A Time-domain Monaural Speech Enhancement with Feedback Learning

Andong Li*† Chengshi Zheng*† Linjuan Cheng*† Renhua Peng*† and Xiaodong Li*†
* Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
† University of Chinese Academy of Sciences, Beijing, China
E-mail: {liandong, cszheng, chenglinjuan, pengrenhua, lxd} @mail.ioa.ac.cn

***Abstract*—In this paper, we propose a type of neural network with feedback learning in the time domain called FTNet for monaural speech enhancement, where the proposed network consists of three principal components. The first part is called stage recurrent neural network, which is introduced to effectively aggregate the deep feature dependencies across different stages with a memory mechanism and also remove the interference stage by stage. The second part is the convolutional auto-encoder. The third part consists of a series of concatenated gated linear units, which are capable of facilitating the information flow and gradually increasing the receptive fields. Feedback learning is adopted to improve the parameter efficiency and therefore, the number of trainable parameters is effectively reduced without sacrificing its performance. Numerous experiments are conducted on TIMIT corpus and experimental results demonstrate that the proposed network can achieve consistently better performance in terms of both PESQ and STOI scores than two state-of-the-art time domain-based baselines in different conditions.**

I. Introduction

Speech is often inevitably degraded by background interference in real environments, which may significantly reduce the performance of automatic speech recognition (ASR), speech communication system and hearing aids. Monaural speech enhancement is dedicated to effectively extracting underlying target speech from its degraded version when only one measurement is available [1]. There are many well-known unsupervised signal-processing-based approaches, such as spectral subtraction [2], Wiener filtering [3] and statistical-based methods [4].

Recent advances in deep neural networks (DNNs) have facilitated the rapid development of speech enhancement research, and a great number of DNN models have been proposed to tackle the nonlinear mapping problem from the noisy speech to the clean speech (see [5], [6] and references therein). A typical DNN-based speech enhancement framework extracts time-frequency (T-F) features of the noisy speech and calculates some T-F representation targets of the clean speech. A model is then trained to establish the complicated mapping from the input features to the output targets with some supervised methods. Training targets can be categorized into two types, where one is the masking-based [7] and the other is the spectral mapping-based [6], [8].

Different from the research line in the T-F domain [6], [7], [8], a multitude of approaches based on time domain has emerged more recently [9], [11], [12], [13], [14]. Compared with T-F domain based methods, the major advantage of time domain approaches is that the phase estimation problem can be mitigated, which is helpful for speech quality [10]. Pandey et al. [13] took the U-Net with fully convolutional networks (FCNs) to directly model the waveform and utilized the domain knowledge from the time domain to the frequency domain to optimize the loss, which was significant for spectral detail recovery. Pascual et al. first applied the generative adversarial network (GAN) into the speech enhancement task in the time domain, where the generator was trained to produce a cleaner waveform whilst the discriminator was enforced to distinguish between the fake and clean versions. Luo et al. [9] utilized a learned encoder and decoder to project the speech waveform into a latent space, and superior performance was observed than short-time Fourier transform (STFT) based approaches in the speech separation task.

Despite the success of time-domain based approaches in the speech enhancement task [11], [12], [13], [14], these processing systems require a large number of trainable parameters, which may increase the computational complexity for practical applications. More recently, progressive learning (PL) has been applied in various tasks like single image deraining [15] and speech enhancement [16], where the whole mapping procedure is decomposed into multiple stages. In our preliminary work, we propose a PL-based convolutional recurrent network (PL-CRN) [17], where the noise components are gradually attenuated with a light-weight convolutional recurrent network (CRN) in each stage. We attribute the success of PL to the accumulation of prior information with the increase of the stages, i.e., all the outputs in the previous stages actually serve as the prior information to facilitate the execution of subsequent stages. Motivated by these studies, we propose a novel time-domain-based network with a feedback mechanism called FTNet, which needs much fewer trainable parameters. It works by recursively incorporating the estimated output from the last stage along with the original noisy feature back to the network, where each temporary output can be regarded as a type of state among different stages and thus trained with a recurrent approach. By doing so, the feature dependencies across different stages can be fully exploited and the output estimation can be refined stage by stage.

The remainder of this paper is structured as follows. Section II formulates the problem and briefly introduces the principal modules of the network. The proposed architecture is described in Section III. Section IV presents the experi-

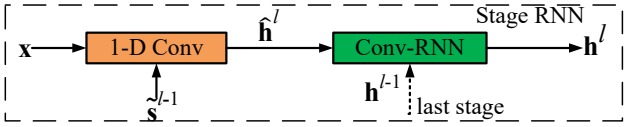


Fig. 1. The internal detail of SRNN module. It includes a 1-D Conv block and a Conv-RNN block. The module is operated with double input and single output (DISO).

mental settings. Experimental results and analysis are given in Section V. Some conclusions are drawn in Section VI.

II. Network Module

In the time domain, a mixture signal is usually formulated as $x(k) = s(k) + d(k)$, where k denotes the time index, $s(k)$, $d(k)$, and $x(k)$ are the clean speech, the noise, and the noisy speech, respectively. The network aims to estimate the time-domain clean speech. For notation convenience, we denote the frame vector of the noisy signal, estimation in lth stage, and the final output in the time domain as $\mathbf{x} \in \mathbb{R}^K$, $\tilde{\mathbf{s}}^l \in \mathbb{R}^K$, $\tilde{\mathbf{s}} \in \mathbb{R}^K$, respectively, where K is the frame length and l is the stage index. The proposed architecture is in essence a type of multi-stage network, where the output speech is estimated and refined stage by stage. Assuming the number of training stages is denoted as Q, in each stage, the estimated output from the last stage and the original noisy input are combined and sent back to the network. For the lth stage, the mapping process can be formulated as:

$$\tilde{\mathbf{s}}^l = g_\theta(\mathbf{x}, \tilde{\mathbf{s}}^{l-1}), \tag{1}$$

where $g_\theta(.)$ represents the network function. As seen from Eq. 1, both the estimation from the last stage and original noisy input are connected to update the estimation in the current stage.

A. Stage recurrent neural network

Theoretically, the learning process from the noisy feature to the clean target can be viewed as a type of sequence learning, where each state represents the intermediate output in one stage. To this end, we propose a type of recurrent convolutional structure named stage recurrent neural network (SRNN) to explore the time dependencies of different stages in this study. As a result, the network can be trained following a recurrent learning paradigm. As shown in Fig. 1, SRNN contains two parts, namely 1-D Conv block and convolutional-RNN (Conv-RNN). Assuming the inputs are $\mathbf{x}$ and $\tilde{\mathbf{s}}^{l-1}$, and the output of the 1-D Conv block is denoted as $\hat{\mathbf{h}}^l$. Then $\hat{\mathbf{h}}^l$ along with the hidden state vector from the last stage $\mathbf{h}^{l-1}$ is sent to Conv-RNN to obtain a updated hidden state, i.e., $\mathbf{h}^l$. As a result, the inference of $\mathbf{h}^l$ can be formulated as

$$\hat{\mathbf{h}}^l = f_{conv}(\mathbf{x}, \tilde{\mathbf{s}}^{l-1}), \tag{2}$$

$$\mathbf{h}^l = f_{conv\_rnn}(\hat{\mathbf{h}}^l, \mathbf{h}^{l-1}), \tag{3}$$

where $f_{conv}(\cdot)$ and $f_{conv\_rnn}(\cdot)$ represent the functions of 1-D Conv block and Conv-RNN block, respectively.

In this study, ConvGRU [18] is adopted as the unit for Conv-RNN, given as follows:

$$\mathbf{z}^l = \sigma\left(\mathbf{W}_z^l \circledast \hat{\mathbf{h}}^l + \mathbf{U}_z^l \circledast \mathbf{h}^{l-1}\right), \tag{4}$$

$$\mathbf{r}^l = \sigma\left(\mathbf{W}_r^l \circledast \hat{\mathbf{h}}^l + \mathbf{U}_r^l \circledast \mathbf{h}^{l-1}\right), \tag{5}$$

$$\mathbf{n}^l = \tanh\left(\mathbf{W}_n^l \circledast \hat{\mathbf{h}}^l + \mathbf{U}_n^l \circledast \left(\mathbf{r}^l \odot \mathbf{h}^{l-1}\right)\right), \tag{6}$$

$$\mathbf{h}^l = \left(\mathbf{1} - \mathbf{z}^l\right) \odot \hat{\mathbf{h}}^l + \mathbf{z}^l \odot \mathbf{n}^l, \tag{7}$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$, respectively, denote the sigmoid and the tanh activation functions. $\mathbf{W}$ and $\mathbf{U}$ refer to the weight matrices of the cell. $\circledast$ represents the convolutional operator and $\odot$ is the element-wise multiplication. Note that all the biases are neglected for notation simplicity.

B. Gated linear unit

Gated convolutional layer is first introduced in [19] to model complicated interactions in the form of a gating mechanism which is beneficial to performance and its modified version named GLU is utilized in [20] by replacing the $\tanh$ nonlinearity with a linear unit and residual learning is also incorporated to mitigate gradient vanishing problem when learning deep features [21]. In this study, we stack multiple GLU modules to explore the sequence correlations among neighboring points. As shown in Fig. 2-(b), two additional branches are introduced compared with the conventional CNN block, where one is the gated operation that is controlled with the sigmoid function to adjust the information flow percentage and the other is residual connection. Dilated convolution is applied to increase the receptive field, which is beneficial to capture more sequence correlations. We use parametric ReLU (PReLU) [22] as the activation function and the kernel size is set to 11 herein.

III. Proposed Architecture

The architecture of FTNet is illustrated in Fig. 2-(a), which includes three parts, namely SRNN, convolutional auto-encoder (CAE) [23] and a series of GLUs. SRNN consists of a 1-D Conv block and a ConvRNN block. 1-D Conv takes the concatenation of both noisy speech vector and the output estimation vector from the last stage along the channel axis. Therefore, the size of network input is $(2, K)$, where 2 refers to channels. After SRNN, the output is sent to the subsequent modules. CAE consists of the convolutional encoder and the decoder. The encoder consists of four 1-D Conv blocks, which compresses and establishes the deep representation of the features by halving the feature length with strided operation while consecutively doubling the channels. The decoder is the symmetric representation compared with the encoder, where the length of the feature is successively expanded through a number of deconvolutional layers [24]. Both encoder and decoder adopt PReLU as the activation nonlinearity except the output layer, where $\tanh$ is used to normalize the value range into $[-1, 1]$. Additionally, skip connections are adopted to connect each encoding layer to its homologous decoding layer, which compensates for the feature loss during the encoding process. To model the time correlations, six concatenated

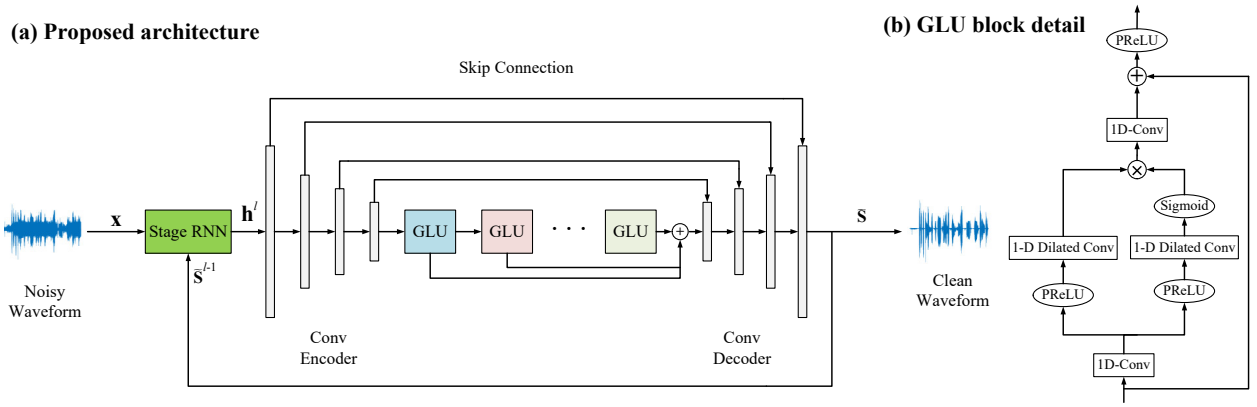


Fig. 2. The framework of proposed network FTNet with feedback learning. (a) The overview of FTNet. $\mathbf{x}$, $\tilde{\mathbf{s}}^{l-1}$, $\tilde{\mathbf{h}}^{l}$ and $\tilde{\mathbf{s}}$ denote the input feature, the estimation output in stage $l-1$, the state in stage l and the final estimation output, respectively. (b) The detail of GLU adopted in this study, where PReLU is adopted and the kernel size is set to 11.

TABLE I
DETAILED PARAMETER SETUP OF THE PROPOSED ARCHITECTURE.

layer name	input size	hyperparameters	output size
conv1d_1	2 × 2048	(11, 2, 16)	16 × 1024
conv_rnn	16 × 1024	(11, 1, 16)	16 × 1024
conv1d_2	16 × 1024	(11, 1, 16)	16 × 1024
conv1d_3	16 × 1024	(11, 2, 32)	32 × 512
conv1d_4	32 × 512	(11, 2, 64)	64 × 256
conv1d_5	64 × 256	(11, 2, 128)	128 × 128
GLUs	128 × 128	$\begin{pmatrix}1,1,64\\11,\mathbf{1},64\\1,1,128\end{pmatrix}\begin{pmatrix}1,1,64\\11,\mathbf{2},64\\1,1,128\end{pmatrix}\begin{pmatrix}1,1,64\\11,\mathbf{4},64\\1,1,128\end{pmatrix}\begin{pmatrix}1,1,64\\11,\mathbf{8},64\\1,1,128\end{pmatrix}\begin{pmatrix}1,1,64\\11,\mathbf{16},64\\1,1,128\end{pmatrix}\begin{pmatrix}1,1,64\\11,\mathbf{32},64\\1,1,128\end{pmatrix}$	128 × 128
skip_1	128 × 128	-	256 × 128
deconv1d_1	256 × 128	(11, 2, 64)	64 × 256
skip_2	64 × 256	-	128 × 256
deconv1d_2	128 × 256	(11, 2, 32)	32 × 512
skip_3	32 × 512	-	64 × 512
deconv1d_3	64 × 512	(11, 2, 16)	16 × 1024
skip_4	16 × 1024	-	32 × 1024
deconv1d_4	32 × 1024	(11, 2, 1)	1 × 2048

GLUs are inserted between the encoder and decoder, where the dilated rates are $(1, 2, 4, 8, 16, 32)$.

When the estimation output of the lth stage is obtained, i.e., $\tilde{\mathbf{s}}^{l}$, it is fed back and concatenated with the noisy input $\mathbf{x}$ along channel axis to execute the next stage. Here we only impose supervision on the final output $\tilde{\mathbf{s}}$, which is consistent with the setting in [15].

A more detailed parameter configuration of the proposed network is summarized in Table I, where the input and output sizes of 2-D tensor representation are specified with $(Channels \times Framesize)$ format. The hyperparameters of the layers except GLUs are specified with $(KernelSize, Strided, Channels)$ format. The hyperparameters of GLUs are specified with $(KernelSize, DilatedRate, Channels)$ format. Bold numbers refer to the dilated rate.

IV. EXPERIMENTS

A. Datasets

Experiments are conducted on TIMIT corpus [25], which includes 630 speakers of eight major dialects of American English with each reading ten utterances. 1000, 200 and 100 clean utterances are randomly selected for training, validation and testing, respectively. Training and validation dataset are mixed under different SNR levels ranging from -5dB to 10dB with the interval 1dB while the testing datasets are mixed under -5dB and -2dB conditions. For training and validation, we use 130 types of noises, including 115 types used in [17], 9 types from [26], 3 types from NOISEX92 [27] and 3 common environmental noise, i.e. aircraft, bus and cafeteria. Another 5 types of noises from NOISEX92, including babble, f16, factory2, m109 and white, are chosen to test the network generalization capacity.

Various noises are first concatenated into a long vector. During each mixed process, the cutting point is randomly generated, which is subsequently mixed with a clean utterance under one SNR condition. As a result, totally 10,000, 2000 and 400 noisy-clean utterance pairs are created for training, validation, and testing, respectively.

B. Baselines

In this study, two advanced time-domain-based networks are selected as the baselines, namely AECNN [13] and RHR-Net [14]. AECNN is a typical 1-D Conv based auto-encoder architecture with a large number of trainable parameters. The number of channels in consecutive layers are {64, 64, 64, 128, 128, 128, 256, 256, 256, 512, 512, 256, 256, 256, 128, 128, 128, 1}, with 11 and PReLU being the filter size and activation

TABLE II
EXPERIMENTAL RESULTS UNDER SEEN NOISE CONDITIONS FOR PESQ AND STOI. **BOLD** INDICATES THE BEST RESULT FOR EACH CASE. THE NUMBER OF STAGES Q ARE SET TO 3, 4 AND 5 FOR MODEL COMPARISONS.

Metrics	PESQ			STOI (in %)		
SNR	-5dB	-2dB	Avg.	-5dB	-2dB	Avg.
Noisy	1.47	1.66	1.57	63.03	68.20	65.62
AECNN	2.25	2.49	2.37	82.70	87.51	85.11
RHR-Net	2.32	2.55	2.44	83.13	87.90	85.51
FTNet (Q = 3)	2.36	2.59	**2.48**	83.18	87.92	85.55
FTNet (Q = 4)	2.35	2.59	2.47	83.75	88.39	86.07
FTNet (Q = 5)	**2.37**	**2.60**	**2.48**	**84.03**	**88.54**	**86.28**

TABLE III
EXPERIMENTAL RESULTS UNDER UNSEEN NOISE CONDITIONS FOR PESQ AND STOI. **BOLD** INDICATES THE BEST RESULT FOR EACH CASE. THE NUMBER OF STAGES Q ARE SET TO 3, 4 AND 5 FOR MODEL COMPARISONS.

Metrics	PESQ			STOI (in %)		
SNR	-5dB	-2dB	Avg.	-5dB	-2dB	Avg.
Noisy	1.44	1.67	1.56	59.64	67.45	63.55
AECNN	1.88	2.20	2.04	77.37	85.10	81.24
RHR-Net	2.06	2.35	2.21	78.13	85.82	81.98
FTNet (Q = 3)	**2.10**	**2.37**	**2.23**	78.59	85.68	82.13
FTNet (Q = 4)	2.06	2.35	2.21	79.31	86.20	82.76
FTNet (Q = 5)	2.09	2.35	2.22	**79.48**	**86.54**	**83.01**

nonlinearity, respectively. RHR-Net has also the form of auto-encoder framework except all the convolutional layers are replaced by bidirectional GRU (BiGRU). In addition, direct skip connections are replaced by PReLU based residual connections. It achieves state-of-the-art metric performance among several advanced speech enhancement models with limited trainable parameters (see [14]). The number of units per layer are $\{1,\ 32,\ 64,\ 128,\ 256,\ 128,\ 64,\ 32,\ 1\}$ and three residual skip connections are introduced. Note that the last layer is a single-directional GRU to output the enhanced signal.

C. Experimental settings

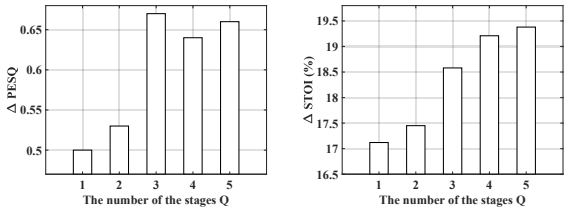
We sample all the utterances at 16kHz. Each frame has a size of 2048 samples (128 ms) with 256 samples (16 ms) offset between adjacent frames. All the models are trained with mean absolute error (MAE) criterion, optimized by Adam algorithm [28]. The learning rate is initialized at 0.0002. We halve the learning rate only if consecutive three validation loss increment arises and the training process is early-stopped only if ten validation loss increment happens. We train all the models for 50 epochs. Within each epoch, the minibatch is set to 2 at the utterance level, where all the utterances are randomly chunked to 4 seconds if they exceed 4 seconds and zero-padded on the contrary.

V. RESULTS AND ANALYSIS

We evaluate the performance of different models in terms of perceptual evaluation of speech quality (PESQ) [29] and short-time objective intelligibility (STOI) [30].

A. Objective results comparison

The objective results are presented in Tables II and III. One can observe the following phenomena. Firstly, all the models



(a) (b)

Fig. 3. PESQ and STOI improvements with the increase of the number of the stages Q. The values are averaged over unseen dataset. Here five values are explored, i.e., $Q = 1, 2, 3, 4, 5$.

significantly improve the scores in terms of PESQ and STOI for both seen and unseen cases, whilst the proposed FTNet achieves the best performance among the three models. For example, for seen cases, when $Q = 3$, FTNet improves PESQ by 0.11 and 0.04, and improves STOI by 0.44% and 0.04% over AECNN and RHR-Net, respectively. This is because the memory mechanism is utilized to refine the network with a stage-wise manner and improve the parameter efficiency. A similar tendency is also observed for unseen cases. Secondly, when comparing between two baselines, RHR-Net obtains consistently better performance than AECNN. This is because BiGRU is adopted as the basic component for both encoding and decoding process, which facilitates better temporal capture capability for long sequences than 1-D Conv, whose performance is limited by kernel size and dilation rate. This can also partly explain the limited advantages of FTNet over RHR-Net.

B. The influence of stage number Q

In this study, we explore the influence of the number of the stages Q, and it takes the values from 1 to 5. Note that $Q = 1$ means that only one stage is applied and no memory mechanism is adopted to bridge the relationship between neighboring stages. The metric improvements are given in Fig. 3. One can observe the following phenomena. Firstly, when $Q \leq 3$, both PESQ and STOI scores are consistently improved with the increase of Q, indicating that both metrics can be effectively refined with feedback learning. Nonetheless, when Q takes from 3 to 5, PESQ falls into saturation even slightly attenuation while STOI is further improved. This is because MAE is adopted as the loss criterion, whose optimization target is inconsistent with the objective evaluation criterion and can not further refine both metrics simultaneously [31]. This phenomenon reveals that further optimization of MAE can improve STOI but may slightly reduce PESQ.

C. Insights into feedback learning

In this subsection, we attempt to analyze the effect of feedback learning. To avoid illustration confusion, we fix the number of stages as 5 herein, i.e., $Q = 5$. First, we give the metric scores in the intermediate stages, and the results are shown in Fig. 4. One can see that when the first stage is finished, the estimation has similar metric scores over the noisy input in both PESQ and STOI. However, when the network is

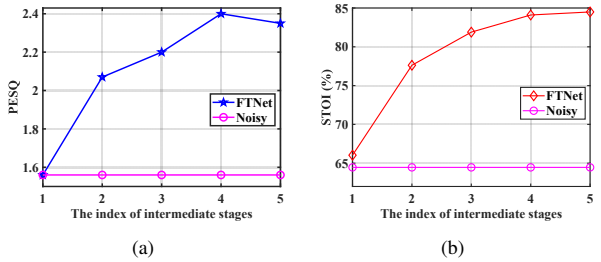


Fig. 4. The metric scores in terms of PESQ and STOI for different intermediate stages given Q = 5. The results are averaged over both seen and unseen conditions. Noisy scores are also presented for comparison.

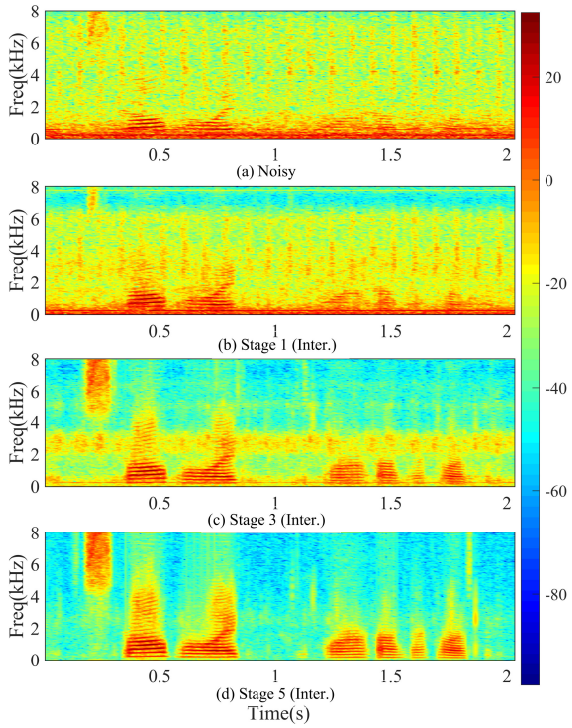


Fig. 5. Spectral visualization for different intermediate stages given Q = 5. (a) Noisy spectrogram under -5dB, PESQ=0.98. (b) Enhanced spectrogram in the first stage, PESQ=1.06. (c) Enhanced spectrogram in the third stage, PESQ=1.61. (d) Enhanced spectrogram in the fifth stage, PESQ=1.83.

recursed for more stages, a notable improvement is observed. This indicates that when the estimation from the previous stage is sent back to the network as the feedback component, more prior information can be accumulated and the network is guided to generate cleaner speech estimation. The spectral visualization of the intermediate stages is also presented in Fig. 5. We only give the first, third, and fifth stage herein for convenience. One can see that compared with the input spectrogram, the estimation in the first stage is also relatively noisy. Nevertheless, when more feedback is applied, the noise components are gradually suppressed, which emphasizes the effectiveness of feedback learning.

As Section II-A states, SRNN is utilized to aggregate the

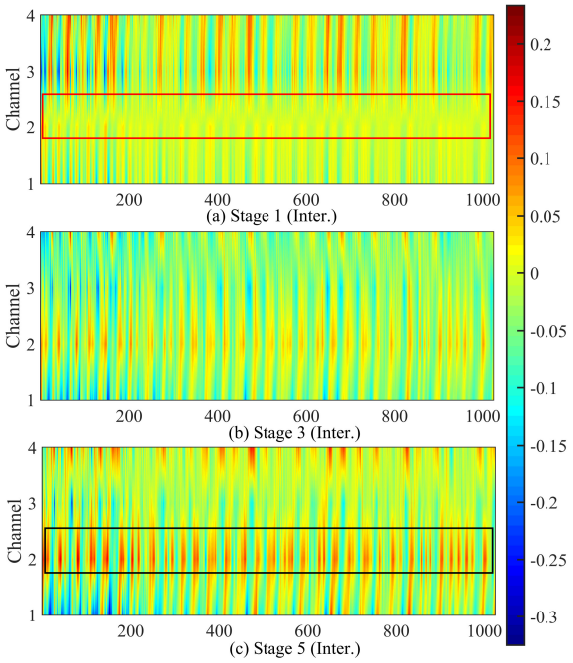


Fig. 6. Visualization of hidden state $\mathbf{h}$ within SRNN. The size of $\mathbf{h}$ is $(16, 1024)$, where 16 and 1024 refer to the channel and feature axis, respectively. We only plot the first 4 channels for convenience. (a) state visualization in the first stage. (b) state visualization in the third stage. (c) visualization in the fifth stage.

TABLE IV
THE NUMBER OF TRAINABLE PARAMETERS AMONG DIFFERENT MODELS. THE UNIT IS MILLION. **BOLD** INDICATES THE LOWEST TRAINABLE PARAMETERS.

Model	AECNN	RHR-Net	FTNet
Para. (million)	6.31	1.95	1.02

feature information across different stages with a memory mechanism. As such, the hidden state $\mathbf{h}^l$ (we omit superscript for simplicity hereafter) is updated in each feedback stage. To emphasize that, we visualize $\mathbf{h}$ in three intermediate stages given Q = 5, which is presented in Fig. 6. As the size of $\mathbf{h}$ is $(16, 1024)$ (see Table I), we only extract the first four channels for convenience. One can observe that, for the first stage, SRNN has yet learned clear prior information, leading to blurring feature representation in the hidden state, as shown in Fig. 6 (a), the red box area. When more stages are applied, the SRNN begins to accumulate more prior information about clean speech. As a result, the representation of $\mathbf{h}$ becomes clearer stage by stage, as shown in Fig. 6 (c), the black box area.

D. Trainable parameters and ideal network depth

The number of trainable parameters for the baselines and proposed FTNet is presented in Table IV. One can see that compared with AECNN and RHR-Net, FTNet further decreases the number of trainable parameters, which demonstrates the high parameter efficiency of feedback learning.

To improve network performance, a deeper network is

needed, which usually results in more trainable parameters. With feedback learning, the network is reused for multiple stages, and we can explore a deeper network without additional parameters. In this paper, considering the gradient flow, the number of the ideal layers for FTNet is $28\times$Q, where 28 represents the number of layers for the feedforward gradient flow. Therefore, a deeper network can be explored by recursing the network for more stages.

VI. Conclusions

In this study, we propose a type of feedback network in the time domain named FTNet for monaural speech enhancement. Stage RNN is proposed to effectively aggregate the deep features across different stages. In addition, concatenated GLUs are adopted to increase the receptive field while controlling the information flow. Experimental results demonstrate that FTNet achieves consistently better performance than the other two advanced time-domain baselines and effectively reduces the number of trainable parameters simultaneously.

References

[1] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.

[2] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing,* vol. 27, no. 2, pp. 113-120, 1979.

[3] X. Hu, S. Wang, C. Zheng, and X. Li, "A cepstrum-based preprocessing and postprocessing for speech enhancement in adverse environments, " *Applied Acoustics,* vol. 74, no. 12, pp. 1458-1462, 2013.

[4] S. Jensen, P. Hansen, S. Hansen, and J. Sorensen, "Reduction of broadband noise in speech by truncated QSVD," *IEEE Transactions on Speech and Audio Processing,* vol. 3, no. 6, pp. 439-448, 1995.

[5] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 26, no. 10, pp. 1702-1726, 2018.

[6] Y. Xu, J. Du, L. Dai, and C. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing,* vol. 23, no. 1, pp. 7-19, 2015.

[7] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM tranactions on audio, speech and language processing,* vol. 22, no. 12, pp. 1849-1858, 2014.

[8] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Proc. INTERSPEECH*, 2018, pp. 3229-3233.

[9] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. ICASSP*. IEEE, 2018, pp. 696–700.

[10] K. Paliwal, K. Wojcicki, and B. Shannon, "The importance of phase in speech enhancement," *speech communication*, vol. 53, no. 4, pp. 465-494, 2011.

[11] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[12] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *Proc. ICASSP*, IEEE, 2018, pp. 5069-5073.

[13] A. Pandey and D. Wang, "A new framework for cnn-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 27, no. 7, pp. 1179-1188, 2019.

[14] J. Abdulbaqi, Y. Gu, and I. Marsic, "RHR-Net: A residual hourglass recurrent neural network for speech enhancement," *arXiv preprint arXiv:1904.07294*, 2019.

[15] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: a better and simpler baseline," in *Proc. CVPR*, 2019, pp. 3937-3946.

[16] T. Gao, J. Du, L. Dai, and C. Lee, "Densely connected progressive learning for lstm-based speech enhancement," in *Proc. ICASSP*. IEEE, 2018, pp. 5054-5058.

[17] A. Li, M. Yuan, C. Zheng, and X. Li, "Speech enhancement using progressive learning-based convolutional recurrent neural network," *Applied Acoustics*, vol. 166, p. 107347, 2020.

[18] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[19] A. Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, "Conditional image generation with pixelcnn decoders," in *Advances in neural information processing systems*, 2016, pp. 4790-4798.

[20] K. Tan, J. Chen, and D. Wang, "Gated residual networks with dilated convolutions for monaural speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 189-198, 2018.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770-778, 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on Imagenet classification," in *Proc. ICCV*, 2015, pp. 1026-1034.

[23] V. Badrinarayanan, A. Handa, and R. Cipolla, "Sgenet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[24] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. ICCV*, 2015, pp. 1520-1528.

[25] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, and D. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[26] Z. Duan, G. Mysore, and P. Smaragdis, "Speech enhancement by online non-negative spectrogram decomposition in nonstationary noise environments," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[27] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247-251, 1993.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749-752.

[30] C. Taal, R. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125-2136, 2011.

[31] S. Fu, T. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1570-1584, 2018.