Combining Multi-Perspective Attention Mechanism With Convolutional Networks for Monaural Speech Enhancement

TIAN LAN¹,², (Member, IEEE), YILAN LYU¹, (Graduate Student Member, IEEE), WENZHENG YE², GUOQIANG HUI¹, ZENGLIN XU³, (Member, IEEE), AND QIAO LIU¹, (Member, IEEE)
¹School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
²CETC Big Data Research Institute Company, Ltd., Guiyang 550008, China
³School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
Corresponding author: Qiao Liu (qliu@uestc.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant U19B2028 and Grant 61772117, in part by the National Science and Technology Commission Innovation Project under Grant 19-H863-01-ZT-003, in part by the CETC Big Data Research Institute under Grant 10-2018039, in part by the Sichuan Hi-Tech Industrialization Program under Grant 2018GFW0150, and in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2019J077.

ABSTRACT The redundant convolutional encoder-decoder network has been proven useful in speech enhancement tasks. This network can capture the localized time-frequency details of speech signals through the fully convolutional network structure and the feature selection capability that results from the encoder-decoder mechanism. However, extracting informative features, which we regard as important for the representational capability of speech enhancement models, is not considered explicitly. To solve this problem, we introduce the attention mechanism into the convolutional encoder-decoder model to explicitly emphasize useful information from three aspects, namely, channel, space, and concurrent space-and-channel. Furthermore, the attention operation is specifically achieved through the squeeze-and-excitation mechanism and its variants. The model can adaptively emphasize valuable information and suppress useless ones by assigning weights from different perspectives according to global information, thereby improving its representational capability. Experimental results show that the proposed attention mechanisms can employ a small fraction of parameters to effectively improve the performance of CNN-based models compared with their normal versions, and generalize well to unseen noises, signal-to-noise ratios (SNR) and speakers. Among these mechanisms, the concurrent space-channel-wise attention exhibits the most significant improvement. And when comparing with the state-of-the-art, they can produce comparable or better results. We also integrate the proposed attention mechanisms with other convolutional neural network (CNN)-based models and gain performance. Moreover, we visualize the enhancement results to show the effect of the attention mechanisms more clearly.

INDEX TERMS Speech enhancement, squeeze-and-excitation, convolutional neural network, attention.

I. INTRODUCTION

Speech enhancement aims to remove background noise from the degraded speech without distorting the clean speech, thereby improving the speech quality and intelligibility. This technique is widely used in many applications, such as speech recognition [1], hearing aids [2], and VoIP [3]. Common speech enhancement techniques fall under two major categories: traditional and machine-learning-based methods. Traditional methods mainly include spectral subtraction [4], Wiener filtering [5], statistical-model-based methods [6], and subspace-based methods [7]. These methods mainly use unsupervised digital signal analysis approaches and achieve separation by decomposing the speech signal to determine the characteristics of clean speech and noise. These methods can eliminate noise to some extent. However, the performance of these methods greatly degrades when dealing with non-stationary noises because they are based on the assumption...
of stable noises. To solve these limitations, supervised methods that can automatically discover the relationship between noisy and clean speech signals are constantly proposed. Among these methods, deep learning-based methods have dramatically boosted their denoising performance recently, thereby attracting numerous researchers and resulting in the proposal of many neural network-based models [8]–[10] like deep neural network (DNN) [11]–[13], recurrent neural networks (RNN) [14]–[16], convolutional neural network (CNN) [17]–[21] and some other variants.

Xu et al. [8] proposed a regression-based DNN for mapping the log-power spectra features of noisy speech to those of clean speech. This model achieves satisfactory results, thereby proving the effectiveness of deep-learning-based methods. However, DNN is composed of several fully connected layers that experience difficulty in modeling the temporal structure of a speech signal [22]. In addition, the number of parameters increases rapidly with the layers and nodes, thereby raising the computational burden.

CNN is introduced into speech processing to capture implicit information in the speech signal remarkably while reducing the number of parameters in recent years. CNN can maintain a small shift in the frequency domain of speech features within a certain range, thus coping with speaker and environmental changes [22]. Fu et al. [18] proposed a signal-to-noise ratio (SNR)-aware CNN to estimate the SNR of an utterance and then enhance adaptively, thus improving the generalization ability. Hou et al. [20] employed both audio and visual information for enhancement. Bhat et al. [21] proposed a multi-objective learning CNN and implemented it on a smartphone as an application. Recently, more encoder-decoder based CNN models are proposed. Park and Lee [23] removed the fully connected layers in CNN and introduced the fully convolutional network (FCN) into the field of speech enhancement considering the disadvantages mentioned previously. Recently, many works have been proposed based on the FCN. Tan and Wang [24] proposed the convolutional recurrent network (CRN), which inserted two long short-term memory (LSTM) layers between the encoder and the decoder of the FCN. Grzywalski and Drągas [25] added gated recurrent unit (GRU) layers into each building block of the FCN. These models improve the representational capability by exploiting the temporal modeling capability of RNN. The max-pooling layers in the FCN are used to extract the most active parts of different features. Considering that the information in temporal structure of a speech signal [22]. In addition, the number of parameters increases rapidly with the layers and nodes, thereby raising the computational burden.

CNN is introduced into speech processing to capture implicit information in the speech signal remarkably while reducing the number of parameters in recent years. CNN can maintain a small shift in the frequency domain of speech features within a certain range, thus coping with speaker and environmental changes [22]. Fu et al. [18] proposed a signal-to-noise ratio (SNR)-aware CNN to estimate the SNR of an utterance and then enhance adaptively, thus improving the generalization ability. Hou et al. [20] employed both audio and visual information for enhancement. Bhat et al. [21] proposed a multi-objective learning CNN and implemented it on a smartphone as an application. Recently, more encoder-decoder based CNN models are proposed. Park and Lee [23] removed the fully connected layers in CNN and introduced the fully convolutional network (FCN) into the field of speech enhancement considering the disadvantages mentioned previously. Recently, many works have been proposed based on the FCN. Tan and Wang [24] proposed the convolutional recurrent network (CRN), which inserted two long short-term memory (LSTM) layers between the encoder and the decoder of the FCN. Grzywalski and Drągas [25] added gated recurrent unit (GRU) layers into each building block of the FCN. These models improve the representational capability by exploiting the temporal modeling capability of RNN. The max-pooling layers in the FCN are used to extract the most active parts of different areas, while the detail information is lost. Therefore, FCN can achieve good results in some fields, such as speech recognition, where obtaining the overall characteristics is enough. However, in speech enhancement, the use of detail information is essential in restoring clean speech. To solve this problem, [23] also proposed the redundant convolutional encoder-decoder (RCED), which discarded the max-pooling layers and the corresponding upsampling layers in the FCN to maintain the feature map size, thereby retaining the details and achieving improved performance.

To further improve the performance of CNN-based models, many methods focus on the depth, width, and cardinality of networks [26]. Unlike previous works, we integrate the attention mechanism with the RCED in this work to improve representational capability. Attention [27] is a brain signal processing mechanism. It allows human brains to assign different attention to each part automatically, thus effectively capturing several informative features. The fusion of deep learning-based models and the attention mechanism can help models emphasize informative features and suppress useless ones. At present, attention has been widely applied in speech recognition [28], answer selection [29], and session prediction [30]. Although the use of attention mechanism is not uncommon in speech enhancement, there are three reasons why we think it can play a role: First, in a noisy environment, the human auditory system can selectively focus on speech while suppressing noise through the attention mechanism [31]. Therefore, the application of attention may help the model simulate the human auditory system and capture speech from noises, thus improving its expressive ability. Second, [32] introduced the attention mechanism into LSTM to assign a weight to past several frames, then calculated their weighted sum as the context frame for each timestep. This model achieved satisfactory results, thereby demonstrating the efficiency of attention mechanism in monaural speech enhancement. Finally, given that spectrograms have a specific pattern, they can be treated as images and processed using an image processing methodology [33].

At present, most attention-based methods multiply two vectors point-to-point to calculate their similarity. In the field of image processing, Hu et al. [34] proposed a new type of attention mechanism for CNNs called squeeze-and-excitation (SE), which can summarize the information of all output channels with a small number of parameters, and learn to give a weight to each channel according to the global information. The SE mechanism consists of two steps and is leveraged to assign a weight to each channel according to all feature maps. The squeeze step integrates the global spatial information and generates a channel descriptor, in which each element corresponds to a feature map information. In the excitation step, the descriptor is adjusted, and the attention weight of each channel is determined. Finally, the weights are used to recalibrate the feature maps through which the model can emphasize useful information. The weight recalibration benefits from the SE layers are accumulated throughout the entire network. Recently, SE has been widely used in image processing and has obtained satisfactory results [33], [35], [36]. Later, Roy et al. [37] and Woo et al. [26] extended SE to space and concurrent space-and-channel domains and achieved satisfactory results.

Motivated by these works, we introduce SE as the attention mechanism, and combine it with RCED, thus solving the problem that RCED has difficulty in effectively exploiting global information [25] or explicitly judging the importance of different features. Considering that the information in time-frequency points are also of great importance, we propose a spatial SE (SSE) mechanism that provides weights spatially. Moreover, we exploit channel-wise and spatial
The representational capability of the original RCED can be improved by explicitly emphasizing useful information through SE. Considering the accumulation benefits of SE, we add one SE layer at the end of each building block in the RCED. Experimental results show that the SE mechanism can improve the performance effectively and show a good generalization ability. We also integrate it with other CNN-based models and find that such an approach can improve performance.

The rest of this paper is organized as follows. The next section describes the general framework of the proposed model. Section III describes the proposed SE mechanisms. Section IV and V respectively provide the configurations and results of the conducted experiments. Section VI presents the concluding remarks of this study.

II. MODEL DESCRIPTION

The original RCED is an encoder-decoder-based structure. Except for the last one, nearly all building blocks contain a convolutional layer, followed by a batch normalization (BN) layer, and a rectified linear unit (ReLU) layer. The last compression block only contains one convolutional layer to summarize the information obtained in the previous operations. Unlike normal FCN, the RCED discards the max-pooling layers and the corresponding upsampling layers to alleviate detail loss. Therefore, the feature map size of each layer in these blocks remains consistent, while the feature map number changes. In this way, the encoder can be perceived to generate many redundant features for each time-frequency point with an increasing number of filters, and each channel corresponds to a feature type. Given that some features are important to the mapping accuracy while some are not, the decoder is used for the gradual removal of unwanted features.

However, the decoder in the RCED compresses features simply by directly changing the number of convolutional kernels. This phenomenon makes the precise identification of whether the feature map is important or not difficult for the model, thereby limiting its representational capability. Considering that the attention mechanism can help the model focus on important features, we introduce it to solve this problem. For the attention mechanism, we choose the SE mechanism in [34]. And we add SE to both of the encoder and the decoder.

The overall architecture of the proposed attention-based redundant convolutional network (ARCN) is illustrated in Fig. 1 (a). ARCN consists of nine attention-based convolutional blocks (ACB) and a final convolutional layer. Each cube corresponds to an ACB block, and the parameter above it (i.e., ACB#id_inNum_outNum) indicates the id and the numbers of input and output feature maps. ACB architecture is illustrated in Fig. 1 (b). Except for the convolutional layer, the batch normalization and activation layers are included in the building block of the RCED, and ACB also contains an attention layer in the end. In this manner, we can increase the capability of the model to determine the importance of each feature. Moreover, instead of ReLU, we use Leaky ReLU [38] as the activation function to avoid zero gradients. For each utterance input, an enhanced utterance is generated by the model after processing.

III. ATTENTION MECHANISM

SE is a self-gating mechanism that can recalibrate the output feature maps adaptively. In this way, the model can selectively emphasize the valuable features and suppress useless ones in consideration of the global information. However, the origi-
nal SE in [34] only focuses on the channel aspect. Meanwhile, in the field of speech enhancement, the information in time-frequency points also plays an important role. Moreover, in image segmentation, Roy et al. [37] successfully extended the SE mechanism to the space domain by point-wise convolution (i.e., squeezes along the channels and excites spatially). Inspired by this work, we considered improving the expressive ability of the model by assigning weights according to the global information from the channel aspect, the space aspect, and both. Our idea is similar to that in [37], but we propose different approaches to calculate the weights.

Let $U \in \mathbb{R}^{H \times W \times C}$ be the output feature maps of each building block $F_{tr}(\cdot)$ in RCED, where $H$, $W$, and $C$ represent the height, width, and the number of the feature maps, respectively. $F_{tr}(\cdot)$ consists of a convolutional layer, a BN layer and an activation layer. Then, we apply a SE layer to $U$ to select valuable information, thereby representing the subsequent blocks better.

A. CHANNEL-WISE SE
The structure of SE mechanism is shown in Fig. 2 (a). As it assigns a weight to each channel, we name it channel-wise SE (CSE). We divide $U$ into $C$ feature maps according to the channels, each of which is denoted as $u_i \in \mathbb{R}^{H \times W}$, so $U = [u_1, u_2, \ldots, u_C]$. First, we obtain channel descriptor $Z$, and each element $z_k$ aggregates the information of the corresponding channel using global average pooling as follows:

$$z_k = \frac{1}{H \times W} \sum_i \sum_j u_k(i,j) \quad k = 1, 2, \ldots, C. \quad (1)$$

Then, the excitation operation is exploited to calculate the weights of all channels as $S \in \mathbb{R}^C$, according to the global information obtained in the previous operation.

$$S = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \quad (2)$$

where $g(\cdot)$ and $\sigma$ represent a gating mechanism and a sigmoid function, respectively. $g(\cdot)$ is formed by two fully connected layers and the ReLU function ($\delta$). $W_1 \in \mathbb{R}^{F \times C}$ is used for dimension reduction, and $W_2 \in \mathbb{R}^{C \times F}$ is for dimension restoration, where $r$ refers to the reduction ratio that can be used to vary the capacity and the calculated amount of the SE layers; $r$ value is set as needed. After obtaining the collection of weights, we apply channel-wise multiplication between each feature map in $U$ and its corresponding weight in $S$ to recalibrate the features.

B. SPATIAL SE
Roy et al. [37] believed that pixel-wise information is important to images and then extended SE to spatial aspect in image segmentation. Unlike the original SE, which calculates the importance of channels, the SSE assigns a weight to each pixel. Given that spectrogram is similar to images, we hypothesize that the information contained in the time-frequency points in speech enhancement is also of great importance. Therefore, the use of SSE can be perceived as removing invalid information from every time-frequency embedding.

The SSE structure is illustrated in Fig. 2 (b). SSE squeezes along the channels and excites spatially. We first cut $U$ into a total of $H \times W$ tensors, with a $1 \times 1 \times C$ shape, and denote it as $U = [u_{1,1}, u_{1,2}, \ldots, u_{i,j}, \ldots, u_{H,W}]$. $u_{i,j}$ is the embedding.
representation of the time-frequency point at time \( i \) and frequency \( j \).

Second, we implement the squeeze operation and obtain the weight for \( u_{ij} \). The extraction of additional information can help predict the spatial weights remarkably. Given that the average pooling layer can reflect the overall information of the feature map, the max-pooling layer can detect certain features, and the dilated convolutional layer can effectively capture the contextual information with different scales, we consider using all of them to process \( U \) to obtain the spatial weights. The output of these operations are denoted as \( \text{AvgDesc} \in \mathbb{R}^{H \times W} \), \( \text{MaxDesc} \in \mathbb{R}^{H \times W} \), \( dlt_4 \in \mathbb{R}^{H \times W} \), \( dlt_8 \in \mathbb{R}^{H \times W} \), and \( dlt_{16} \in \mathbb{R}^{H \times W} \). \( dlt_x \) indicates that the dilation rate of the dilated convolutional layer is \( x \). Then, we concatenate them and integrate to obtain the spatial weight by a convolutional layer.

\[
A = \text{Conv}_1([\text{MaxDesc}, \text{AvgDesc}, dlt_4, dlt_8, dlt_{16}]),
\]

where \( \text{Conv}_1 \) is the convolution operation, and \( n \) is the output feature map number. The kernel size in time and the frequency axis are 11 and 9, respectively. As the attention weight should be in \([0, 1]\), a sigmoid function is then applied:

\[
weight_{ij} = \text{sigmoid}(A_{ij}).
\]

Then, we can obtain the tensor \( weight \in \mathbb{R}^{H \times W} \) and use it in the excitation operation. Each recalibrated time-frequency point is obtained as follows:

\[
u_{ij} = weight_{ij}^t u_{ij} \quad (i = 0, 1, \cdots, H; j = 0, 1, \cdots, W),
\]

where \( u_{ij} \) is the embedding representation that corresponds to the point at time \( i \) and frequency \( j \). \( weight_{ij} \) is the relative importance of \( u_{ij} \). In this way, the model can concentrate on other informative features from the time-frequency aspect.

**C. SPACE-CHANNEL-WISE SE**

In addition to assigning weights to the feature maps alone and to the time-frequency embeddings alone, we also explore the use of both to simultaneously recalibrate \( U \) spatially and channel-wise. We use four ways of combining them: parallel addition, parallel concatenation, sequential channel-space operation, and sequential space-channel operation. These ways can encourage the model to extract important features accurately by exploiting different aspects of information.

1) PARALLEL CONCATENATION

In the parallel space-channel-wise concatenation method (SCConcat), we first obtain the channel weighted output and the space weighted output concurrently according to CSE and SSE and denote them as \( U_{sp} \) and \( U_{ch} \), respectively. Then, we concatenate them on channel relation. Finally, we use pointwise convolution to integrate these two aspects of information and obtain an output of the same size as the original input. The formula is expressed as follows:

\[
output = \text{PConv}_C([U_{sp}, U_{ch}]),
\]

where \( \text{PConv}_C \) is the point-wise convolutional layer. \( n \) is the output feature map number and \( C \) is the number of input feature maps, and \( output \) is the output of the SE layer.

2) PARALLEL ADDITION

In the parallel space-channel-wise addition method (SCAdd), we obtain \( U_{sp} \) and \( U_{ch} \) in the same way as that in the parallel concatenation method. We can directly add them point-to-point to obtain the output of each ACB because they have the same shape.

\[
output = U_{sp} + U_{ch},
\]

3) SEQUENTIAL CHANNEL-SPACE

In the sequential channel-space method (S1C2), we use the channel-wise and SSE methods sequentially to process input \( U \). The formula is expressed as follows:

\[
output = \text{SE}_{sp}(\text{SE}_{ch}(input)),
\]

where \( input \) is the input of the SE layer. \( \text{SE}_{sp} \) and \( \text{SE}_{ch} \) represent the SSE and CSE operations, respectively.

4) SEQUENTIAL SPACE-CHANNEL

C1S2 is similar to the channel-space method. The difference is that this method executes the SSE method first and the CSE method next. The formula is expressed as follows:

\[
output = \text{SE}_{ch}(\text{SE}_{sp}(input)).
\]

**IV. EXPERIMENTAL CONFIGURATION**

In this work, we select the TIMIT corpus [39] as the clean speech. TIMIT contains 6300 sentences, of which 10 sentences are spoken by each of the 630 speakers from 8 major dialect regions of the evaluation. We remove the dialect sentences (the SA sentences) from its training set and use the remaining 3696 utterances for training. The TIMIT core test containing 192 utterances is used as the test set. The training set is mixed with four kinds of noises (babble, factory1, destroyerops, and destroyerengine) at three SNR levels (−4, 0 and 5dB). And in the test set, we additionally choose non-stationary noise factory2 and stationary noise, and two other SNRs (−10 and 10 dB). All noises used come from Noisex92 [31]. For evaluation, we choose short-time objective intelligibility (STOI) [29] and perceptual evaluation of speech quality (PESQ) [30]. STOI is positively related to subjective speech intelligibility, with a value range of 0 to 1. The larger the value is, the better the speech intelligibility is. PESQ focuses on evaluating the subjective quality of the perceived speech, with values between −0.5 and 4.5 [31]. Like STOI, the larger value indicates the clearer the speech.

We choose short-time fourier transform (STFT) to compute the spectral vectors. STFT uses a Hanning window with 256 points and an overlap interval of 128. Given that the 256-point STFT magnitude vector is symmetric, we only use half. All data are resampled at 8 kHz. Considering that the window length, the shift, and the normal length of vowels
are approximately 32, 16, and 99 ms, respectively, [40], we set the convolutional kernel in the time axis to 11 in all building blocks. Ultimately, the convolutional kernel can cover approximately 192 ms, which is approximate twice the vowel length. The kernel size in the frequency axis is 11. The filter number of each layer is 12-24-36-48-60-48-36-24-12-1.

In the training phase, the Adam optimizer [33] is used for parameter optimization, with a learning rate of 0.0002. We use 8 as the mini-batch size on the utterance-level for training, with mean squared error (MSE) as the loss function. For fair comparison, each model is trained for the same number of epochs. Then, the best model is selected from them for
TABLE 2. The STOI (%) and PESQ improvement of all models at unseen −10dB and Seen −5dB under different noises.

| Method          | Seen          | Unseen         |          |          |
|-----------------|---------------|----------------|----------|----------|
|                 | babble        | factory1       | destroyerengine | destroyercrops |
|                 | STOI | PESQ | STOI | PESQ | STOI | PESQ | STOI | PESQ | STOI | PESQ |
| RCED            | -5  | 17.45 | 23.72 | 27.46 | 42.47 | 34.39 | 43.04 | 23.71 | 45.16 | 19.82 | 38.01 | 20.18 | 51.41 |
|                 | -10 | 6.71  | 12.03 | 24.54 | 32.56 | 34.54 | 36.23 | 23.28 | 48.82 | 22.03 | 44.37 | 24.72 | 46.72 |
| ARCN-CSE        | -5  | 18.03 | 26.28 | 28.09 | 44.52 | 35.18 | 44.94 | 24.26 | 46.10 | 20.60 | 39.77 | 20.28 | 54.93 |
|                 | -10 | 7.91  | 14.29 | 25.65 | 32.56 | 36.03 | 37.68 | 23.54 | 48.82 | 22.35 | 42.96 | 23.95 | 52.46 |
| ARCN-SSE        | -5  | 20.24 | 27.56 | 29.54 | 47.26 | 36.32 | 46.20 | 25.11 | 47.74 | 21.39 | 39.76 | 21.06 | 53.52 |
|                 | -10 | 9.97  | 13.53 | 27.73 | 36.43 | 37.76 | 38.41 | 24.34 | 51.18 | 23.78 | 44.37 | 24.37 | 48.36 |
| ARCN-SCconcat   | -5  | 21.66 | 30.13 | 31.08 | 48.63 | 37.75 | 50.63 | 26.27 | 50.32 | 22.88 | 43.27 | 23.28 | 58.45 |
|                 | -10 | 10.89 | 17.29 | 30.48 | 38.76 | 40.63 | 41.30 | 26.06 | 53.54 | 27.04 | 47.89 | 28.27 | 58.20 |

V. ANALYSIS

A. ENHANCEMENT RESULTS

In Table 1, we show the STOI and PESQ scores for different models. Optimal values are marked in bold. Overall, the performance of all proposed models is better than that of the RCED in general. This fact shows the effectiveness of all the SE operations. In most scenarios, the four space-channel-wise SE models achieve higher PESQ scores than adding SSE only and adding CSE only, which proves that both channel and spatial information is helpful for performance gain. Among them, RCED adding SCconcat yields the most significant improvement for noisy utterances in terms of STOI and PESQ metrics in all cases. For example, in the babble noise, SCconcat provides 8.9% and 0.44 STOI and PESQ improvements over the noisy utterances on average, respectively.

Then we compare best-performed ARCN-SCconcat with the state-of-the-art [32]. We can find that in lower SNRs (i.e., 0dB), ARCN-SCconcat obtains significantly higher STOI values than [32]. However, as the SNR increases, the gap between ARCN-SCconcat and [32] gradually decreases, and the differences are negligible at 10 dB. Similar trends can be observed in terms of PESQ. We analyze that, under the condition of low SNRs, after the LSTM transformation, the embedding of the t-th frame and its z frames before and after (i.e., $t - z \sim t + z$) all still include some noise components. The mask obtained by calculating the correlation between these noisy embeddings and following transformations may be deviated from the ground truth, so that the enhanced spectrogram still has noise. However, when the SNR is high and the speech component dominates in embedding, the obtained mask can be much closer to the ground truth, thus improving the model performance. And our work can continuously filter out noises through multiple SE layers, so it can obtain satisfactory results at low SNRs.

B. GENERALIZATION CAPABILITY

For supervised training methods, generalization ability is an important aspect of performance evaluation. The generalization of the model is mainly evaluated from three perspectives, namely, noise, speaker, and SNR generalization ability. Next, the three generalization abilities are analyzed separately.

First is the noise generalization. In Table 1, we can see the evaluation for different models on unseen noises (i.e. stationary factory2 noise and un-stationary white noise). A trend similar to that in seen conditions can be observed. That is, the models combined with SE mechanisms perform better than the RCED itself. For example, in the factory2 noise condition, ARCN-CSE, ARCN-SSE, ARCN-SCconcat, ARCN-SCadd, ARCN-S1C2, and ARCN-C1S2 achieve average STOI improvements of 0.38, 0.55, 1.38, 0.61, 0.2, and 0.94, respectively, and average PESQ improvements of 0.05, 0.04, 0.1, 0.05, 0.05 and 0.03 respectively.

To clearly show the SNR generalization capability of the proposed models, we illustrate the percentage growth of the models (i.e., RCED, ARCN-CSE, ARCN-SSE, and ARCN-SCconcat, which have the best performance among the four space-channel-wise models) compared with the unprocessed utterances under trained and untrained SNRs. We use −10 dB and 10 dB as the untrained SNRs in the experiments. Given that the noisy utterances themselves are already clear enough under 10 dB, the models have little room for the improvement of the STOI and PESQ metrics. Therefore, we only list the results at −10 dB. As for seen SNR, we randomly select −5 dB to show for comparison. The results are illustrated in Table 2. We can see that under most noises, the performance increase percentage at −10dB is slightly lower than that at −5dB. This phenomenon is normal because that model
can remember some features that appeared during training. Moreover, noisy utterances at $-10$ dB contain much more noise components, making the difficulty of enhancement increases. And in babble noise, the improvement ratio at $-5$dB is much lower than that in $-10$dB. We hypothesize it is because that babble noise is more complicated and thus more difficult to eliminate. Noticeably, in some cases, such as factory2, the growth rate under unseen $-10$dB is higher than seen $-5$dB, which proves the SNR generalization capability of proposed mechanisms.

As for speaker generalization, given that the TIMIT core test set contains all the SX and SI sentences read by 24 speakers (2 male and 1 female form each dialect region), all utterances for the test are read by untrained speakers. Thus, the experimental results can effectively prove that the models have good speaker generalization capability.

### C. THE LOCATION OF SE

In this work, we add the attention mechanism to both of the encoder and the decoder of RCED for the following two reasons: First, SE can excite informative features in the early layers and becomes specialized in later layers. Second, its benefit can be accumulated through the network, more SE layers can lead to better performance [34].

To further verify the effect of the SE module, the decoder for speech enhancement tasks through experiments, we respectively add the proposed $\text{SCconcat}$ mechanism to the encoder of RCED only (denoted as $\text{RCED-SCconcat-en}$), the decoder of RCED only (denoted as $\text{RCED-SC-de}$), and both of them (denoted as $\text{RCED-SC-concat}$). The results are given in Table 3, and each value corresponds to the average result across five SNR levels ($-10, -5, 0, 5,$ and $10$ dB) and six noise types (babble, destroyerops, destroyerengine, factory1, factory2 and white). From the table, we can find that the overall performance from low to high is $\text{RCED}$, $\text{SC-concat-en}$, $\text{SC-concat-de}$ and $\text{SC-concat-end}$. That means SE affects whether it is added in the encoder or the decoder. It is easy to understand why SE affects decoder as it can filter out more valuable information to decoder, thus improving the expressive power of the model. As for encoder, we analysis the reasons for the performance improvement as follow: The encoding process is used to
generate redundant information. If we consider the limiting case, i.e. the information generated by the encoder is too redundant, then no matter how good the filtering effect of the decoder is, it is difficult to completely capture useful information from much useless information. As a result, the generated spectrogram still contains some noise components. Therefore, it is necessary to add SE to the encoder as well.

### D. SE Generalization

The generalization of SE mechanisms indicates that not only can they play a role in the RCED, but also improve the performance of other CNN-based models. To prove it, we test the performance of a simple CNN and its SE equivalents. The detailed description of the CNNs we used with and without SE mechanisms is shown in Table 4 (a) and (b). The STOI and PESQ results are shown in Fig. 3 (a) and (b), respectively.

Considering that many state-of-the-art CNN speech enhancement methods have shortcuts [24], [25], we also investigate the effect of the proposed SE mechanisms when combined with shortcut-based CNNs. We build a shortcut-based convolutional network (SCN) that adds shortcuts between the corresponding layers in the encoder and decoder of the CNN used before and then evaluate its performance with and without SE mechanisms. The detailed description is presented in Table 4 (c) and (d). The STOI and PESQ scores of all models are shown in Fig. 3 (c) and (d).

The results show that models with SE outperform the normal version in most cases, indicating that the introduction of the proposed SE is beneficial. Therefore, SE can be combined with a wide range of CNN-based models for performance gain.

### E. SSE 1 VS C

The output shape of the SSE mechanism is \(H^*W\), and each value corresponds to a time-frequency point. The attention operation multiplies this value with all the values at the corresponding time-frequency embedding. This means that each dimension of the time-frequency point embedding is filtered according to the same weight. However, every dimension of the time-frequency point embedding represents a kind of feature. Some of these features are important, while some are not. If they are treated equally, then the model cannot make full use of informative features and suppress redundant information, thereby limiting the expressive power of the model.

To solve this problem, we change the output of the SSE operation (with the shape of \(H^*W\)) from 1 (ARCN-SSE-1) to \(C\) (ARCN-SSE-C) by setting subscript \(n\) in (3) from 1 to \(C\). For example, the convolutional layer in the first building block has 12 output feature maps (i.e. in that building block, \(C = 12\)). Take them as the input of the following SE layer, and then generate 12 attention maps. That is, the \(C\) in each SE layer is 12-24-36-48-60-48-36-24-12, which is consistent with the number of output feature maps. In Table 5, we evaluate the performance of ARCN-SSE-1 and ARCN-SSE-C under seen and unseen noises. Each value corresponds to the average result across five SNR levels (\(-10, -5, 0, 5,\) and \(10\) dB). In all cases, ARCN-SSE-C yields significant improvements over SSE-1 in terms of STOI and PESQ scores. For example, in the condition of seen babble noise, the STOI and PESQ scores improve by 0.86 and 0.05, respectively. In unseen factory2 noise, the STOI and PESQ scores improve by 0.45 and 0.04, respectively. This result shows that our intuition is correct, that is, each dimension of the embedding has a different importance to the time-frequency prediction. Notably, although the performance of ARCN-SSE-C improve significantly, the number of its parameters also increased. The parameters of ARCN-SSE-1 and ARCN-SSE-C are 1.171 and 1.315 million, respectively, with a growth rate of 12.3%, which is low. However, in the models with many channels, the amount of calculation may increase significantly. Therefore, when the performance requirements are high, we consider the use of ARCN-SSE-C.

| Model       | Layer configuration                                                                 | Parameters                                                                 |
|-------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| (a) CNN-noSE| Encoder: (Conv, BN, ReLU) × 5                                                        | Number of filters: 16-32-64-128-256                                       |
|             | Decoder: (DeConv, BN, ReLU) × 4, DeConv                                              | -128-64-32-16-1                                                           |
| (b) CNN-SE  | Encoder: (Conv, BN, ReLU, SE) × 5                                                    | Kernel size: (3, 3)                                                       |
|             | Decoder: (DeConv, BN, ReLU, SE, Concat) × 4, DeConv                                  | Stride: (1, 2)                                                            |
| (c) SCN-noSE| Encoder: (Conv, BN, ReLU) × 5                                                        |                                                                           |
|             | Decoder: (DeConv, BN, ReLU, SE, Concat) × 4, DeConv                                  |                                                                           |
| (d) SCN-SE  | Encoder: (Conv, BN, ReLU, SE) × 5                                                    |                                                                           |
|             | Decoder: (DeConv, BN, ReLU, SE, Concat) × 4, DeConv                                  |                                                                           |

| Model       | ARCN-SSE-1 | ARCN-SSE-C | ARCN-SSE-1 | ARCN-SSE-C |
|-------------|------------|------------|------------|------------|
| Babble      | 73.91      | 74.77      | 2.29       | 2.34       |
| Destroyerengine | 81.94  | 82.79      | 2.52       | 2.57       |
| Destroyerops | 79.71      | 80.13      | 2.51       | 2.54       |
| Factory1    | 76.65      | 77.39      | 2.39       | 2.43       |
| Factory2    | 82.75      | 83.20      | 2.59       | 2.63       |
| White       | 79.36      | 80.66      | 2.43       | 2.46       |
FIGURE 4. An example of spectrogram (a) clean speech (b) noisy speech (c) enhanced by RCED (d) enhanced by ARCN-CSE (e) enhanced by ARCN-SSE (f) enhanced by ARCN-SCconcat. The noisy speech is obtained by mixing a TIMIT utterance spoken by an unseen speaker with babble noise under the condition of 0dB.

TABLE 6. Parameter quantity comparison.

| Model          | Parameters (million) | Rate (%) |
|----------------|----------------------|----------|
| RCED           | 1.144                | -        |
| ARCN-CSE       | 1.156                | 1.04     |
| ARCN-SSE       | 1.171                | 2.36     |
| ARCN-SCconcat  | 1.207                | 5.51     |
| ARCN-SCadd     | 1.183                | 3.41     |
| ARCN-S1C2      | 1.183                | 3.41     |
| ARCN-C1S2      | 1.183                | 3.41     |

In circumstances that require high processing speed, such as real-time applications, we opt for ARCN-SSE-1, which has few parameters.

F. PARAMETER COMPARISON

Table 6 shows the number of parameters and growth rate each proposed model compared with those of the RCED. The ARCN-SCconcat network with the most parameters increased by 5.51%. Therefore, the introduction of SE mechanisms only increases the computational burden by a small fraction and is perceived as lightweight mechanisms.

Besides, CNN has a weight sharing and local connection mechanism that enables the reduction of the number of parameters. As mentioned previously, modeling by combining SE and CNN-based models may have great potential in application scenarios that have requirements on model size, such as embedded systems.

G. QUALITATIVE ANALYSIS

1) ENHANCEMENT RESULTS

To show the denoising effect of each model more clearly, we select a TIMIT utterance spoken by an unseen speaker corrupted by the babble noise at 0dB and then draw the spectrograms for each case (clean speech, noisy speech, enhanced by RCED, enhanced by ARCN-CSE, enhanced by ARCN-SSE and enhanced by ARCN-SCconcat) as shown in Fig. 4.

From the figure, it can be noticed that all models can effectively remove the noise components. However, there is still some remaining background noise in (c). And some of the recovered speech structures especially in high frequency is somewhat rough. As for adding CSE only (d) and SSE only (e), we can see that both of them can eliminate the background noise and restore the speech component effectively. This proves that the addition of SE enables the model to pay more attention to speech. The enhancement results of (d) and (e) are similar but differ in details. Therefore, when adding SCconcat which combines both, the model can obtain the information obtained by CSE and SSE at the same time and complement each other. In this way, the model can achieve the elimination of noise and the preservation of speech details, making the enhancement effect better.

When comparing all the figures, we find that the spectrogram obtained by RCED (c) contains the most noise components and the spectrogram of ARCN-SCconcat (f) is closest to that of the clean utterance. This finding matches the results that RCED and ARCN-SCconcat produce the lowest and the highest metric (i.e. STOI and PESQ) scores respectively in Table 1.

2) CHANNEL-WISE SE VISUALIZATION

Each channel can be regarded as a set of certain characteristics of all time-frequency points. Some channels concentrate on speech while some concentrate on noise. The purpose of channel-wise SE is to give greater weights to the channels corresponding to speech and smaller weights to those
FIGURE 5. The visualized images of feature maps. (a) Clean speech (b) noisy speech (c) the feature map with min weight in the first ACB (d) the feature map with max weight in the first ACB (e) the feature map with min weight in the last ACB (f) the feature map with max weight in the last ACB.

FIGURE 6. The visualization of masks. (a) Clean speech (b) actual mask (c)-(i) are intermediate attention maps generated by spatial SE layers in each building block, and they are in order from lower-level to higher-level layers (closer to the input layer) to higher-level layers (closer to the output level). The input noisy speech is obtained by mixing a TIMIT utterance spoken by an unseen speaker with babble noise under the condition of 0dB.
corresponding to noise. To prove whether channel-wise SE achieves its purpose, in Fig. 5, we respectively visualize the feature maps assigned to the minimum and the maximum weights in the first and the last building block of ARCN-SSE. And the input utterance we used is the same as that in 1) ENHANCEMENT RESULTS.

(c) and (d) are the feature maps with the minimum and maximum weights in the first building block. We can see that although (c) is dominated by the noise components, there still exist some speech components in the black solid frame, and in (d) clear speech texture can be seen, even if they are broken. This is because in the early layer, although SE has a recalibration effect, it is not strong enough. In the last building block, the feature map with the minimum and maximum weights should be similar with the noise and clean speech as the recalibration effect is consistently accumulated through the entire network, and the pattern of (e) and (f) confirm this. Most of (e) is a noise component, and there is almost no speech component. In (f), we can observe a more coherent and clearer outline of the clean speech.

3) SPATIAL SE VISUALIZATION
Spatial SE assigns a weight to each time-frequency point.

Therefore, inputting a noisy speech, the generated attention map (with the shape of $H^*W$, each value corresponds to a time-frequency point) should be similar with the distribution of the clean speech. In Fig. 6, we visualize the attention map in each intermediate building block of ARCN-SSE. We use the same input utterance with that in 1) ENHANCEMENT RESULTS.

In the lower-level layer (c), though the color of the part corresponding to speech is slightly deeper than the surrounding, the colors of the whole picture are similar. Therefore, the speech and noise are still not distinguished. In (d), we can find the sporadic distribution of the speech component, but it is still very broken. Then obvious patterns of the speech begin to appear in the following layers (e and f), but still incoherent. The vertical bars appearing in (g) are similar to the outline of the clean speech, but with little texture of the speech, which is further reflected in (h) and (i).

In general, from Fig. 6 we can see that in lower-level layers, the spatial SE mechanism is still unable to recognize speech and noise very accurately. As multiple spatial SE layers are accumulated, the ability to distinguish between speech and noise is enhanced, which meet the conclusion in [34]. In this way, the model can filter out the noise components and retain the speech components, to better restore the enhanced speech spectrogram.

VI. CONCLUSION
In this paper, we propose ARCN, which combines the attention mechanism and the RCED model for speech enhancement. Attention weight assignment is achieved through the SE mechanism, which improves speech enhancement performance by emphasizing valuable information. The original SE mechanism assigns weights to channels according to global information. Considering that spatial information is also of great importance, we propose to assign a weight to each time-frequency point through SSE. We further boost the performance by concurrently exploiting both of CSE and SSE using four different ways. The experimental results prove that all proposed SE mechanisms can effectively improve the model performance without adding a heavy computational burden and can be generalized well to untrained noises, SNRs, and speakers. The best results are obtained by concatenating the two aspects (i.e., space and channel) of information. In addition to RCED, we also combine SE with other CNN-based models, thereby also achieving performance improvement. This means that the SE mechanism can be treated as a plugin and can be introduced in other CNN-based models for performance gain.

REFERENCES
[1] C. Donahue, B. Li, and R. Prabhavalkar, “Exploring speech enhancement with generative adversarial networks for robust speech recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Calgary, AB, Canada, Apr. 2018, pp. 5024–5028.
[2] C. K. A. Reddy, N. Shankar, G. S. Bhat, R. Charan, and I. Panahi, “An individualized super-Gaussian single microphone speech enhancement for hearing aid users with smartphone as an assistive device,” IEEE Signal Process. Lett., vol. 24, no. 11, pp. 1601–1605, Nov. 2017.
[3] J. Liu, X. Yang, M. Zhu, and B. He, “Speech enhancement with stacked frames and deep neural network for VoIP applications,” in Proc. 17th Int. Conf. Opt. Commun. Net. (ICOCON), Zhuhai, China, Feb. 2019, Art. no. 1104808.
[4] M. K. Hasan, S. Salahuddin, and M. R. Khan, “A modified a priori SNR for speech enhancement using spectral subtraction rules,” IEEE Signal Process. Lett., vol. 11, no. 4, pp. 450–453, Apr. 2004.
[5] B. Xia and C. Bao, “Wiener filtering based speech enhancement with weighted denoising auto-encoder and noise classification,” Speech Commun., vol. 60, pp. 13–29, May 2014.
[6] Y. Ephraim, “Statistical-model-based speech enhancement systems,” Proc. IEEE, vol. 80, no. 10, pp. 1526–1555, Oct. 1992.
[7] K. Hermus, P. Wambacq, and H. Van Hamme, “A review of signal subspace speech enhancement and its application to noise robust speech recognition,” EURASIP J. Adv. Signal Process., vol. 2007, no. 1, Dec. 2006, Art. no. 045821.
[8] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “A regression approach to speech enhancement based on deep neural networks,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 23, no. 1, pp. 7–19, Jan. 2015.
[9] L. Sun, J. Du, L.-R. Dai, and C.-H. Lee, “Multiple-target deep learning for LSTM-RNN based speech enhancement,” in Proc. Hands-Free Speech Commun. Microphone Arrays (HSCMA), San Francisco, CA, USA, 2017, pp. 136–140.
[10] A. Pandey and D. Wang, “A new framework for CNN-based speech enhancement in the time domain,” IEEE/ACM Trans. Audio, Speech,Lang. Process., vol. 27, no. 7, pp. 1179–1188, Jul. 2019.
[11] A. Kumar and D. Florencio, “Speech enhancement in multiple-noise conditions using deep neural networks,” in Proc. Interspeech, Sep. 2016, pp. 1–5.
[12] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. J. A. P. A. Lee, “Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement,” in Proc. Interspeech, Dresden, Germany, Sep. 2015, pp. 1508–1512.
[13] T. G. Kang, J. W. Shin, and N. S. Kim, “DNN-based monaural speech enhancement with temporal and spectral variations equalization,” Digit. Signal Process., vol. 74, pp. 102–110, Mar. 2018.
[14] F. Weninger, “Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR,” in Proc. Int. Conf. Latent Variable Anal. Signal Separat., Liberec, Czech Republic: Springer, 2015, pp. 91–99.
[15] D. Shan, X. Zhang, C. Zhang, and L. Li, “A novel encoder-decoder model via NS-LSTM used for bone-conducted speech enhancement,” IEEE Access, vol. 6, pp. 62638–62644, 2018.
A. G. Roy, N. Navab, and C.-H. Lee, “Densely connected progressive learning for LSTM-based speech enhancement,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5054–5058.

E. W. Healy, S. E. Yoho, Y. Wang, and D. Wang, “An algorithm to improve speech recognition in noise for hearing-impaired listeners,” *J. Acoust. Soc. Am.*, vol. 134, no. 4, pp. 3029–3038, Oct. 2013.

S.-W. Fu, Y. Tsao, and X. Lu, “SNR-aware convolutional neural network modeling for speech enhancement,” in *Proc. Interspeech*, Sep. 2016, pp. 3768–3772.

T. Koumovkovs and J. Malek, “Single channel speech enhancement using convolutional neural network,” in *Proc. Int. Workshop Electron. Control. Meas., Signals Appl. Mechatron. (ECMSM)*, May 2017, pp. 1–5.

J.-C. Hou, S.-S. Wang, Y.-H. Lai, Y. Tsao, H.-W. Chang, and H.-M. Wang, “Audio-visual speech enhancement using multimodal deep convolutional neural networks,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 2, pp. 117–128, Apr. 2018.

G. S. Bhat, N. Shankar, C. K. A. Reddy, and I. M. S. Panahi, “A real-time convolutional neural network based speech enhancement for hearing impaired listeners using smartphone,” *IEEE Access*, vol. 7, pp. 78421–78433, 2019.

S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai, “Raw waveform-based speech enhancement by fully convolutional networks,” in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 006–012.

S. R. Park and J. W. Lee, “A fully convolutional neural network for speech enhancement,” in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 1993–1997.

K. Tan and D. Wang, “A convolutional recurrent neural network for real-time speech enhancement,” in *Proc. Interspeech*, Hyderabad, India, Sep. 2018, pp. 3229–3233.

T. Grzywalski and S. Dugas, “Using recurrences in time and frequency within U-net architecture for speech enhancement,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 6970–6974.

S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 3–19.

M. I. Posner and S. E. Petersen, “The attention system of the human brain,” *Annu. Rev. Neurosci.*, vol. 13, no. 1, pp. 25–42, Mar. 1990.

J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 577–585.

B. Wang, K. Liu, and J. Zhao, “Inner attention based recurrent neural networks for answer selection,” in *Proc. 44th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Berlin, Germany, 2016, pp. 1288–1297.

P. Loyola, C. Liu, and Y. Hirate, “Modeling user session and intent with an attention-based encoder-decoder architecture,” in *Proc. 11th ACM Conf. Recommeder Syst.*, Como, Italy, Aug. 2017, pp. 147–151.

J. B. Fritz, M. Elhilali, S. V. David, and S. A. Shamma, “Auditory attention—focusing the searchlight on sound,” *Current Opinion Neurobiol.*, vol. 17, no. 4, pp. 437–455, Aug. 2007.

X. Hao, C. Shan, Y. Xu, S. Sun, and L. Xie, “An attention-based neural network approach for single channel speech enhancement,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 6895–6899.

O. Ernst, S. E. Chazan, S. Gannot, and J. Goldberger, “Speech dereverberation using fully convolutional networks,” in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Roma, Italy, Sep. 2018, pp. 390–394.

J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

X. Cheng, X. Li, J. Yang, and Y. Tai, “SESR: Single image super resolution with recursive squeeze and excitation networks,” in *Proc. 24th Int. Conf. Pattern Recognit.* (ICPR), Beijing, China, Aug. 2018, pp. 147–152.

A. G. Roy, N. Navab, and C. Wachinger, “Recalibrating fully convolutional networks with spatial and channel ‘squeezes & excitation’ blocks,” *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 540–549, Feb. 2019.

A. G. Roy, N. Navab, and C. Wachinger, “Concurrent spatial and channel ‘squeezes & excitation’ in fully convolutional networks,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Springer, Granada, Spain, 2018, pp. 421–429.

K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1026–1034.