




# Speech Enhancement of Complex Convolutional Recurrent Network with Attention

Jiangjiao Zeng<sup>1</sup> · Lidong Yang<sup>1</sup> 

Received: 27 January 2022 / Revised: 7 August 2022 / Accepted: 8 August 2022 /

Published online: 30 August 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Speech enhancement aims to separate pure speech from noisy speech, to improve speech quality and intelligibility. A complex convolutional recurrent network with a parameter-free attention module is proposed to improve the effect of speech enhancement. First, the feature information is enhanced by improving the convolutional layer of the encoding layer and the decoding layer. Then, the redundant information is suppressed by adding a parameter-free attention module to extract features that are more effective for the speech enhancement task, and the middle layer is selected for the bidirectional gated recurrent unit. Compared with the best of several baseline models, in the Voice Bank + DEMAND dataset, Perceptual Evaluation of Speech Quality (PESQ) increased by 0.17 (6.23%), MOS predictor of intrusiveness of background noise (CBAK) increased by 0.14 (4.34%), (MOS predictor of overall processed speech quality) COVL increased by 0.40 (12.42%), and (MOS predictor of speech distortion) CSIG index increased by 0.57 (15.28%). Experimental results show that the proposed approach has higher theoretical significance and practical value for actual speech enhancement.

**Keywords** Speech enhancement · Parameter-free attention module · Convolutional recurrent network · Bidirectional gated recurrent unit

---

Jiangjiao Zeng and Lidong Yang have contributed equally to this work.

---

✉ Lidong Yang  
yld\_nkd@imust.edu.cn

Jiangjiao Zeng  
18797919556@163.com

<sup>1</sup> School of Information Engineering, Inner Mongolia University Of Science and Technology, 7 ardin street, BaoTou 014010, Inner Mongolia, China

## 1 Introduction

Speech enhancement technology is essential to improve the clarity and quality of noisy speech signals. Traditional speech enhancement techniques include spectral subtraction [2, 7], Wiener filtering [14], minimum mean square error (MMSE) estimator [6], and optimized logarithmic spectral amplitude speech estimator [5]. Although these traditional methods based on the time-frequency domain have achieved better performance in a stationary noise environment, the effect of processing nonstationary noise is poor in most scenarios.

Wang et al. initially proposed a supervised speech enhancement algorithm based on deep learning in 2012 [29]. This algorithm achieves more ideal results under nonstationary noise conditions than the traditional methods. In 2016, Park et al. proposed a fully convolutional network to achieve speech enhancement tasks [15]. The enhancement effect achieved by this method is better than the previous method. In addition, considering that the speech signal has timing characteristics, recurrent neural network (RNN) has been introduced into the field of speech enhancement [17]. To solve the problem of gradient disappearance or gradient explosion [1], which is common in RNN, long short-term memory (LSTM) is widely used [8, 28]. Lei Sun et al. proposed a method for speech enhancement using LSTM-RNN in a 2017 study [20]. The results show that LSTM-RNN has a better effect on improving the speech quality and intelligibility of speech enhancement than DNN. In 2021, Kumar, B proposed the CS-based technique using generalized orthogonal matching pursuit algorithm yields better performance than the other recovery algorithms in terms of speech quality and distortion [13].

U-Net is a new type of network model first used for image segmentation [16]. It is valued in the field of speech signal processing because of its excellent performance in image segmentation tasks. Wave-U-Net was proposed by Daniel et al. [19] and used in the task of sound source separation. The network adopts a fully convolutional U-Net structure and is different from the model based on time–frequency characteristics. It can directly convolve the speech signal in the time domain in 1D without time–frequency transformation and separate the voice of music. The task has achieved the best performance so far, showing strong feature extraction and signal recovery capabilities. Tan et al. proposed the convolutional recurrent neural network (CRNN) in 2018 [21], using the LSTM module as the middle layer of the U-Net model to calculate the timing-related information of the speech signal. The multilayer LSTM improves the quality and intelligibility of enhanced speech while reducing the training parameters.

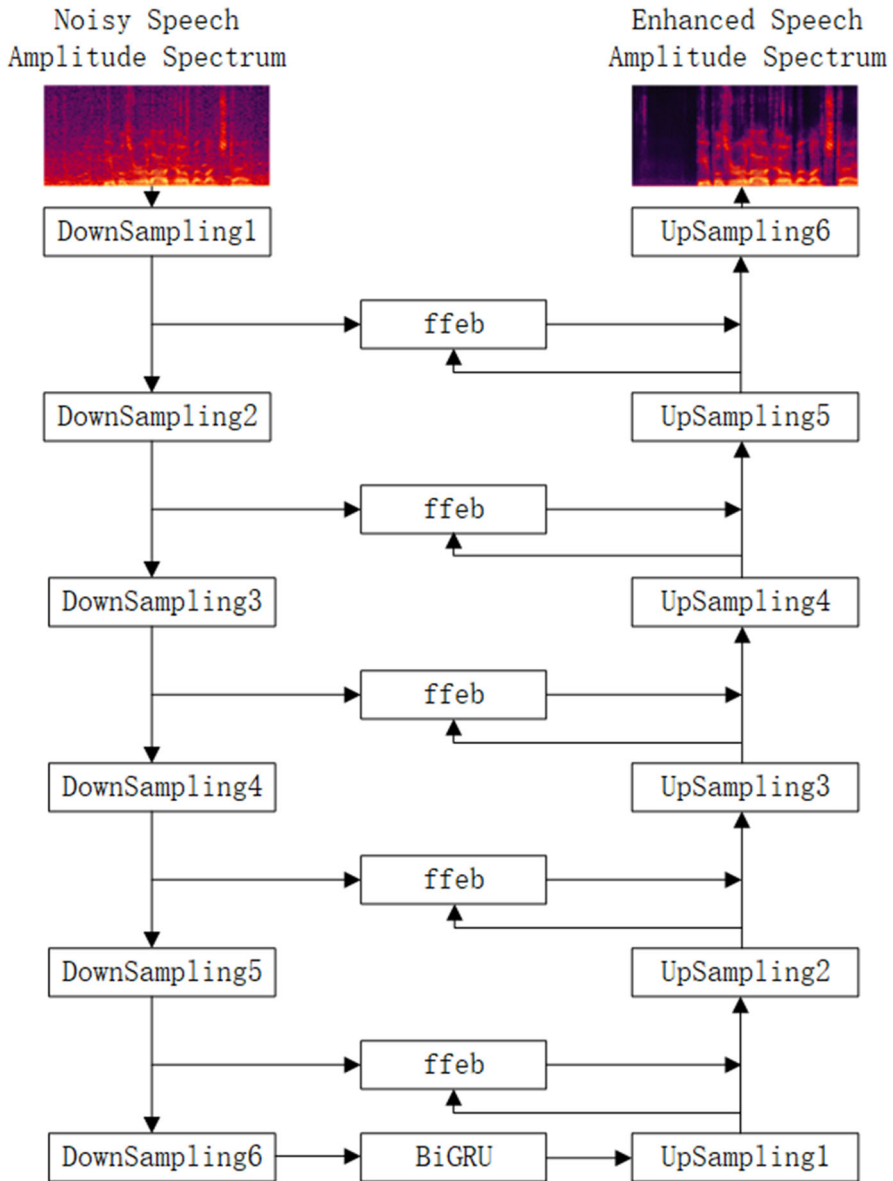
To further improve the effect of speech enhancement, the approach is improved based on the convolutional recurrent network. Inspired by Tian et al. [25], the feature space is enriched by improving the convolutional layer and the inverted convolutional layer in the encoder and decoder, and the expression ability of the model is improved. Key features useful for enhancement tasks are extracted by adding an attention mechanism to suppress redundant features to improve the accuracy of the model. Finally, a soft pool [18] is used to replace the pooling layer in the network to solve the problem of the loss of important features in the traditional pooling layer or the significant reduction of the overall feature strength.

## 2 Model of the Study

The network model is shown in Fig. 1. The model includes an encoder layer, middle layer, decoder layer, and skip connection with feature enhancement module. The encoder layer contains six downsampling layers, and the input is a noisy speech amplitude spectrum, that is, a 2D feature map with time and frequency as the scale. The six convolutional layers extract features from the signal layer by layer, and the number of convolution kernels used in each layer is set to 32, 64, 128, 256, 256, and 256, and the time–frequency feature map output by each layer is on time, and the frequency dimension is reduced. During the experiment, this paper tries to increase the number of downsampling layers. The increase in the number of downsampling layers does not bring about an improvement in the speech enhancement effect, or even a decline. Adding too many downsampling layers at the same time will increase the complexity of the model. Speech enhancement works best when downsampling is set to six layers. The middle layer uses a bidirectional gated recurrent unit (BiGRU) [4]. The multidimensional feature tensor (Tensor) output by the encoder layer needs to be reduced by the dimensionality adjustment (Reshape) operation because BiGRU can only process 2D feature maps. The requirements of BiGRU enable the network to complete the learning of signal timing characteristics, and at the same time, reconstructing the tensor output by BiGRU is necessary to increase the dimension. The decoder layer can be regarded as the inverse process of the encoder layer, which includes six upsampling layers. Each upsampling layer expands the feature map by moving the convolution kernel, in which the convolution is reversed in a specific step. The output of the decoder layer is the enhanced speech amplitude spectrum, and the phase of the noisy speech is used to reconstruct the waveform signal. The network has a feature enhancement module skip connection setting. It inputs the output of the downsampling layer and the input of the corresponding upsampling layer into the feature-enhancing module at the same time, and then splices it with the output of the corresponding upsampling module, thereby combining the feature map is doubled. This operation is conducive to the recovery of the pair number of the decoding layer.

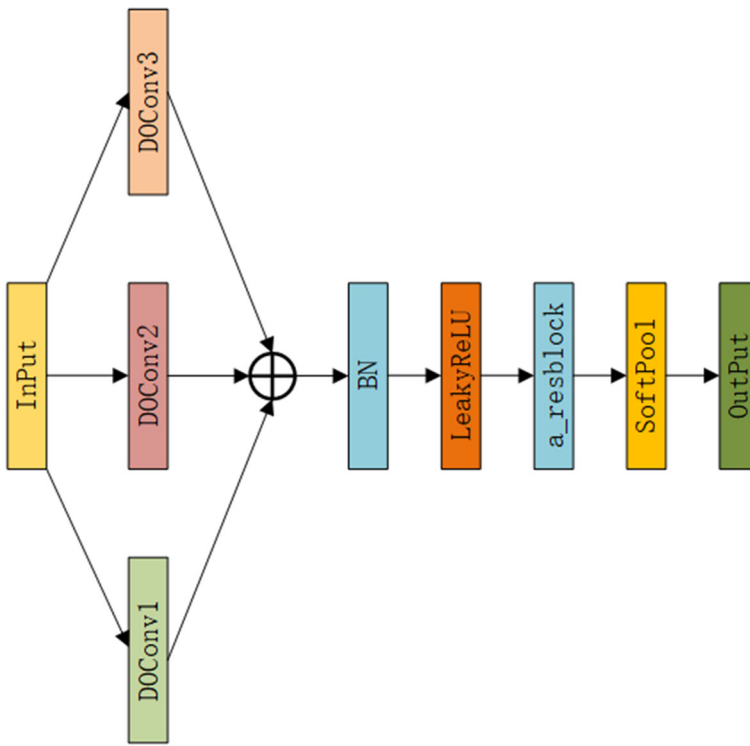
### 2.1 Downsampling Module

As shown in Fig. 2, the downsampling layer consists of three deep over-parameterized convolutional layers (DO-Conv) [3], batch normalization layer (BN), activation function, residual learning with attention mechanism, and composition of the soft pool. The deep over parametric convolutional layer is composed of deep convolution and traditional convolutional layers. First, the convolution kernel is deeply convolved to form a new convolution kernel, and then, the features are traditionally convolved. They are DO-Conv1, DO-Conv2, and D-OConv3, and their convolution kernels are set to  $1*3$ ,  $3*3$ , and  $3*1$ , respectively. D-OConv1 and DO-Conv3 are used to enhance the horizontal and vertical features, and then, residual learning is used to influence DO-Conv2 with the obtained information, the feature space is enriched, and the expression ability of the model is promoted. The role of BN is to maintain the output data of the convolutional layer and to satisfy the independent and identical distribution assumption during



**Fig. 1** This is the model of this study. The model includes downsampling layers, intermediate layers, upsampling layers, and skip connections with feature enhancement modules

the neural network training process. The batch normalization layer is conducive to the convergence of the network error function and speeds up the training efficiency. The activation function is LeakyReLU. In the last layer, a residual learning module with an attention mechanism is used.



**Fig. 2** This is downsampling module. The downsampling layer consists of DO-Conv, BN layer, LeakyRelu layer, a\_resblock, and soft pool

This work is inspired by the deep residual shrinkage network [33], combines the attention mechanism with residual learning, and replaces the attention mechanism in the basic module of the deep residual shrinkage network with a parameter-free attention module [31]. Among them, the parameter-free attention module is derived from the basic theories of neuroscience. In neuroscience, information-rich neurons usually exhibit different firing patterns from peripheral neurons. Moreover, activating neurons usually inhibits peripheral neurons, that is, spatial inhibition [30]. Neurons with spatial inhibitory effects should be given higher importance. The importance of neurons is determined by defining the energy function as follows:

$$X_n(e^{jw}) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-jwn} \quad (1)$$

Among them,  $\hat{t} = w_t t + b_t$  and  $\hat{x}_i = w_i x_i + b_i$  are the linear transformations of  $t$  and  $x_i$ , where  $t$  and  $x_i$  are the target neuron and the other neurons in a single channel of the input feature, respectively.  $i$  is the index in the spatial dimension, and each channel has neurons. All values in Eq. (1) are scalars. When  $\hat{t}$  is equal to  $y_t$  and all other  $\hat{x}_i$  is  $y_0$ , Eq. (1) reaches the minimum value, where  $y_t$  and  $y_0$  are two different values. By minimizing this equation, Eq. (1) is equivalent to finding the linear separability

between the target neuron and all other neurons in the same channel. For simplicity, a binary label is used, and a regular term is added. The minimum energy equation can be obtained by solving as the following:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (2)$$

In Eq. (2), lower energy indicates greater difference between neuron  $t$  and surrounding neurons and higher importance. Therefore, the importance of neurons can be obtained  $1/e_t^*$ .

The equation derives the energy function and taps the importance of neurons. In the attention mechanism, the features are enhanced, as follows:

$$\tilde{X} = \text{sigmoid}\left(\frac{1}{E}\right) \odot X \quad (3)$$

Among them,  $E$  groups all  $e_t^*$  across the channel and spatial dimensions. The sigmoid function is added to limit the larger value in 1. This function does not affect the relative importance of each neuron. In summary, in addition to the calculation of the channel average and variance, all calculations in the module are based on element calculations and do not involve structural adjustments. This finding is very different from the previous attention mechanisms.

Usually, convolutional neural networks (CNNs) use the pool operation to reduce the size of the feature map. This process is essential to achieve the local space invariance and increase the receptive field of subsequent convolution. Therefore, the pooling operation should minimize the loss of information in the feature map and limit the calculation and memory overhead. The risk of losing important information is introduced to avoid the risk of losing most activations brought about by the maximum pooling. At the same time, to avoid the equal contribution of the activation values in the average pooling, the overall regional feature strength is significantly reduced. SoftPool assigns a weight to each activation value, and the weight is used as a nonlinear transformation together with the corresponding activation value. A higher activation value is more dominant than a lower activation value. Its weight is calculated as the ratio of the natural index of the activation to the sum of the natural indices of all activations in the neighborhood  $R$ , as follows:

$$\omega_i = \frac{e^{a_i}}{\sum_{j \in R} e^{a_j}} \quad (4)$$

The weight in Eq. (4) is the corresponding activation weight. To ensure that a larger activation value has a greater impact on the output, this method uses the natural index  $e$  as the bottom.

The output value of the SoftPool operation is obtained by summing all the weighted activation criteria in the kernel neighborhood, as follows:

$$\tilde{a} = \sum_{i \in R} \omega_i * a_i \quad (5)$$

Highlighting activations with greater effects is a more balanced approach than simply selecting the maximum value because most pooling operations are performed in a high-dimensional feature space.

## 2.2 Upsampling Module

The upsampling module in the decoder has a similar structure to the downsampling module, consisting of a transposed convolutional layer composed of three transposed convolutional layers, a batch normalization layer, a LeakyReLU activation layer, and a dropout layer. Transposed convolution can be regarded as the inverse process of traditional convolution. The size settings of the convolution kernels of the three transposed convolutions are consistent with the settings in the downsampling. The difference is that the upsampling module realizes the expansion of the 2D time–frequency feature map by changing the size of the convolution kernel step size of the transposed convolution, and the final output is the 2D time–frequency feature map with the same size as the network input.

A dropout layer is added after the activation layer to solve the problem to avoid the over-fitting of the model and improve the robustness of the network model. Two neurons do not necessarily appear in a dropout network every time because of the dropout layer. As a result, the update of weights no longer depends on the joint action of implicit nodes with fixed relationships, thereby preventing some features from being effective only under other specific features. The network is forced to learn more robust features, which also exist in random subsets of other neurons. From this perspective, dropout is slightly similar to L1 and L2 regular, reducing the weight that causes the network to become more robust to the loss of specific neuron connections.

Finally, this article adds an output layer after the sixth upsampling layer, which is composed of a transposed convolution layer. The size of the convolution kernel is set to  $3 \times 3$ , the step size is set to  $1 \times 1$ , and the activation function uses the sigmoid function.

## 2.3 BiGRU

Ordinary RNNs are prone to the problem of gradient disappearance in the process of training time-series data. The proposal of LSTM and gated recurrent units solves this problem. The performance of GRU and LSTM is similar, but it has a more concise internal structure. The parameters of the network model can be reduced, and the ability to prevent overfitting is improved. LSTM can increase or remove the ability of data information to its memory space through the “gate” structure, and the “gate” structure can also selectively allow information through. LSTM controls the input value, memory value, and output value by three gate functions, namely input gate, forget gate and output gate. GRU only has updated gates and reset gates, and the structure is more concise, as shown in Fig. 3. The calculation method of the GRU structure is shown in Eqs. (6–9), as follows:

$$z = \delta(U^z x_t + W^z h_{t-1}) \quad (6)$$

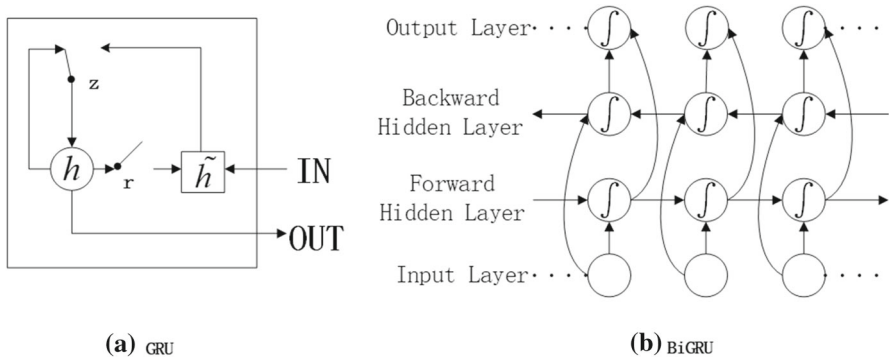


Fig. 3 This is GRU and BiGRU structure diagram

$$r = \delta(U^r x_t + W^r h_{t-1}) \tag{7}$$

$$s = \tanh(U^s x_t + W^h (h_{t-1} \otimes r)) \tag{8}$$

$$h_t = (1 - z) \otimes s + z \otimes h_{t-1} \tag{9}$$

Among them,  $z$  is the update gate,  $r$  is the reset gate,  $\delta$  is the activation function,  $\otimes$  is the vector corresponding element multiplication operation,  $s$  is the current hidden state, and  $h_t$  is the final output state.

In the original RNN network, the state transmission is the unidirectional transmission from front to back, and the bidirectional GRU network structure enables the model to obtain forward dependency information and reverse dependency information. It is up and down by two GRU networks. Stacking structure, the output is jointly determined by two GRU networks, and its structure is shown in Fig. 3.

### 2.4 Feature Fusion Module

A feature fusion module is used to enhance and compensate for the loss of information in the process of downsampling to upsampling [25]. Feature fusion modules include Conv2+RELU, Conv2, and RELU. Among them, Conv2+RELU indicates that Conv2 is tightly connected to RELU.

The realization of the feature fusion module is divided into two stages. The first stage provides complementary information by learning the output of the features by the corresponding upsampling layer and downsampling layer. Specifically, Conv2+RELU and Conv2 are used in the corresponding upsampling and downsampling, respectively, where their input and output channel number is 64. The size of the filter is  $3 \times 3$ . RELU is used to perform dual-path fusion on the extracted features, and the nonlinear transformation is performed on the extracted features. The entire process is expressed in Eq. (10), as follows:

$$O_{\text{ffb}} = R(C(R(C(O_d))) + C(R(C(O_u)))) \tag{10}$$



In Eq. (10),  $O_{\text{feb}}$  represents the output of the first stage of the feature enhancement module,  $R$  represents the RELU layer,  $C$  represents the convolutional layer,  $O_d$  and  $O_u$  are the output of downsampling and upsampling, respectively.

The second stage is used to prevent excessive feature enhancement problems to obtain more robust features. It has two types of operations, namely Conv2+RELU and Conv2. First, the two layers use Conv2+RELU with a size of  $64 \times 3 \times 3 \times 64$ , where the number of input and output channels is 64, and the size of the Conv2 convolution kernel is  $3 \times 3$ . The last layer only uses Conv2 with a size of  $64 \times 3 \times 3 \times 3$ , where the number of input and output channels is 64 and 3, respectively. This process can be expressed in Eq. (11), as follows:

$$O = C(R(C(R(C(O_{\text{feb}})))))) \quad (11)$$

### 3 Experimental Setup

#### 3.1 Data Set

The data set used in this article was published by Valentini et al. [26]. They are widely used in speech enhancement research, including clean speech data and noisy speech data with a sampling frequency of 48 kHz, including different speakers and various types of noise. These clean speech data are recordings from different text paragraph sentences. Thirty English speakers were selected from the Voice Bank corpus [27], including males and females with different accents, of which 28 and two speakers were assigned to the training set and test set, respectively. The test set consists of 20 different noise conditions. Five types of noise are derived from the DEMAND database [24], resulting in 824 test items, and each test speaker has approximately 20 different sentences under different conditions. In the process of calculating the amplitude spectrum from the speech waveform, short-time Fourier transform (STFT) should be performed on the speech signal. To this end, the speech should be framed and windowed to obtain a speech spectrum with a specific frequency resolution. In the experiment, the sampling rate of the corpus is all downsampled from 48 kHz to 16 kHz, the speech frame length is set to 32 ms, that is, 512 sampling points, the frame shift is set to 10 ms and a Hamming window is added to reduce the spectrum leakage. After STFT is performed frame by frame, the amplitude spectrum should be considered. The obtained STFT amplitude spectrum is used as the input feature of noisy speech and the training target of pure speech.

#### 3.2 Experimental Design

The training models in this article are built using TensorFlow. The network iterative training process requires the use of a loss function to calculate the error between the input features of the network and the label. The error updates the weights of all nodes in each layer of the network through reverse transmission and finally completes the learning of the feature-to-label mapping relationship by selecting a suitable optimizer

**Table 1** Model and baseline evaluation results

Method	PESQ	CBAK	COVL	CSIG	STOI
Noisy	1.97	2.44	2.63	3.35	0.91
Wiener	2.22	2.68	2.67	3.23	–
SEGAN	2.16	2.94	2.80	3.48	–
U-Net	2.48	3.21	3.05	3.65	–
CRN	2.61	3.26	3.17	3.78	0.94
GCRN	2.51	3.24	3.09	3.71	0.94
DCCRN-E	2.73	3.22	3.22	3.73	0.941
Our-SIMA	<b>2.90</b>	<b>3.36</b>	<b>3.60</b>	<b>4.30</b>	<b>0.95</b>

Bold indicates the best results

at a specific learning rate to gradually reduce the error in a gradient descent manner. Here, this article sets the batch size of each batch of corpus entering the network to 64, uses the mean square error (MSE) function and uses the Adam optimizer [12] to optimize the network parameters. The Adam parameter is set to B1=0.5 and B2=0.9, and the learning rate is set to 1e-4. This article uses the same training set to train the baseline model and the model in this article. The network convergence indicates that the error function drops to basic stability and then stops training; it uses the test set corpus to test the enhancement effect of the model.

The baseline model of this experiment is set apart from the Wiener filtering method, Speech Enhancement Generative Adversarial Network (SEGAN), U-Net method, there are three recently proposed, namely complex spectral mapping convolution recurrent network (CRN) [22] proposed in 2019, including six convolutional layer encoders and six symmetry. Each layer has two lstm layers. In addition to the gate convolution layer, the causal composite spectral mapping gate convolution recurrent network (GCRN) [23] proposed in 2019 has the same characteristics and the same structure as crn. Deep complex convolution recurrent network for phase-aware (DCCRN) [9] proposed in 2020 is based on the CRN model, but the convolutional layer is more complicated.

## 4 Results

This article uses the following objective evaluation indicators to evaluate the network training results:

1. PESQ: objective speech quality evaluation, objective MOS value evaluation provided by Recommendation ITU-TP.862 (0.5-4.5) [11];
2. STOI: Short-term objective intelligibility (0-1) [32];
3. CSIG: Focuses only on the average opinion score (MOS) prediction of the signal distortion of the speech signal (1-5) [10];
4. CBAK: Intrusive MOS prediction of background noise (1-5) [10];
5. COVL: MOS prediction of the overall effect (1-5) [10]. The higher score of the five indicators results in improved speech enhancement effect.

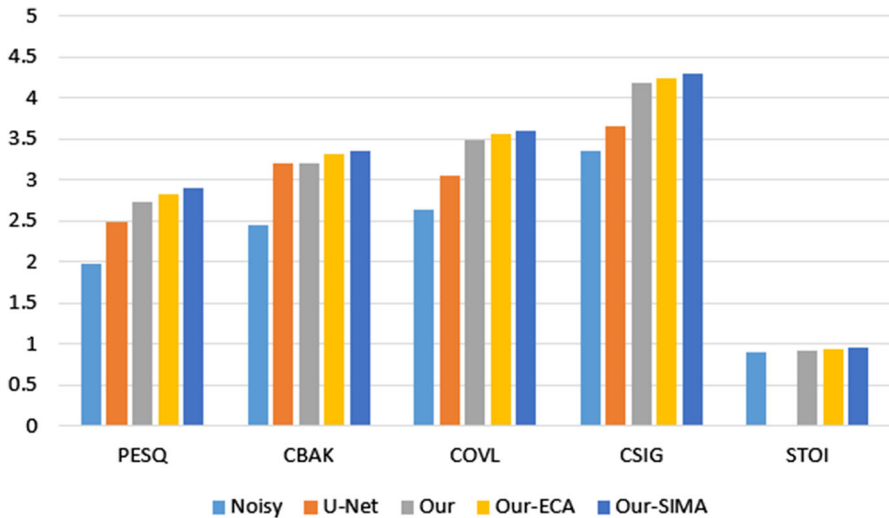


Fig. 4 Evaluation results of the proposed model

Our SIMA model in Table 1 uses a nonparticipant attention mechanism. Evidently, compared with the baseline model, the proposed model greatly improved the evaluation indicators of PESQ, CBAK, COVL, and CSIG. Compared with the DCCRN-E model in the literature [22], the PESQ index increased by 0.17 (6.23%), the CBAK index increased by 0.14 (4.34%), and the COVL index increased by 0.40 (12.42%). The CSIG indicator increased by 0.57 (15.28%). However, the STOI indicator hardly exhibits any growth. The results show that the proposed model improves the coding layer and the decoding layer in the speech enhancement task to achieve feature information enhancement and reduce information loss. A nonparameter attention mechanism is added to suppress redundant information, thereby extracting features that are more effective for speech enhancement tasks. In addition, the middle layer selects Bi-GRU to process the timing information of the voice signal and has achieved good results.

Figure 4 shows evaluation results of the proposed model. The difference is that the attention mechanism is not added. Our ECA is different from the proposed model in that the attention mechanism uses an efficient channel attention mechanism. Our SIMA represents the proposed model of the attention mechanism. The experimental results prove that adding the attention mechanism improves speech enhancement. The previous work indicated that the high-efficiency channel attention mechanism has a good improvement for speech enhancement tasks. For the proposed model, the efficient channel attention mechanism also improves the speech enhancement task to a certain extent. However, the enhancement effect brought by the parameter-free attention module is more evident. At the same time, all calculations in the parameter-free attention module are element-based operations, which do not involve structural adjustments and do not increase the number of parameters. Compared with the model before improvement, the model in this paper improves the enhancement effect and brings a certain increase in computational complexity, but in the experiment, the size

of the convolution kernel is reduced, and the number of convolution kernels is reduced as much as possible. Therefore, the computational complexity and the increase in the training and testing time of the model are also acceptable.

## 5 Conclusion

A complex convolutional recurrent network model with an attention mechanism is proposed. The model is based on a convolutional recurrent network. After fine-tuning the structure, a residual learning module with a parameter-free attention module is added to suppress redundant information, and the downsampling work is completed through the soft pool module. The horizontal and vertical features are enhanced through asymmetric convolution blocks, and the traditional convolution layer is replaced by DO-Cnov to improve the model training speed. Finally, feature fusion is added to the skip connection module; the accuracy of the model is improved without increasing the calculation amount. The experimental results show that the proposed method has strong competitiveness.

**Funding** The research was supported by the National Natural Science Foundation of China (62161040), Natural Science Foundation of Inner Mongolia (2021MS06030) and Inner Mongolia Science and Technology Project (2021GG0023), Supported By Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (NJYT22056)

**Data Availability** The datasets generated during and/or analyzed during the current study are available from the first author on reasonable request.

**Code Availability** The code will be made available on reasonable demand.

## Declarations

**Conflict of interest** None

## References

1. Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **5**, 157–166 (1994)
2. S. Boll, Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust. Speech Signal Process.* **27**, 113–120 (1979)
3. J. Cao, et al. Do-conv: depthwise over-parameterized convolutional layer. *arXiv preprint arXiv:2006.12030* (2020)
4. K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014)
5. I. Cohen, B. Berdugo, Speech enhancement for non-stationary noise environments. *Signal Process.* **81**, 2403–2418 (2001)
6. Y. Ephraim, D. Malah, Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Trans. Acoust. Speech Signal Process.* **32**, 1109–1121 (1984)
7. H. Gustafsson, S.E. Nordholm, I. Claesson, Spectral subtraction using reduced delay convolution and adaptive averaging. *IEEE Trans. Speech Audio Process.* **9**(8), 799–807 (2001)
8. S. Hochreiter, S. Schmidhuber, Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)

9. Y. Hu, et al. DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement. arXiv preprint [arXiv:2008.00264](https://arxiv.org/abs/2008.00264) (2020)
10. Y. Hu, P.C. Loizou, Evaluation of objective quality measures for speech enhancement. *IEEE Trans. Audio Speech Lang. Process.* **16**, 229–238 (2007)
11. ITU, R. I-T. P. 862.2: wideband extension to recommendation P. 862 for the assessment of wideband telephone networks and speech codecs. ITU-Telecommunication Standardization Sector (2007)
12. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
13. B. Kumar, Comparative performance evaluation of greedy algorithms for speech enhancement system. *Fluct. Noise Lett.* **20**(02), 2150017 (2021)
14. J.S. Lim, A.V. Oppenheim, Enhancement and bandwidth compression of noisy speech. *Proc. IEEE* **67**, 1586–1604 (1979)
15. S.R. Park, J. Lee, A fully convolutional neural network for speech enhancement. arXiv preprint [arXiv:1609.07132](https://arxiv.org/abs/1609.07132) (2016)
16. O. Ronneberger, P. Fischer, T. Brox, in U-net: convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241 (2015)
17. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986)
18. A. Stergiou, R. Poppe, G. Kalliatakis, Refining activation downsampling with Softpool. arXiv preprint [arXiv:2101.00440](https://arxiv.org/abs/2101.00440) (2021)
19. D. Stoller, S. Ewert, S. Dixon, Wave-u-net: A multi-scale neural network for end-to-end audio source separation. arXiv preprint [arXiv:1806.03185](https://arxiv.org/abs/1806.03185) (2018)
20. L. Sun, J. Du, L.-R. Dai, C.-H. Lee, in Multiple-target deep learning for LSTM-RNN based speech enhancement. *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, pp. 136–140 (IEEE, 2017)
21. K. Tan, D. Wang, in A convolutional recurrent neural network for real-time speech enhancement. *Interspeech*, pp. 3229–3233 (2018)
22. K. Tan, D. Wang, in Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6865–6869 (IEEE, 2019)
23. K. Tan, D. Wang, Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement. *IEEE/ACM Trans. Audio, Speech, Language Process.* **28**, 380–390 (2019)
24. J. Thiemann, N. Ito, E. Vincent, in The diverse environments multi-channel acoustic noise database (DEMAND): a database of multichannel environmental noise recordings. *Proceedings of Meetings on Acoustics ICA2013*, vol. 19 035081 (Acoustical Society of America, 2013)
25. C. Tian, Y. Xu, W. Zuo, C.-W. Lin, D. Zhang, Asymmetric CNN for image superresolution. *IEEE Trans. Syst. Man Cybernet. Syst.* (2021)
26. C. Valentini-Botinhao, others. Noisy speech database for training speech enhancement algorithms and tts models. (2017)
27. C. Veaux, J. Yamagishi, S. King, in The voice bank corpus: design, collection and data analysis of a large regional accent speech database. *2013 International Conference Oriental COCODSA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCODSA/CASLRE)*, pp. 1–4 (IEEE, 2013)
28. T.H. Vu, J.-C. Wang, Acoustic scene and event recognition using recurrent neural networks. *Detect. Classif. Acoust. Scenes Events* (2016)
29. Y. Wang, D. Wang, in Boosting classification based speech separation using temporal dynamics. *Thirteenth Annual Conference of the International Speech Communication Association* (2012)
30. B.S. Webb, N.T. Dhruv, S.G. Solomon, C. Tailby, P. Lennie, Early and late mechanisms of surround suppression in striate cortex of macaque. *J. Neurosci.* **25**, 11666–11675 (2005)
31. L. Yang, R.-Y. Zhang, L. Li, X. Xie, X. Simam, in A simple, parameter-free attention module for convolutional neural networks. *International Conference on Machine Learning*, pp. 11863–11874 (PMLR, 2021)
32. H. Zhang, X. Zhang, G. Gao, in Training supervised speech separation system to improve STOI and PESQ directly. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5374–5378 (IEEE, 2018)

33. M. Zhao, S. Zhong, X. Fu, B. Tang, M. Pecht, Deep residual shrinkage networks for fault diagnosis. *IEEE Trans. Industr. Inf.* **16**, 4681–4690 (2019)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.