|---|---|---|
| LSTM | 2.96M | **0.30G** |
| MSTasNet | 14.8M | 23G |
| DSDPRNN_t | 2.84M | 22G |
| DSDPRNN_f | **2.77M** | 22G |

**Table 2** Performance of our proposed methods and several typical RES methods

| Echo | Model | PESQ | SDR | STOI |
|---|---|---|---|---|
| Artificial speech | LAEC | 1.48 | −2.60 | 0.622 |
| | LSTM | 2.14 | 6.33 | 0.780 |
| | MSTasNet | 2.54 | 11.6 | 0.857 |
| | DSDPRNN_ty | 2.61 | 12.3 | 0.866 |
| | DSDPRNN_tx | 2.66 | **12.8** | 0.876 |
| | DSDPRNN_fy | **2.75** | 12.4 | 0.880 |
| | DSDPRNN_fx | 2.74 | 12.5 | **0.882** |
| Artificial music | LAEC | 1.48 | −2.90 | 0.634 |
| | LSTM | 2.08 | 5.46 | 0.755 |
| | MSTasNet | 2.43 | 10.7 | 0.830 |
| | DSDPRNN_ty | 2.50 | 11.5 | 0.842 |
| | DSDPRNN_tx | 2.61 | **12.6** | **0.865** |
| | DSDPRNN_fy | 2.62 | 11.4 | 0.857 |
| | DSDPRNN_fx | **2.64** | 11.6 | 0.863 |
| ER speech | LAEC | 16.1 | −2.05 | 0.697 |
| | LSTM | 2.13 | 4.85 | 0.799 |
| | MSTasNet | 2.66 | 11.6 | 0.890 |
| | DSDPRNN_ty | 2.68 | **11.7** | 0.892 |
| | DSDPRNN_tx | 2.62 | 11.5 | 0.887 |
| | DSDPRNN_fy | **2.77** | 11.3 | **0.904** |
| | DSDPRNN_fx | 2.66 | 10.6 | 0.895 |
| ER music | LAEC | 1.70 | −1.12 | 0.730 |
| | LSTM | 2.25 | 5.95 | 0.826 |
| | MSTasNet | 2.72 | 12.2 | 0.898 |
| | DSDPRNN_ty | 2.75 | **12.6** | 0.900 |
| | DSDPRNN_tx | 2.68 | 12.3 | 0.897 |
| | DSDPRNN_fy | **2.79** | 11.9 | **0.907** |
| | DSDPRNN_fx | 2.76 | 11.7 | **0.907** |
| LL speech | LAEC | 1.95 | 1.67 | 0.806 |
| | LSTM | 2.55 | 9.23 | 0.884 |
| | MSTasNet | 2.99 | 15.0 | 0.932 |
| | DSDPRNN_ty | 3.00 | 15.6 | 0.932 |
| | DSDPRNN_tx | 2.87 | 14.9 | 0.920 |
| | DSDPRNN_fy | 3.02 | 15.3 | **0.938** |
| | DSDPRNN_fx | **3.04** | **15.7** | **0.938** |
| LL music | LAEC | 1.97 | 2.16 | 0.820 |
| | LSTM | 2.60 | 9.07 | 0.889 |
| | MSTasNet | 3.04 | 15.6 | 0.934 |
| | DSDPRNN_ty | 3.07 | 16.0 | 0.935 |
| | DSDPRNN_tx | 2.89 | 14.8 | 0.921 |
| | DSDPRNN_fy | 3.12 | 15.8 | **0.944** |
| | DSDPRNN_fx | **3.13** | **16.0** | 0.943 |

### 4.2 Fine-tuning for off-the-shelf loudspeakers

Though the proposed methods generalize well to real loudspeakers, better performance can be expected by training on echo recorded from loudspeakers. The well-trained model in the artificial-echo dataset can be regarded as a pre-trained model and then fine-tuned by the recorded-echo dataset in practice. We only test the performance of the DSDPRNN with the auxiliary signal $\hat{y}(n)$. The purpose of the fine-tuning is to improve the performance under limited supplementary training data. We have tried the fine-tuning on the suppression module, but found that the model overfits severely with small amount of recorded data. Thus, we propose two strategies to fine-tune the model by mainly retraining the decoder. (1) Train the decoder module only and freeze the other parameters. (2) Train the decoder and the last DSDPRNN block and freeze the other parameters. We conduct two experiments in the fine-tuning stage for cross validation. In each experiment, we only use 12-h echo signals from one loudspeaker as the training set. The batch size is set to 16 and the exponential-decay strategy is used to halve the learning rate every 1350 steps. The fine-tuning stage uses two Nvidia GeForce GTX 1080Ti and takes only about 3 h for training since the partly frozen parameters reduce the computational complexity for training and the size of the recorded training data is far below the size of the artificial echo. We use "Time" and "TF" to distinguish the time-domain and the TF-domain DSDPRNN methods and use the suffix "1", "2" to represent the models using the above two fine-tuning strategies respectively. The performance of the pre-trained model is presented with no suffix as benchmark. Compared to strategy 2, the training time in the fine-tuning stage of strategy 1 decreases by 14% and the memory cost is reduced by half.

The performances of the proposed methods after fine-tuning with the ER echo dataset and the LL echo dataset are shown in Tables 3 and 4 respectively. In artificial-echo conditions, the performance degrades slightly after fine-tuning, and similar results are observed using both the fine-tuning strategies. The test results of the model fine-tuned using the recorded training dataset from the same kind of loudspeaker are highlighted by blue font. The efficacy of both fine-tuning strategies can be seen, and strategy 2 has significantly better performance when the model is fine-tuned by the training dataset from the same kind of loudspeaker. It also should be noted that the performance improves slightly even when the model is fine-tuned with training data from different loudspeakers, indicating the generalization capability of the fine-tuning method. Considering that only a very limited data is required in the fine-tuning stage, this scheme is easy to be applied to any off-the-shelf loudspeakers.

**Table 3** Performance of the pre-trained model and the fine-tuned models with ER recorded echo

| Echo | Model | PESQ | SDR | STOI |
|---|---|---|---|---|
| Artificial speech | LAEC | 1.48 | −2.60 | 0.622 |
| | Time | 2.61 | 12.3 | 0.866 |
| | Time_1 | 2.56 | 12.2 | 0.865 |
| | Time_2 | 2.57 | 12.1 | 0.864 |
| | TF | **2.75** | **12.4** | **0.880** |
| | TF_1 | 2.70 | **12.4** | 0.875 |
| | TF_2 | 2.69 | 12.3 | 0.875 |
| Artificial music | LAEC | 1.48 | −2.90 | 0.634 |
| | Time | 2.50 | **11.5** | 0.842 |
| | Time_1 | 2.44 | 11.4 | 0.841 |
| | Time_2 | 2.46 | 11.3 | 0.841 |
| | TF | **2.62** | 11.4 | **0.857** |
| | TF_1 | 2.58 | 11.3 | 0.853 |
| | TF_2 | 2.57 | 11.3 | 0.852 |
| ER speech | LAEC | 1.61 | −2.05 | 0.697 |
| | Time | 2.68 | 11.7 | 0.892 |
| | Time_1 | 2.70 | 12.0 | 0.894 |
| | Time_2 | 2.75 | **12.5** | 0.899 |
| | TF | 2.77 | 11.3 | 0.904 |
| | TF_1 | 2.80 | 11.9 | 0.905 |
| | TF_2 | **2.88** | 12.4 | **0.912** |
| ER music | LAEC | 1.70 | −1.12 | 0.730 |
| | Time | 2.75 | 12.6 | 0.900 |
| | Time_1 | 2.76 | 12.8 | 0.901 |
| | Time_2 | 2.80 | **13.0** | 0.906 |
| | TF | 2.79 | 11.9 | 0.907 |
| | TF_1 | 2.83 | 12.3 | 0.908 |
| | TF_2 | **2.91** | 12.6 | **0.914** |
| LL speech | LAEC | 1.95 | 1.67 | 0.806 |
| | Time | 3.00 | 15.6 | 0.932 |
| | Time_1 | 3.00 | 15.8 | 0.933 |
| | Time_2 | 3.03 | **16.1** | 0.935 |
| | TF | 3.02 | 15.3 | 0.938 |
| | TF_1 | 3.08 | 15.8 | 0.939 |
| | TF_2 | **3.13** | **16.1** | **0.943** |
| LL music | LAEC | 1.97 | 2.16 | 0.820 |
| | Time | 3.07 | 16.0 | 0.935 |
| | Time_1 | 3.03 | 16.1 | 0.936 |
| | Time_2 | 3.04 | **16.2** | 0.937 |
| | TF | 3.12 | 15.8 | 0.944 |
| | TF_1 | 3.17 | 16.1 | 0.944 |
| | TF_2 | **3.18** | **16.2** | **0.946** |

## 5 Conclusion

In this paper, we propose efficient RES methods in both the time domain and the TF domain on the modification of DPRNN. We adopt the residual signal and the auxiliary

**Table 4** Performance of the pre-trained model and the fine-tuned models with LL recorded echo

| Echo | Model | PESQ | SDR | STOI |
|---|---|---|---|---|
| Artificial speech | LAEC | 1.48 | −2.60 | 0.622 |
| | Time | 2.61 | 12.3 | 0.866 |
| | Time_1 | 2.59 | 12.1 | 0.866 |
| | Time_2 | 2.60 | 12.0 | 0.864 |
| | TF | **2.75** | **12.4** | **0.880** |
| | TF_1 | 2.73 | **12.4** | 0.879 |
| | TF_2 | 2.70 | 12.2 | 0.875 |
| Artificial music | LAEC | 1.48 | −2.90 | 0.634 |
| | Time | 2.50 | **11.5** | 0.842 |
| | Time_1 | 2.47 | 11.4 | 0.842 |
| | Time_2 | 2.47 | 11.2 | 0.840 |
| | TF | **2.62** | 11.4 | **0.857** |
| | TF_1 | 2.61 | 11.4 | 0.856 |
| | TF_2 | 2.58 | 11.2 | 0.852 |
| ER speech | LAEC | 1.61 | −2.05 | 0.697 |
| | Time | 2.68 | 11.7 | 0.892 |
| | Time_1 | 2.70 | **11.9** | 0.894 |
| | Time_2 | 2.72 | 11.8 | 0.894 |
| | TF | 2.77 | 11.3 | 0.904 |
| | TF_1 | 2.81 | 11.7 | 0.906 |
| | TF_2 | **2.83** | 11.6 | **0.908** |
| ER music | LAEC | 1.70 | −1.12 | 0.730 |
| | Time | 2.75 | 12.6 | 0.900 |
| | Time_1 | 2.75 | 12.7 | 0.900 |
| | Time_2 | 2.77 | **12.8** | 0.902 |
| | TF | 2.79 | 11.9 | 0.907 |
| | TF_1 | 2.83 | 12.2 | 0.909 |
| | TF_2 | **2.86** | 12.3 | **0.911** |
| LL speech | LAEC | 1.95 | 1.67 | 0.806 |
| | Time | 3.00 | 15.6 | 0.932 |
| | Time_1 | 3.02 | 15.9 | 0.934 |
| | Time_2 | 3.07 | 16.3 | 0.936 |
| | TF | 3.02 | 15.3 | 0.938 |
| | TF_1 | 3.08 | 15.8 | 0.940 |
| | TF_2 | **3.22** | **16.5** | **0.947** |
| LL music | LAEC | 1.97 | 2.16 | 0.820 |
| | Time | 3.07 | 16.0 | 0.935 |
| | Time_1 | 3.07 | 16.2 | 0.936 |
| | Time_2 | 3.10 | 16.4 | 0.939 |
| | TF | 3.12 | 15.8 | 0.944 |
| | TF_1 | 3.17 | 16.1 | 0.945 |
| | TF_2 | **3.24** | **16.5** | **0.948** |

signal extracted from the LAEC system to form dual-stream for the DPRNN. Experiments validate the efficacy of the proposed methods in double-talk situations compared with several typical RES methods. Furthermore, we propose an efficient and applicable way to improve the performance on off-the-shelf loudspeakers by regarding the well-trained model on artificial-echo dataset as a pre-trained model, and fine-tuning it on recorded-echo dataset. Two fine-tuning strategies are evaluated in experiments, showing that the fine-tuning strategy of training the decoder and the last DSSPRNN block achieves more effective echo suppression on the recorded-echo dataset.

#### Abbreviations

ASR: Automatic speech recognition; Conv-TasNet: Fully convolutional time-domain audio separation network; DNN: Deep neutral network; DPRNN: Dual-path recurrent neural network; DSDPRNN: Dual-stream dual-path recurrent neural network; DSRNN: Dual-stream recurrent neural network; ER: Edifier R12U; ERLE: Echo return loss enhancement; FCN: Fully connected network; FDKF: Frequency-domain adaptive Kalman filter; GRU: Gated recurrent unit; LAEC: Linear acoustic echo cancelation; LL: Loyfun LF-501; LMS: Least mean square; LSTM: Long short-term memory; MACCPs: Multiply-accumulate operations per second; PESQ: Perceptual evaluation of speech quality; RES: Residual echo suppression; RIR: Room impulse response; RNN: Recurrent neural network; SDR: Signal-to-distortion ratio; SER: Signal-to-echo ratio; SNR: Signal-to-noise ratio; SOTA: State-of-the-art; STFT: Short-time Fourier transform; STOI: Short-time objective intelligibility; TF: Time-frequency

#### Acknowledgements

This work was supported by the National Science Foundation with grant no. 11874219.

#### Authors' contributions

H.C., G.C., and J.L. analyzed the DNN-based RES method. H.C. implemented the method. H.C., K.C., and J.L. conducted the experiments. H.C. and J.L. drafted the manuscript. All authors have reviewed the results and the final manuscript. The authors read and approved the final manuscript.

#### Availability of data and materials

The source codes of the network are released on https://github.com/Mo-yun/DSDPRNN, and exemplary audio samples are available online at https://github.com/Mo-yun/dsdprnn-samples. Further materials are also available from the corresponding author upon request.

## Declarations

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

[1]Key Laboratory of Modern Acoustics, Nanjing University, 210093 Nanjing, China. [2]NJU-Horizon Intelligent Audio Lab, Horizon Robotics, 100094 Beijing, China. [3]Nanjing Institute of Advanced Artificial Intelligence, 210014 Nanjing, China.

Received: 5 April 2021 Accepted: 18 August 2021
Published online: 07 September 2021

#### References

1. E. Hänsler, G. U. Schmidt, Hands-free telephones–joint control of echo cancellation and postfiltering. Signal Process. **80**(11), 2295–2305 (2000)
2. S. S. Haykin, *Adaptive Filter Theory*. (Prentice Hall, New Jersey, 2002)
3. F. Albu, H. K. Kwan, in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*. Combined echo and noise cancellation based on Gauss-Seidel pseudo affine projection algorithm, vol. 3, (2004), p. 505
4. G. Enzner, P. Vary, Frequency-domain adaptive kalman filter for acoustic echo control in hands-free telephones. Signal Process. **86**(6), 1140–1156 (2006)
5. F. Yang, G. Enzner, J. Yang, Frequency-domain adaptive Kalman filter with fast recovery of abrupt echo-path changes. IEEE Signal Process. Lett. **24**(12), 1778–1782 (2017)

6. W. Fan, K. Chen, J. Lu, J. Tao, Effective improvement of under-modeling frequency-domain Kalman filter. IEEE Signal Process. Lett. **26**(2), 342–346 (2019)
7. A. N. Birkett, R. A. Goubran, in *Proceedings of 1995 Workshop on Applications of Signal Processing to Audio and Accoustics*. Limitations of handsfree acoustic echo cancellers due to nonlinear loudspeaker distortion and enclosure vibration effects, (1995), pp. 103–106
8. S. Gustafsson, R. Martin, P. Vary, Combined acoustic echo control and noise reduction for hands-free telephony. Signal Process. **64**(1), 21–32 (1998)
9. E. A. P. Habets, S. Gannot, I. Cohen, P. C. W. Sommen, Joint dereverberation and residual echo suppression of speech signals in noisy environments. IEEE Trans. Audio Speech Lang. Process. **16**(8), 1433–1451 (2008)
10. N. K. Desiraju, S. Doclo, M. Buck, T. Wolff, Online estimation of reverberation parameters for late residual echo suppression. IEEE Trans. Audio Speech Lang. Process. **28**, 77–91 (2020)
11. S. Gustafsson, R. Martin, P. Jax, P. Vary, A psychoacoustic approach to combined acoustic echo cancellation and noise reduction. IEEE Trans. Speech Audio Process. **10**(5), 245–256 (2002)
12. A. S. Chhetri, A. C. Surendran, J. W. Stokes, J. C. Platt, in *Proc. IWAENC*. Regression-based residual acoustic echo suppression, vol. 5, (2005)
13. M. L. Valero, E. Mabande, E. A. P. Habets, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Signal-based late residual echo spectral variance estimation, (2014), pp. 5914–5918
14. G. Carbajal, R. Serizel, E. Vincent, E. Humbert, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Multiple-input neural network-based residual echo suppression, (2018), pp. 231–235
15. H. Zhang, D. Wang, Deep learning for acoustic echo cancellation in noisy and double-talk scenarios. Training. **161**(2), 322 (2018)
16. F. Kuech, W. Kellermann, in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*. Nonlinear residual echo suppression using a power filter model of the acoustic echo path, vol. 1, (2007), pp. 73–76
17. C. Zhang, X. Zhang, in *Proc. Interspeech*. A robust and cascaded acoustic echo cancellation based on deep learning, (2020), pp. 3940–3944
18. Y. Luo, N. Mesgarani, Conv-tasnet: surpassing ideal time-frequency magnitude masking for speech separation. IEEE/ACM Trans Audio Speech Lang. Process. **27**(8), 1256–1266 (2019)
19. H. Chen, T. Xiang, K. Chen, J. Lu, in *Proc. Interspeech*. Nonlinear residual echo suppression based on multi-stream Conv-TasNET, (2020), pp. 3959–3963
20. Y. Luo, Z. Chen, T. Yoshioka, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation (IEEE, 2020), pp. 46–50
21. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE International Conference on Computer Vision*. Delving deep into rectifiers: surpassing human-level performance on imagenet classification, (2015), pp. 1026–1034
22. A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, H. A. PHAN, Tensor decompositions for signal processing applications: from two-way to multiway component analysis. IEEE Signal Process. Mag. **32**(2), 145–163 (2015). https://doi.org/10.1109/MSP.2013.2297439
23. D. Yin, C. Luo, Z. Xiong, W. Zeng, Phasen: a phase-and-harmonics-aware speech enhancement network. arXiv preprint arXiv:1911.04697 (2019)
24. Y. Wu, K. He, in *Proceedings of the European Conference on Computer Vision (ECCV)*. Group normalization, (2018), pp. 3–19
25. V. Panayotov, G. Chen, D. Povey, S. Khudanpur, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Librispeech: an ASR corpus based on public domain audio books (IEEE, 2015), pp. 5206–5210
26. D. Snyder, G. Chen, D. Povey, Musan: a music, speech, and noise corpus. arXiv preprint arXiv:1510.08484 (2015)
27. S. Malik, G. Enzner, State-space frequency-domain adaptive filtering for nonlinear acoustic echo cancellation. IEEE Trans. Audio Speech Lang. Process. **20**(7), 2065–2079 (2012). https://doi.org/10.1109/TASL.2012.2196512
28. D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, A. Uncini, in *2017 25th European Signal Processing Conference (EUSIPCO)*. Full proportionate functional link adaptive filters for nonlinear acoustic echo cancellation, (2017), pp. 1145–1149
29. E. A. Lehmann, A. M. Johansson, Diffuse reverberation model for efficient image-source simulation of room impulse responses. IEEE Trans. Audio Speech Lang. Process. **18**(6), 1429–1439 (2009)
30. N. J. Kasdin, Discrete simulation of colored noise and stochastic processes and 1/f/sup/spl alpha//power law noise generation. Proc. IEEE. **83**(5), 802–827 (1995)
31. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
32. D. P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
33. A. W. Rix, J. G. Beerends, M. P. Hollier, A. P. Hekstra, in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs, vol. 2, (2001), pp. 749–7522
34. E. Vincent, R. Gribonval, C. Févotte, Performance measurement in blind audio source separation. IEEE Trans. Audio Speech Lang. Process. **14**(4), 1462–1469 (2006)
35. C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, C. C. Raffel, in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. mir_eval: a transparent implementation of common MIR metrics (Citeseer, 2014)
36. C. H. Taal, R. C. Hendriks, R. Heusdens, J. Jensen, in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. A short-time objective intelligibility measure for time-frequency weighted noisy speech, (2010), pp. 4214–4217


## Publisher's Note