Speech Communication 136 (2022) 1–13

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

PACDNN: A phase-aware composite deep neural network for speech enhancement

Mojtaba Hasannezhad [a,*], Hongjiang Yu [a], Wei-Ping Zhu [a], Benoit Champagne [b]

[a] Department of Electrical and Computer Engineering, Concordia University, Canada
[b] Department of Electrical and Computer Engineering, McGill University, Canada

ARTICLE INFO

Keywords:
Speech enhancement
Deep neural network
Model complexity
Spectral mask
Phase derivative

ABSTRACT

Most of the current approaches for speech enhancement (SE) using deep neural network (DNN) face a number of limitations: they do not exploit information contained in the phase spectrum while their high computational complexity and memory requirements make them unsuited for real-time applications. In this paper, a new phase-aware composite deep neural network (PACDNN) is introduced to address these challenges. Specifically, magnitude processing with spectral mask and phase reconstruction with phase derivative are proposed as key subtasks of the new network to simultaneously enhance the magnitude and phase spectra. Besides, the DNN is meticulously designed to take advantage of strong temporal and spectral dependencies of speech, while its components perform independently and in parallel to speed up the computation. The advantages of the proposed PACDNN model over some well-known DNN-based SE methods are demonstrated through extensive comparative experiments.

1. Introduction

Speech signals acquired in real-word environments are often corrupted by background noise. This degradation occurs in many applications, such as speech recognition, hearing prosthesis, voice communications, smart home devices, etc. Speech enhancement (SE) aims to suppress the unwanted ambient noise contained in the acquired speech signal, either to improve its quality or as a preprocessing procedure to make these applications robust to various noises. A SE method can be either unsupervised or supervised. Traditional Wiener filtering (Abd El-Fattah et al., 2008; Wang and Chen, 2018) and statistical model-based methods (Martin, May 2002; Parchami et al., 2016) are two well-known classes of unsupervised methods, which rely on the statistical properties of speech and noise, and yield good performance when these properties are known or properly modeled. However, these methods suffer from performance degradation in real-world scenarios where the statistical properties are unknown or difficult to model, especially for non-stationary noise conditions.

In recent years, with the development of ever faster computing hardware and the availability of large datasets, supervised methods have received increasing attention in many areas. In particular, deep learning-based methods have achieved revolutionary progress in speech processing, including SE. The remarkable capability of DNN in modeling highly complex transformations has vastly advanced SE in adverse and variable acoustic scenarios. Moreover, a well-trained DNN can offer low latency processing, which is crucial to many real-time applications, such as hearing aids (Agnew and Thornton, 2000). Various DNN-based SE methods have been proposed during the past decade, as exposed in further details below.

Xu et al. (2014) utilized a multi-layer perceptron (MLP) to map the log-power spectrum of the noisy speech to the clean one. In this work, some critical MLP issues, such as the over-fitting and global variance normalization problems, were also investigated. Although the MLP model achieves very good SE results, it involves a large number of parameters, and thus has a high-complexity. Besides, MLP processes speech samples independently in the sense that it does not consider sequential information, whereas speech exhibits strong temporal dependencies. Chen and Wang (2017) adopted the long short-term memory (LSTM) network, a variation of recurrent neural networks (RNN), to model contextual information of speech sequentially along time, and showed that LSTM could keep track of such speaker dependent information under difficult noisy conditions. They also demonstrated that the LSTM network outperforms the MLP in generalizing the model to numerous speakers and noises. Recently, an LSTM network operating along both time and frequency was used to extract time–frequency patterns for low bit-rate audio restoration (Abbaszadeh, 2016). Although LSTM shows very good SE performance, it is considered a high

* Corresponding author.
E-mail addresses: m_hasann@encs.concordia.ca (M. Hasannezhad), ho_yu@encs.concordia.ca (H. Yu), weiping@ece.concordia.ca (W.-P. Zhu), benoit.champagne@mcgill.ca (B. Champagne).

https://doi.org/10.1016/j.specom.2021.10.002
Received 3 April 2021; Received in revised form 25 July 2021; Accepted 10 October 2021
Available online 30 October 2021

complexity model. To alleviate this issue of LSTM, two of its variations, namely the gated recurrent unit (GRU) (Dey and Salemt, 2017) and simple recurrent unit (SRU) (Cui et al., 2020), have been recently employed for SE. However, while GRU and SRU provide efficient implementations of LSTM, they do not perform as well as LSTM in the SE application.

Park and Lee (2016) investigated convolutional neural network (CNN) for SE and compared its required number of parameters with that of MLP and LSTM. In particular, they showed that these three methods almost give the same SE performance although CNN requires a smaller number of parameters. However, this study only considers the number of parameters, while the actual complexity and implementation cost also depend on the memory footprint, which can be significantly larger for CNN than for LSTM and MLP. It is also noted that CNN was originally conceived to capture local information from an image, while speech spectrograms generally exhibit non-local correlations. Moreover, the CNN network's max-pooling layers only retain the coarse information of its input. Consequently, a generative model with no max-pooling layer but containing instead a stack of dilated causal convolutional layer was introduced in Oord et al. (2016). This model expands the CNN filters' receptive field without adding more complexity to the model. Inspired by this work, a fully-convolutional model in the frequency domain was introduced in Ouyang et al. (2019) showing promising SE results.

In contrast to the aforementioned stand-alone methods, some recent studies have considered a combination of networks as the learning engine for SE. Tan and Wang (2018) introduced a convolutional recurrent neural network (CRN) as an encoder–decoder network for SE. They also extended CRN by introducing a gated convolutional recurrent network and obtained better SE results in Tan and Wang (2019). Some other CRN-based networks operating in the frequency and time domain were proposed in Zhao et al. (2018), Hsieh et al. (2020), respectively. In Hu et al. (2020), a deep complex CRN was introduced in which CNN and RNN are designed to emulate complex-valued targets. The merits of this model have been shown in terms of both objective and subjective metrics. Although the CRN model yields good SE results, Strake et al. (2020) argued that the internal relations and local structures of CNN feature maps are ravished due to the reshaping of data among different CRN components. Thus, they employed convolutional LSTM for SE, where the fully-connected mappings in LSTM are replaced with convolutional mappings. Based on this argument, another model block named *gruCNN* was recently utilized for SE in Shifas et al. (2020), where recurrency is added to feature extracting CNN layers. These combined networks achieved good SE results; however, they all exhibit very high complexity models, and some of them (due to their non-causal formulation) introduce additional latency. Moreover, CRN-based methods perform well when the training and testing datasets are the same but break down on unseen datasets (Pandey and Wang, 2020).

Although the above-mentioned methods have achieved remarkable performance in SE, most of them only focus on speech magnitude enhancement and leave the phase unprocessed. This is because the phase spectrogram is highly unstructured, rendering its estimation by DNN difficult. However, the role and importance of phase enhancement in the context of speech enhancement was pointed out in Krawczyk and Gerkmann (2014), and consequently, different phase-aware methods were proposed. One of the earliest attempts to incorporate phase information into the magnitude processing with DNN was presented in Erdogan et al. (2015) where a phase-sensitive mask (PSM) was introduced. However, this approach mainly exploits the PSM to process the speech magnitude and employs the noisy phase in the speech reconstruction. A complex ideal ratio mask (cIRM) was introduced in Williamson et al. (2015) where the mask is divided into real and imaginary components to enhance the complex spectrogram. Unfortunately, use of the cIRM introduces distortion in the enhanced speech due to the lack of recognizable patterns in its imaginary component (Yin et al., 2020; Hasannezhad et al., 2020a). Direct estimation of the complex spectrogram was alternatively proposed in Fu et al. (2017a), Tan and Wang (2018), Ouyang et al. (2019), Tan and Wang (2019), where the DNN is employed to estimate the real and imaginary parts of the clean speech complex spectrogram from those of the noisy speech. However, these methods require large datasets to learn an accurate mapping function; besides, their performance on unseen data might be worse than a simple spectral magnitude mapping method (Pandey and Wang, 2020). Yin et al. (2020) introduced a phase and harmonics-aware model for noise reduction, where a two-stream DNN architecture with information exchange between the magnitude and phase spectra is proposed to recover the complex spectrogram of the clean speech. Since the phase spectrogram itself has an irregular structure, researchers have also investigated alternative quantities derived from the phase that exhibit a similar structure as the magnitude for speech reconstruction (Mowlaee and Saeidi, 2014). Takamichi et al. (2018) tried to reconstruct phase based on the DNN-estimated amplitude. These authors introduced a von-Mises-distribution DNN for phase reconstruction with a loss function between the predicted and actual group delay (GD). In their subsequent work (Takamichi et al., 2020), they used a directional-statistics DNN in the same framework, and introduced a sine-skewed generalized cardioid distribution DNN to model GD. Zheng and Zhang (2018) presented a phase-aware model to jointly process the magnitude and phase spectrogram, where the estimated magnitude is obtained with a spectral mask, and the phase is reconstructed through a phase derivative (PD), specifically the so-called instantaneous frequency deviation (IFD). Experimental results demonstrate that this phase-aware model performs better than approaches based on the cIRM or the magnitude-only mask. Nevertheless, it uses MLP and LSTM to estimate the target, which limits the attainable accuracy while incurring a high computational cost. These networks are used in the estimation, which may limit target accuracy while incurring high computational costs. Besides, while the IFD is used effectively in phase reconstruction, other PDs such as GD could potentially lead to superior performance.

In a preliminary study (Hasannezhad et al., 2021), we proposed a composite model integrating CNN and LSTM for SE. Specifically, this model employs improved LSTM and CNN structures to exploit a complementary set of features containing spectral and temporal contextual information of speech and thus outperforms some known DNN-based SE methods. In this paper, we further investigate this model by introducing new ideas and processing modules for both phase and magnitude enhancements. Inspired by Zheng and Zhang (2018), we present a new model called phase-aware composite deep neural network (PACDNN) that involves two subtasks: magnitude processing with a spectral mask and phase reconstruction with PD, where a DNN estimates both targets simultaneously. We investigate different types of masks and PDs as well as their possible combinations to select the best targets for the DNN. Our analysis and experimental studies reveal that the proposed PACDNN model yields a improved SE performance compared to several existing DNN based methods while exhibiting a significantly lower computational complexity and memory footprint.

The rest of this paper is organized as follows: Section 2 describes the proposed PACDNN model and its components. Then, Section 3 presents experimental presents the experimental methodology along with comparative results and discussion. Finally, the paper is concluded in Section 4.

2. Proposed PACDNN model

A high-level block diagram of the proposed PACDNN model is shown in Fig. 1. The composite model integrates CNN and LSTM streams to extract a complementary set of features that are then transformed into the network targets. The composite model input consists of the noisy speech, while its output includes a spectral mask and PD. The mask and PD are calculated and set used as target model output in the training stage. The clean speech is reconstructed using the estimated mask and PD in the testing stage. The individual components of the PACDNN model are discussed in the following.

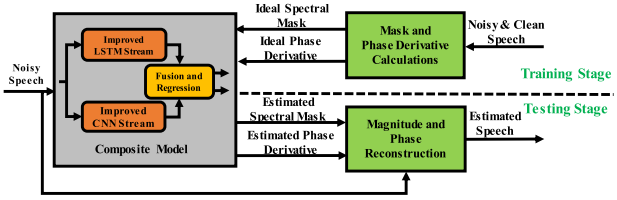

Fig. 1. High-level PACDNN model.

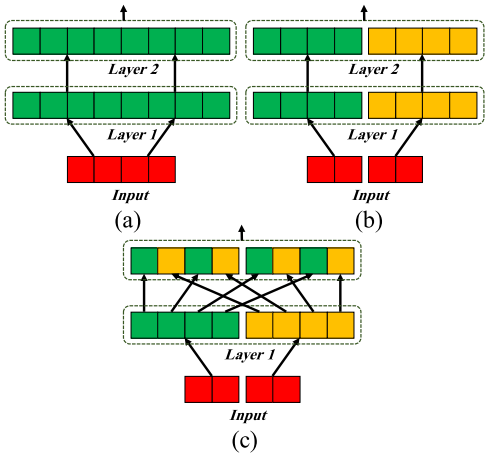

Fig. 2. 2-layer Grouped LSTM: (a) no grouping; (b) grouped with K=2; (c) grouping and representation rearrangement.

2.1. Composite model

2.1.1. Improved LSTM stream

Speech spectrogram exhibits strong temporal dependencies that can be useful for SE. LSTM can model these long-term dependencies since it treats the input frames as a sequence. Specifically, it can model the changes over time and learn temporal dynamics of speech (Wang and Chen, 2018). LSTM is made up of a memory cell and three control gates: forget, input, and output. Denoting by M and N the dimensions of the input vector and cell state, respectively, the number of trainable parameters of an LSTM is $4 \cdot (N^2 + NM + N)$ (Dey and Salemt, 2017).

One critical issue with an LSTM network is its high complexity, which stems from parameter redundancy in the weight and recurrent matrices. The former transforms feature representations while the latter transfers hidden states between consecutive steps. To circumvent these redundancies, a grouped recurrent network was introduced in Gao et al. (2018). Consider a two-layer LSTM network, as illustrated in Fig. 2(a). As known, the number of parameters of LSTM for a single gate while neglecting biases is $(N^2 + NM)$. By splitting input and hidden layers into K groups performing independently, the number of parameters is reduced by a factor K as follows,

$$K \cdot \left(\left(\frac{N}{K} \right)^2 + \frac{N}{K} \cdot \frac{M}{K} \right) = \frac{N^2 + N \cdot M}{K} \tag{1}$$

Such a grouped network is shown in Fig. 2(b) where K=2. Although the model complexity is reduced, the grouping strategy deteriorates the network efficiency. Indeed, while the intra-group temporal dependencies are captured, the inter-group ones are lost since different groups cannot communicate. To tackle this issue, an alternative rearrangement was proposed in Gao et al. (2018) whereby different groups are connected. This technique can be implemented using basic tensor operations without introducing additional parameters, as shown in Fig. 2(c). Hence, we use this grouping and connecting rearrangement to reduce the model complexity while keeping the performance almost the same.

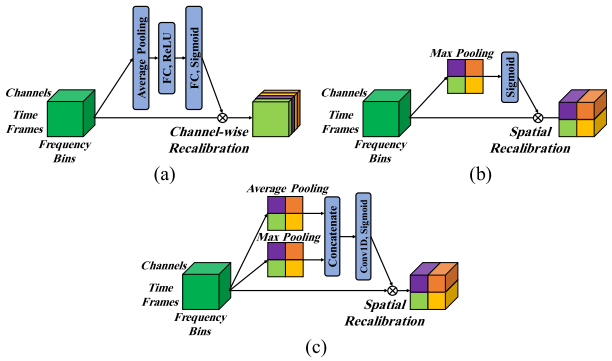

Fig. 3. Attention technique: (a) channel-wise with average-pooling; (b) spatial with max-pooling; (c) spatial with max and average-pooling.

2.1.2. Improved CNN stream

Dilated frequency convolution: CNN was initially designed for image classification. A conventional CNN is made up of pairs of convolutional and pooling layers followed by a fully-connected network. The purpose of the former is to extract features, while the latter accomplish classification.

Considering speech spectrogram as an image, its spectral contextual information can be exploited by a CNN. However, CNN captures only local information in its input due to the limited receptive field (the local area from the previous layer) of its kernels, while the speech spectrogram exhibits non-local correlations along the frequency axis. Dilated convolution was introduced to expand the receptive field of CNN kernels in image processing applications (Yu and Koltun, 2015). Following our recent study (Hasannezhad et al., 2021), we use a CNN with stacked dilated convolutions to capture non-local spectral correlations without increasing model complexity. Furthermore, residual learning and skip connection techniques are adopted to facilitate training and accelerate convergence. It is worth noting that this fully-convolutional CNN structure has no pooling layers.

Attention driven CNN: CNN contains many feature maps that may have different levels of significance. Accordingly, emphasizing informative feature maps improves the model performance. By recalibrating feature maps, an attention mechanism adaptively emphasizes the informative ones while suppresses the others. A successful attention mechanism termed *squeeze-and-excitation* (SAE) was introduced in Hu et al. (2018) focusing on channel relationships. In this approach, illustrated in Fig. 3(a), an average-pooling operation spatially aggregates the global information of each feature map to a channel descriptor in the *squeeze* stage. Then, a fully-connected network captures channel-wise dependencies by adjusting the descriptor in the *excitation* stage. Finally, the original feature maps are recalibrated by the excitation values, and the results are delivered to the subsequent layer. Inspired by SAE but aiming to take advantage of pixel-wise spatial information, Roy et al. (2018) introduced spatial SAE, illustrated in Fig. 3(b), wherein the squeeze operation is performed along channels while the excitation is spatial. Woo et al. (2018) introduced a convolutional block attention module, as shown in Fig. 3(c), which is a combination of channel-wise and spatial SAE. In this approach, both average and max-pooling are applied as the squeezer. The outputs of the squeezing modules are then concatenated and passed through a sigmoid activation function. Finally, the resulting weights are element-wise applied to the original feature maps. In this paper, we investigate the use of these attention techniques in the PACDNN model.

2.1.3. Regression

Referring to Fig. 1, the two parallel streams in the composite model of the proposed PACDNN exploit a complementary set of features,

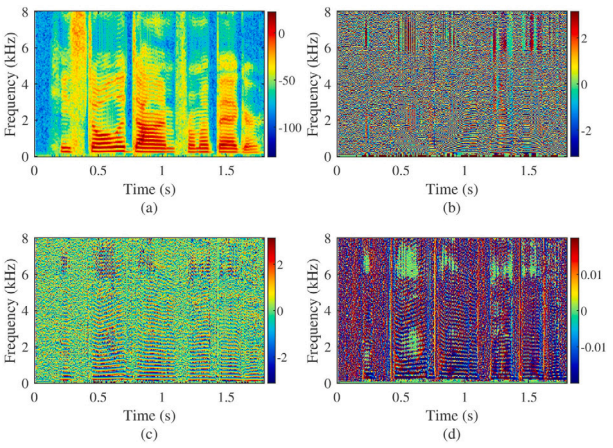

Fig. 4. Spectrogram plot of speech at sampling frequency 8 kHz: (a) magnitude (log scale); (b) phase; (c) IFD; (d) GD.

which are then transformed into a spectral mask and PD values. This transformation can be achieved by either a low-complexity CNN or an MLP network. These two DNN types have different attributes although they can both accomplish the required regression task. As stated in Fu et al. (2017b), CNN can model rapid fluctuations between contiguous elements while MLP fails to do so. Besides, CNN requires much less model parameters than MLP, while the latter requires way less memory for its computations. We will compare these two networks for the regression task from different perspectives, including SE performance and computational complexity.

2.2. Spectral mask and phase derivative calculation

As mentioned above, the targets of the composite model in the PACDNN consists of two parts, i.e., spectral mask and PD. The former is applied to the noisy magnitude spectrum to obtain the enhanced one, while the latter is employed to reconstruct the phase spectrum. Selecting appropriate targets is crucial for the final enhancement performance.

Considering the noisy speech $y(t)$ as the addition of the clean speech $s(t)$ and background noise $n(t)$, where t is the discrete time index, the time domain noisy speech can be transformed into the TF domain using short-time Fourier transform (STFT), that is,

$$Y(k,l) = S(k,l) + N(k,l) \tag{2}$$

where $Y(k,l)$, $S(k,l)$ and $N(k,l)$ denote the STFT spectrograms of noisy speech, clean speech and noise, respectively, with k and l indicating the frame index and frequency bin index. These complex spectrograms can be expressed in polar coordinates, i.e., magnitude and phase spectra. For instance, the spectrogram of the clean speech can be decomposed as follows,

$$S(k,l) = |S(k,l)|\, e^{i\phi_S(k,l)} \tag{3}$$

where ϕ and $|\cdot|$ denotes phase and magnitude, respectively. Our goal in this paper is to obtain the enhanced speech by jointly reconstructing the magnitude and phase spectra, given the noisy observations. The following introduces several popular masks as well as PDs. In our study, the enhancement performance is evaluated by considering different possible combinations of these masks and PDs.

2.2.1. Spectral mask

Inspired by the masking effects of the human auditory system (Wang and Chen, 2018), masking algorithms aim to retain the speech-dominant regions of the noisy speech in the TF domain while suppressing the noise-dominant ones. To this end, different masks have been introduced in the literature, as summarized below for this study:

Ideal ratio mask (IRM) (Srinivasan et al., 2006) is defined as the ratio between the energy of the clean speech and that of the noisy speech within a TF unit, under the assumption that the noise signal and the clean speech are uncorrelated. That is,

$$\mathrm{IRM}(k,l) = \left(\frac{|S(k,l)|^2}{|S(k,l)|^2 + |N(k,l)|^2}\right)^{\frac{1}{2}} \tag{4}$$

Spectral Magnitude Mask (SMM) (Wang et al., 2014), which is conceptually similar to IRM, is defined as the ratio of the spectral magnitude of the clean speech to that of the noisy speech, that is,

$$\mathrm{SMM}(k,l) = \frac{|S(k,l)|}{|Y(k,l)|} \tag{5}$$

Optimal Ratio Mask (ORM) (Liang et al., 2013) is derived based on the minimization of the mean square error (MSE) between the clean speech and the estimated speech. It is given by,

$$\mathrm{ORM}(k,l) = \frac{|S(k,l)|^2 + \Re(S(k,l)N^*(k,l))}{|S(k,l)|^2 + |N(k,l)^2| + 2\Re(S(k,l)N^*(k,l))} \tag{6}$$

where $*$ and $\Re$ denote the conjugate operation and the real part, respectively. The main difference between ORM and IRM is the presence of the term $\Re(S(k,l)N^*(k,l))$ in the former. Accordingly, ORM can be viewed as an improved version of IRM, which takes the correlation between the clean speech and noise into consideration.

Phase Sensitive Mask (PSM) (Erdogan et al., 2015) is defined as the real part of the ratio between the clean speech spectrogram and the noisy speech spectrogram, as given by,

$$\mathrm{PSM}(k,l) = \Re\left(\frac{S(k,l)}{Y(k,l)}\right) \tag{7}$$

Since we use the sigmoid as the output layer's activation function in PACDNN, training targets' values have to be limited to [0, 1]. Although IRM values fall in the desired range, those of ORM, PSM, and SMM are not limited to this range. Hence, these three masks' outlier values are truncated to [0, 1].

2.2.2. Phase derivative

Processing PD instead of the phase itself has been adopted in some phase-aware speech enhancement methods. In this regard, the instantaneous frequency (IF) (Stark and Paliwal, 2008) and group delay (GD) (Hegde et al., 2007) are two of the most well-known PDs.

Instantaneous frequency (IF) formally defined as the first time-derivative of the phase spectrum. In the case of spectrograms, IF can be approximated by the phase difference between two successive frames as,

$$\mathrm{IF}(k,l) = \mathrm{princ}\{\phi(k+1,l) - \phi(k,l)\} \tag{8}$$

where the function $\mathrm{princ}\{\cdot\}$ denotes the principal value operator, which projects the phase difference onto $[-\pi, \pi)$. Since the IF is limited to its principle value, the wrapping effects would occur along the frequency axis. To alleviate the problem, the instantaneous frequency deviation (IFD) has been adopted in Stark and Paliwal (2008) as follows,

$$\mathrm{IFD}(k,l) = \mathrm{IF}(k,l) - \frac{2\pi}{N}kL \tag{9}$$

where $\frac{2\pi}{N}kL$ is the center frequency of $\mathrm{IFD}(k,l)$.

It is demonstrated in Stark and Paliwal (2008) that the IF values track the frequencies of pitch harmonic peaks, while the IFD values capture pitch and formant structures as in the magnitude spectrogram. Similar findings are presented in Zheng and Zhang (2018), in which the authors reconstructed the phase from the estimated IFD for speech enhancement. They also showed that the IFD could be estimated with DNN as it exhibits similar patterns as the magnitude spectrogram, as illustrated in Fig. 4 (a, c).

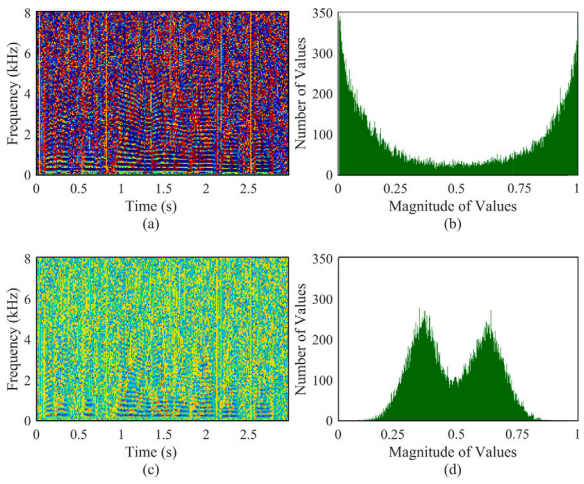

Fig. 5. Group delay regularization clean speech at sampling frequency 16 KHz: (a) GD spectrogram; (b) distribution of GD values; (c) RGD spectrogram; (d) distribution of RGD values.

Group delay (GD) is the negative of the derivative of the spectral phase with respect to frequency, as is given by:

$$\mathrm{GD}(k,l) = -\left[\phi(k,l+1) - \phi(k,l)\right] \tag{10}$$

The authors demonstrated that the GD function behaves like a squared magnitude response at the resonance frequency in Hegde et al. (2007). It also exhibits structural patterns similar to the magnitude spectrum, as seen from Fig. 4 (a, d). Moreover, the high-resolution property discussed in Prasad et al. (2004) reveals that GD has a higher resolving power than the magnitude spectrum. Specifically, the formants are resolved more accurately in the group delay spectrum when compared to the magnitude or linear prediction spectrum. Based on this finding, we infer that GD can also be employed as a training target of the DNN-based SE, in the same way as the widely-adopted magnitude targets or their variants.

As the mask and PD are jointly estimated with a single DNN, their values should be in the same range to balance the training process. We adopted the normalization scheme in Zheng and Zhang (2018), where the range of the spectral mask is truncated into $[0, 1]$, and the PD values are normalized as follows,

$$\mathrm{PD_n}(k,l) = \frac{1}{2\pi}\mathrm{PD}(k,l) + \frac{1}{2} \tag{11}$$

As can be seen from Fig. 5 (a, b), the distribution of the normalized GD values exhibits a U-shaped over the range $[0, 1]$, which renders their accurate estimation with DNN more difficult (Zheng and Zhang, 2018). As such, we propose to use the following transformation to regularize the normalized GD, namely,

$$\mathrm{RGD}(k,l) = \mu + \sqrt{2}\sigma \cdot \mathrm{erf}^{-1}\left(2\mathrm{GD}_n(k,l) - 1\right) \tag{12}$$

where $\mathrm{erf}^{-1}(\cdot)$ is the inverse error function, and σ and μ are set to 0.1 and 0.5, respectively. The RGD and its distribution are shown in Figs. 5 (c, d), where the values are pulled close to the center point (0.5), which makes the RGD a better training target.

2.3. Magnitude and phase reconstruction

In this subsection, we explain how to recover the magnitude and phase spectra from the spectral mask and the PD estimates.

2.3.1. Magnitude reconstruction

After obtaining the estimated spectral mask $\hat{M}(k,l)$ from the trained DNN, the magnitude reconstruction is accomplished by applying the spectral mask to the magnitude spectrogram of the noisy speech, i.e.,

$$|\hat{S}(k,l)| = \hat{M}(k,l)\,|Y(k,l)| \tag{13}$$

Typically, if a TF unit is speech dominated, $\hat{M}(k,l)$ will have a large value which help preserve the speech information in the unit. Otherwise, $\hat{M}(k,l)$ will be small, thereby contributing to suppress the background noise. As mentioned in Section 2.2.1, four types of mask $M(k,l)$, namely IRM, SMM, ORM, and SMM, are investigated in this work.

2.3.2. Phase reconstruction

Phase reconstruction is performed after obtaining the estimated PDs by the well-trained DNN. Since IF and GD are defined as phase differences between TF units of the spectrogram along the time and frequency axes, respectively, an appropriate initial phase estimate over some chosen TF unit is required to recover the phase spectrogram. Based on the initial estimate, the entire phase spectrogram can be reconstructed along the time and frequency axes through the difference equations in (8) and (10).

(1) Initial phase estimation: When the clean speech power is much larger than the noise power, the noisy phase is approximately equal to the clean phase. Hence, using the noisy phase as an initial estimate is justified in TF units with higher signal-to-noise ratio (SNR). As suggested in Zheng and Zhang (2018), we adopt the noisy phase spectrogram as the initial estimate of the clean phase, that is,

$$\hat{\phi}_{init}(k,l) = \phi_Y(k,l), \forall k,l. \tag{14}$$

We then use the local SNR of each TF unit as an index to determine the initial estimate's reliability, where the local SNR is approximated by the estimated mask $\hat{M}(k,l)$.

(2) Phase reconstruction with GD: At first, the estimated RGD, denoted as $\widehat{\mathrm{RGD}}(k,l)$, should be mapped back to $\mathrm{GD}_n(k,l)$ using the following transformation,

$$\widehat{\mathrm{GD}}_n(k,l) = \frac{1}{2}\left(\mathrm{erf}\left(\frac{\widehat{\mathrm{RGD}}(k,l) - \mu}{\sqrt{2}\sigma}\right) + 1\right) \tag{15}$$

where the $erf(\cdot)$ is the error function. Then the estimated GD is obtained by denormalizing $\widehat{\mathrm{GD}}_n$,

$$\widehat{\mathrm{GD}}(k,l) = 2\pi\left(\widehat{\mathrm{GD}}_n(k,l) - \frac{1}{2}\right) \tag{16}$$

Inspired by the phase reconstruction with IFD in Zheng and Zhang (2018), we compute the phase spectrogram using the initial phase estimate and the GD between the initial estimate and the target phase. For each TF unit, we generate $2N_s + 1$ frame-conditioned phase estimates, given by,

$$\hat{\phi}^i(k,l) = \begin{cases} \hat{\phi}_{init}(k,l+i) + \sum_{n=0}^{i-1} \widehat{\mathrm{GD}}(k,l+n), & i \neq 0 \\ \hat{\phi}_{init}(k,l+i), & i = 0 \end{cases}, \tag{17}$$

where $-N_s \leq i \leq N_s$ is the frame distance between the initialized and the target TF units. In this work, we $N_s = 2$. These phase estimates are then unwrapped, i.e.,

$$\bar{\phi}^i(k,l) = \mathrm{unwrap}(\hat{\phi}^i(k,l)\,|\hat{\phi}^i(k,l-1)) \tag{18}$$

The reconstructed phase of the $(k,l)^{\mathrm{th}}$ unit is finally obtained by smoothing the frame-conditioned estimates $\bar{\phi}^i(k,l)$ with the following weighted average operation,

$$\hat{\phi}(k,l) = \frac{\sum_{i=-N_s}^{N_s}\left(s(i)\hat{M}(k,l+i)\right)\bar{\phi}_i(k,l)}{\sum_{i=-N_s}^{N_s} s(i)\hat{M}(k,l+i)} \tag{19}$$

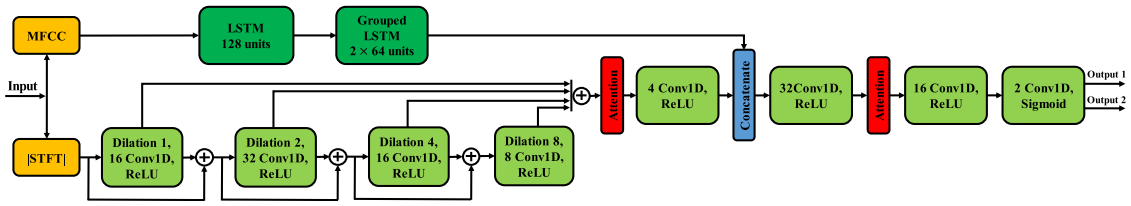

Fig. 6. Composite Model Architecture.

where $s(i)$ denotes the proximity weight for $\bar{\phi}^i(k,l)$, which is inversely related to the absolute value of the frame distance, that is, a phase estimate $\bar{\phi}^i(k,l)$ with a larger distance $|i|$ is assigned a smaller proximity weight $s(i)$, and to lessen its effect on $\hat{\phi}(k,l)$. In this work, following (Zheng and Zhang, 2018), we chose $s(i)$ as the Hamming window. Moreover, the estimated mask $\hat{M}(k,l)$ is used as an measure of the initial estimate's reliability. For instance, a larger value of $\hat{M}(k,l+i)$ indicates that the local SNR of the ith TF unit is higher. In this case, the phase estimate $\bar{\phi}^i(k,l)$ is more reliable and contributes more to the final estimate $\hat{\phi}(k,l)$.

(3) Phase reconstruction with IFD: The procedure of phase reconstruction with IFD is presented in Zheng and Zhang (2018). To begin with, the estimated IFD_n, denoted as $\widehat{\text{IFD}}_n(k,l)$, should be denormalized and converted to $\widehat{\text{IF}}(k,l)$. Then, the phase spectrogram is reconstructed using the noisy phase spectrogram and $\widehat{\text{IF}}(k,l)$, with the help of the spectral mask $\hat{M}(k,l)$. Note that phase reconstruction with IFD is similar to the reconstruction with GD. The only difference is that the former is reconstructed along the time axis, while the latter is reconstructed along the frequency axis.

Besides reconstructing the phase with GD only or with IFD only, we also propose the following combination schemes for reconstruction and investigate their corresponding performance in the next section.

- *Two-step reconstruction:* In this scheme, we first take the noisy phase as the initial estimate and use GD/IFD to get the preliminary reconstructed phase. The later is then treated as the initial estimate, which is employed to obtain the final reconstructed phase with IFD/GD.
- *Average reconstruction:* In this scheme, we separately reconstruct the preliminary phase with IFD and GD, respectively. The final reconstructed phase is obtained by averaging the preliminary ones.

With the combination schemes, the final phase estimate $\hat{\phi}_S(k,l)$ is obtained along both time and frequency axes.

Finally, the estimated clean speech spectrogram can be obtained by combining, the reconstructed magnitude and phase spectra.

2.4. Detailed PACDNN architecture

The composite neural network architecture of our proposed PACDNN model is shown in Fig. 6. The upper stream comprises an LSTM network with two layers, each having 128 LSTM units. We use Mel-frequency cepstral coefficients (MFCCs) as the LSTM network input, since MFCC is an optimal input for the LSTM network in terms of complexity and performance, as demonstrated in Hasannezhad et al. (2020b). More specifically, MFCC features are concatenated with their first and second differences, and then normalized to zero mean and unit variance. As mentioned in Section 2.1.1, the grouping strategy is adopted to reduce the LSTM network complexity where the input and hidden layers are divided into K groups. Empirically, we found that grouping only the second layer with K=2 leads to the best SE results.

In the bottom stream of Fig. 6, the noisy speech STFT magnitude is used as input to the CNN network, which consists of a stack of four dilated-frequency convolutional layers with increasing dilation rates of 1, 2, 4, and 8. The number of kernels in these layers is 16, 32, 16, and 8, with rectified linear unit (ReLU) activation function. Since we want this stream to capture spectral contextual information, the convolutions are 1-dimensional with kernel sizes of 1 along with the time and 7 along the frequency dimension. The feed-forward lines around these layers are residual paths, in the form of convolutional layers with kernel size (1, 1), are used to improve the training procedure. As shown, the outputs of each layer are added up (with a skip connection) to make the output of the CNN network. The output then goes to an attention block, as explained in Section 2.1.2.

The outputs of the LSTM and CNN networks are then concatenated along the channel dimension to form the complementary feature set. Subsequently, another low complexity attention-driven CNN transforms this feature set into the desired targets. This CNN is made up of three convolutional layers with kernel size (1,3) where the number of channels is 32, 16, and 2. The first two layers are followed by ReLU, while the activation function of the output layer is the sigmoid. As explained before, the network estimates the spectral mask and the PDs, respectively, in the two CNN channels. Since the structures of these estimators are similar, they are included as two subtasks for the same network through a parameter sharing mechanism. This mechanism provides better generalization and improves learning because it induces a regularization effect between the two subtasks (Tan and Wang, 2019). In the signal reconstruction block, the information from these two channels is used to resynthesize both magnitude and phase as explained in Section 2.3. Finally, the clean speech samples in the time domain are generated using inverse STFT and overlap-add operation.

3. Experimental evaluation

3.1. Experimental setup

To evaluate the performance of the proposed PACDNN model, the TIMIT database (Garofolo et al., 1993) and IEEE corpus (Rothauser, 1969) are utilized. TIMIT dataset consists of 6300 utterances spoken by 630 male and female speakers, representing eight major dialect divisions of American English, each speaking ten phonetically-rich sentences. The IEEE corpus contains 720 utterances spoken by a single male speaker. For the noise dataset, we use 20 noises (airport, babble, buccaneer1, car, destroyerengine, destroyerops, exhibition, f16, factory, hfchannel, leopard, m109, machinegun, pink, restaurant, street, subway, train, volvo, and white) from NOISEX-92 (Varga and Steeneken, 1993). All the noise files are divided into two parts where random portions of the first part are used for training. The utterances are additively mixed with the noises at SNR levels of −5, 0, 5, and 10 dB in the training stage. In the testing stage, 60 unmatched utterances are randomly selected from each dataset and mixed with random portions of the second part of the noise files at unmatched SNR levels of −6, 0, 6, and 12 dB. Besides, four unseen highly-nonstationary noises, namely, *Coffee Shop, Busy City Street, Car Interior,* and *Street Traffic*, are selected from Premium Beat to evaluate the generalization capability of the proposed model.

The sampling rate is set to 16 kHz, and each mixture is divided into 20 ms time frames with a 10 ms frame shift, i.e., 50% overlap. For each frame, a Hanning window is applied and the 320-point discrete Fourier transform (DFT) is then computed; hence, each frame is represented by 160 STFT coefficients or frequency bins. The STFT are used to extract

Table 1
Comparison of different model targets.

Cases	Objective	PESQ	STOI	SSNR
A	IRM	2.61	0.818	4.33
	ORM	2.61	0.814	4.10
	PSM	2.66	0.824	4.28
	SMM	2.54	0.831	4.26
B	IFD+IRM	2.67	0.847	5.51
	IFD+ORM	2.66	0.837	4.52
	IFD+PSM	2.71	0.853	6.42
	IFD+SMM	2.64	0.860	5.73
C	GD+IRM	2.67	0.848	5.60
	GD+ORM	2.68	0.844	5.64
	GD+PSM	**2.75**	0.853	**6.47**
	GD+SMM	2.66	**0.861**	5.62
D	(GD-IFD)+IRM	2.70	0.849	5.69
	(GD-IFD)+ORM	2.70	0.843	5.73
	(GD-IFD)+PSM	2.74	0.854	6.43
	(GD-IFD)+SMM	2.65	0.860	5.58
E	(IFD-GD)+IRM	2.70	0.850	5.72
	(IFD-GD)+ORM	2.70	0.844	5.74
	(IFD-GD)+PSM	2.74	0.854	6.42
	(IFD-GD)+SMM	2.65	0.860	5.56
F	Avg(IFD&GD)+IRM	2.70	0.849	5.67
	Avg(IFD&GD)+ORM	2.70	0.843	5.72
	Avg(IFD&GD)+PSM	2.74	0.853	6.41
	Avg(IFD&GD)+SMM	2.65	0.860	5.58

26 MFCC, using a suitable mel-scale filter bank. The MFCCs are finally concatenated with their first and second time differences. Hence, the total length of the feature vector used as input to the LSTM network is 78 (i.e., 26 × 3). The MSE is selected as the cost function, while the Adam optimizer is used as an extension to the stochastic gradient descent (Kingma and Ba, 2014) to minimize the error between ideal (ground truth) and estimated values of the desired mask and PD, as follows,

$$\text{MSE} = \frac{1}{LK}\sum_{l}\sum_{k}[(M(k,l)-\hat{M}(k,l))^2 + (PD(k,l)-\widehat{PD}(k,l))^2] \tag{20}$$

where L and K respectively denote the number of time frames and frequency bins.

The speech enhancement performance is evaluated in terms of the following network objective measures: PESQ, short-time objective intelligibility (STOI), and segmental signal-to-noise ratio (SSNR) metrics. PESQ compares the enhanced and clean speech in terms of quality; it generates a score between −0.5 and 4.5, where a higher value corresponding to better quality. STOI measures speech intelligibility by using the correlation between short-time temporal envelopes the clean and enhanced speech; the corresponding range lies between 0 and 1, with a higher value corresponding to better intelligibility. SSNR quantifies amount of residual noise in a in the enhanced speech by computing and averaging the weighted SNR over segments with speech activity. As stated in Hu and Loizou (2007), these three metrics are highly correlated with subjective measurements. The comparisons are accomplished with the same dataset and configuration using a GeForce RTX 2080 graphic card and 2.2 GHz AMD 12-Core Processor.

3.2. Phase-aware method evaluation

The proposed DNN aims to simultaneously estimate the values of both PD and spectral mask. We treat IFD, GD, and their combinations as general PDs. Besides, we investigate four spectral masks, i.e., IRM, ORM, PSM, and SMM.

The comparative performance of the PACDNN model using different combinations of the masks and PDs is shown in Table 1. The experiments are performed using the TIMIT dataset and *restaurant*, *factory*, *street*, and *babble* as noises. The numbers in the table are averaged over all noises and SNR levels. The table is made up of six parts as explained below.

A. This part shows the evaluation metric scores when only a mask is considered as the network's training target with no PD. As seen, PSM yields the best PESQ score while SMM and IRM lead to better STOI and SSNR scores, respectively.

B. This part compares the use of different masks alongside IFD. The results are better than the previous scenario, showing the advantage of enhancing phase alongside magnitude. In this case, IFD+PSM performs better in terms of PESQ and SSNR, while IFD+SMM yields a slightly better STOI score.

C. This part compares the use of different spectral masks alongside GD. The results are better than both previous scenarios illustrating GD's advantage over IFD. GD+PSM outperforms other combinations in this group in terms of PESQ and SSNR, but not STOI.

D. In this part, a two-stage phase reconstruction is investigated where the noisy phase and GD estimation are used to reconstruct the phase in the first stage, and then the reconstructed phase and IFD estimation are used to obtain the final clean phase estimate in the second stage.

E. This part is similar to the previous one, but in a reverses order, i.e.: the noisy phase and IFD are first used to reconstruct the phase, and the reconstructed phase is the used with GD estimation to obtain the final phase estimate.

F. This part shows the results when the average of the reconstructed phase using IFD and GD estimation is considered as the clean phase. Although these combinations give good results, the best PESQ and SSNR are obtained using GD+PSM, and the best STOI with GD+SMM.

Hence, we can conclude that the model using PSM+GD as the training target outperforms other scenarios, and thus we adopt it for the rest of our experiments.

3.3. Advantages of grouped LSTM

In the PACDNN model, the LSTM stream exploits the input speech spectrogram's temporal contextual information. LSTM is the most common RNN variation, which is used in this work to avoid the exploding and vanishing gradient problems (Chen and Wang, 2017). Other RNN variations are also considered, such as, GRU and bidirectional forms called BLSTM and BGRU. Furthermore, we adopt the grouping strategy in the LSTM stream to reduce its complexity, as explained in Section 2.1.1. This section evaluates the PACDNN model performance using the above-mentioned RNN variations with and without the grouping strategy.

In addition to the metrics mentioned in Section 3.1, we compare these variations in terms of: the number of parameters and the required memory to store them; computation time for processing one second of input noisy speech; and memory footprint, measured in terms of the required floating-points operations (FLOPs). These additional measurements are essential for characterizing the implementation complexity of SE algorithms. These measurements are all made during the testing stage since the trained model parameters are to be saved in the device hardware.

Fig. 7 presents the performance results of the PACDNN model using GRU, LSTM, BGRU, BLSTM, and their grouped versions. In this figure, *M* and *MB* denote million and megabyte, respectively. Note that the dataset is the same as in Section 3.2, and the values for PESQ, STOI, and SSNR (dB) show the average improvement over all the noises and SNR levels. As shown in the figure, using grouped-LSTM yields the best STOI and SSNR scores, while LSTM outperforms others in terms of PESQ score. While the objective results do not show a considerable difference, the results for the complexity measures, especially FLOPs and number of parameters, display huge variations. With respect to processing time, GRU is clearly the fastest while BLSTM is the slowest approach. The grouped variations lead to models with smaller number

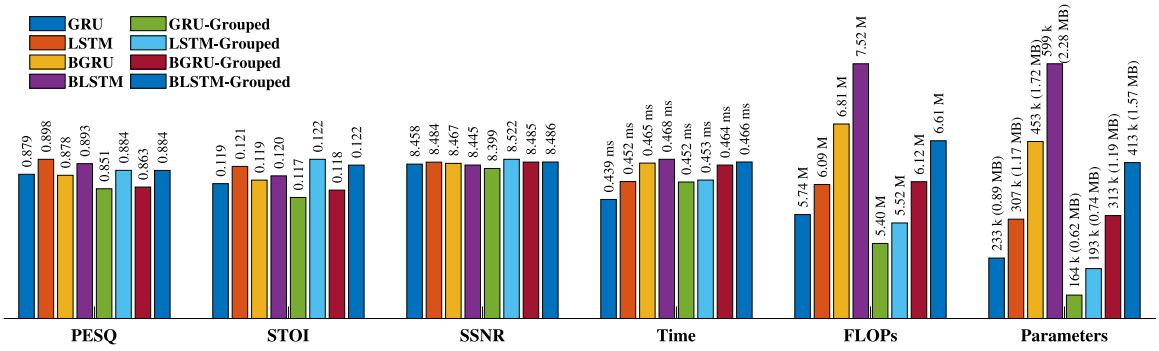

Fig. 7. Comparison of PACDNN performance when using different RNN variations.

of parameters and FLOPs, and among them, grouped-GRU requires the least number of parameters and FLOPs, while grouped-LSTM ranks second. Considering both objective speech quality and computational complexity metrics, the grouped-LSTM offers the best trade-off among the RNN variations in the PACDNN model.

3.4. Benefits of attention-driven CNN

CNN generates many feature maps, each containing some spectrogram characteristics. These feature maps mostly convey noise or speech information. In the PACDNN model, the attention technique is embedded in CNNs to recalibrate feature map weights and emphasize the speech-bearing ones. As mentioned in Section 2.1.2, we consider three attention techniques, i.e., channel-wise, spatial, and parallel, to be embedded in the PACDNN model, and compare the overall model performance. The results of the different cases, using the same dataset as in Section 3.2, are illustrated in Fig. 8, where the values show the average improvement over all the noises and SNR levels. Considering the PESQ score, the PACDNN model with no attention gives the lowest score, while embedding the parallel attention technique yields the highest score. Regarding STOI, the model with the parallel attention again outperforms others, while that with no attention leads to the lowest score. These results demonstrate the effectiveness of attention techniques in emphasizing the informative feature maps. The parallel attention technique made up of average and max pooling can also capture important information of the input feature maps from different perspectives and further improve their representation power. Regarding SSNR, the use of attention model tends to reduce, although marginally, the attainable values.

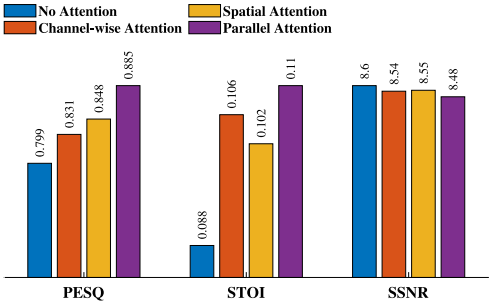

Fig. 8. Comparison of PACDNN performance when embedding different attention methods.

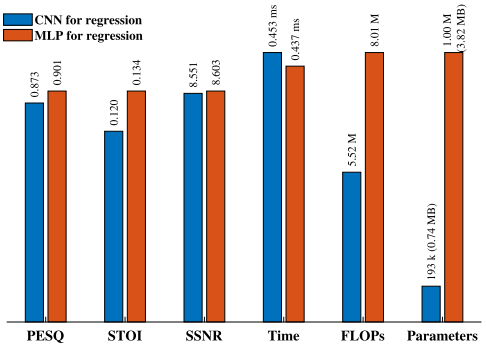

Fig. 9. Comparison of PACDNN model performance while using CNN or MPL for the final regression.

3.5. Investigation of the regression model

This section evaluates the CNN in Section 2.4 against an MLP for the final regression part of the PACDNN model. The MLP contains three layers, each having 512 nodes with a ReLU activation function. A dropout at the rate of 0.3 is also applied to avoid over-fitting. The output layer consists of 322 nodes with sigmoid activation functions to build the desired mask and PD.

The comparative performance of the two networks in terms of objective speech quality and computational complexity metrics is presented in Fig. 9. As shown, MLP yields slightly better results in terms of objective measurements. This marginal advantage of MLP stems from its number of parameters. Using MLP in PACDNN requires around five times more trainable parameters than using CNN, which means the model with MLP can learn more specific patterns of the training dataset. It is worth mentioning that a low complexity model is preferable from the implementation and generalization perspectives. While a model with a low number of parameters does not have the capacity to learn specific patterns or detailed information about noise and speech utterances in the training dataset, and it can perform very well under unseen acoustic conditions. Apart from that, using CNN and MLP in the model respectively requires 0.74 MB and 3.82 MB of memory to store the fixed model parameters, which is proportional to the number of parameters. While the basic computations in MLP are conceptually simpler than in CNN, the former still requires 1.46 times more FLOPs than the former, which is due to the larger number of model parameters. At last, the computation time of CNN, which performs a large number of matrix multiplications, slightly exceeds that of MLP.

3.6. Comparison with other DNN-based methods

This section compares the proposed PACDNN model with some well-known DNN models in the SE task. The selected models have moderate complexity. All the selected methods consider phase information for SE along with magnitude enhancement. All the models, including PACDNN, are trained and tested with the same dataset under the same condition to ensure a fair comparison. The selected methods are summarized below:

Table 2
Comparison of different methods with unseen **male** utterances from TIMIT dataset.

SNR	Method	PESQ				STOI				SSNR			
		bble	ftry	rtrt	strt	bble	ftry	rtrt	strt	bble	ftry	rtrt	strt
−6 dB	Unprocessed	1.23	1.08	1.29	1.25	0.522	0.509	0.516	0.609	−10.6	−10.3	−9.97	−9.64
	IRM-MIFD-MLP	1.57	1.57	1.68	1.95	0.588	**0.591**	0.647	0.724	−3.83	−2.52	−1.78	−0.30
	cIRM-MLP	1.57	1.74	1.68	2.06	0.562	0.568	0.624	0.712	−1.02	**0.29**	0.14	1.65
	MCIRM-CNNGRU	1.57	1.71	1.53	1.90	0.523	0.544	0.545	0.657	−2.42	−0.80	−2.01	0.45
	cIRM-CNNLSTM	1.67	1.80	1.84	2.04	0.578	0.589	**0.667**	0.716	−1.29	−0.80	−0.26	1.53
	CS-CNN	1.53	1.41	1.48	1.72	0.515	0.518	0.506	0.612	−4.25	−3.86	−5.22	−1.95
	DCTCRN	1.68	1.70	1.79	**2.32**	0.586	**0.591**	0.687	0.721	−0.22	−0.71	0.08	1.22
	TCNN	1.59	1.64	1.69	2.02	0.586	0.584	0.638	0.722	**−0.17**	−0.35	0.00	1.05
	Proposed	**1.72**	**1.81**	**1.87**	2.19	**0.591**	**0.591**	0.660	**0.740**	−1.10	−0.46	**0.17**	**1.66**
0 dB	Unprocessed	1.66	1.52	1.65	1.74	0.659	0.647	0.651	0.718	−5.97	−5.72	−5.45	−4.84
	IRM-MIFD-MLP	2.12	2.15	2.24	2.49	0.732	0.738	0.762	0.803	0.97	1.96	2.22	3.98
	cIRM-MLP	2.17	2.25	2.24	2.53	0.724	0.716	0.744	0.796	2.21	2.84	2.76	4.18
	MCIRM-CNNGRU	2.10	2.17	2.03	2.42	0.696	0.700	0.698	0.767	1.36	1.65	1.44	3.22
	cIRM-CNNLSTM	2.24	2.31	2.32	2.53	0.736	0.743	0.773	0.813	1.84	2.12	2.52	3.91
	CS-CNN	2.01	1.87	1.88	2.12	0.667	0.662	0.652	0.722	0.17	0.28	−1.23	1.35
	DCTCRN	2.22	2.30	2.28	**2.75**	0.745	0.743	0.769	0.821	1.13	2.10	2.56	4.16
	TCNN	2.16	2.14	2.22	2.49	0.741	**0.762**	0.763	**0.850**	**3.07**	2.18	2.17	**5.17**
	Proposed	**2.28**	**2.34**	**2.34**	2.67	**0.751**	0.750	**0.778**	0.824	2.48	**3.04**	**3.23**	4.60
6 dB	Unprocessed	2.12	1.98	2.08	2.23	0.778	0.780	0.789	0.806	−0.17	−0.21	0.21	0.29
	IRM-MIFD-MLP	2.71	2.70	2.73	2.97	0.837	0.840	0.852	0.862	5.46	5.83	5.73	6.79
	cIRM-MLP	2.74	2.73	2.74	2.94	0.828	0.824	0.845	0.854	5.49	5.35	5.59	6.49
	MCIRM-CNNGRU	2.60	2.63	2.57	2.82	0.811	0.810	0.819	0.836	4.13	4.13	4.33	5.40
	cIRM-CNNLSTM	2.79	2.86	2.84	**3.13**	0.850	0.850	0.859	0.872	5.33	5.38	6.32	5.44
	CS-CNN	2.41	2.24	2.33	2.51	0.780	0.770	0.776	0.800	3.48	3.48	2.87	4.06
	DCTCRN	**2.90**	2.85	2.80	3.01	0.855	0.845	0.861	0.877	6.26	5.58	5.59	6.89
	TCNN	2.66	2.55	2.68	2.93	0.833	0.848	0.867	0.822	**6.68**	5.21	5.71	7.20
	Proposed	2.83	**2.89**	**2.85**	3.08	**0.860**	**0.855**	**0.875**	**0.882**	6.26	**6.66**	**6.51**	**7.72**
12 dB	Unprocessed	2.53	2.46	2.51	2.66	0.877	0.890	0.896	0.886	5.51	5.46	5.88	6.05
	IRM-MIFD-MLP	3.22	3.16	3.20	3.37	0.906	0.907	0.918	0.913	8.42	8.56	8.28	9.09
	cIRM-MLP	3.16	3.17	3.19	3.33	0.897	0.896	0.909	0.903	7.86	7.80	7.79	8.49
	MCIRM-CNNGRU	3.02	3.08	3.04	3.25	0.888	0.882	0.896	0.891	6.73	6.31	6.72	7.51
	cIRM-CNNLSTM	3.27	3.25	3.26	**3.49**	0.910	0.910	0.920	0.916	8.13	8.19	8.21	8.78
	CS-CNN	2.77	2.63	2.74	2.86	0.854	0.848	0.856	0.860	6.01	5.96	5.83	6.42
	DCTCRN	3.26	3.16	3.09	3.43	0.912	0.922	0.924	0.923	9.93	9.27	9.05	10.13
	TCNN	3.10	2.94	3.11	3.24	0.914	**0.932**	0.926	**0.950**	**10.74**	9.23	9.65	**11.54**
	Proposed	**3.30**	**3.26**	**3.27**	3.47	**0.920**	0.921	**0.930**	0.927	10.08	**10.33**	**10.18**	11.11

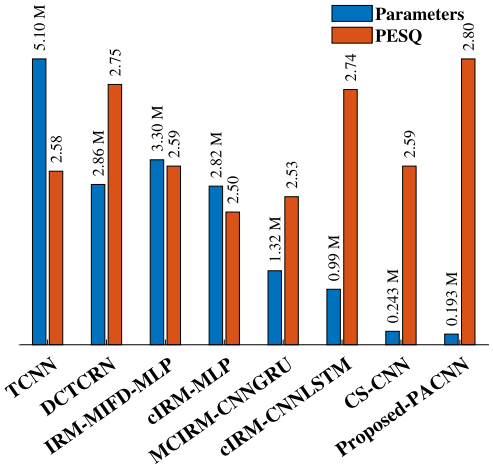

Fig. 10. Comparison of the number of trainable parameters and average PESQ score of different methods.

1. IRM-MIFD-MLP (Zheng and Zhang, 2018): A MLP with three layers is employed in this multi-objective DNN method. Each hidden layer contains 1024 nodes with ReLU activation function while the output layer contains 512 nodes with sigmoid activation function. IRM and IFD are used as training targets.
2. cIRM-MLP (Williamson et al., 2015): In this method, three-layer MLP is employed to approximate cIRM. Each layer has 1024 nodes with ReLU activation function. The output layer with linear activation function estimates the real and imaginary parts of cIRM. The input to the network is a complementary set of acoustic features. To incorporate temporal information, the features from 5 frames are concatenated and fed to the network at once.
3. MCIRM-CNNGRU (Hasannezhad et al., 2020a): In this method, a hybrid model is used to estimate the real and imaginary parts of a modified cIRM. The network is made up of a CNN for feature extraction and a GRU network for regression. The complex spectrogram is used as the input and a 322-node output layer with linear activation function generates the desired mask values.
4. cIRM-CNNLSTM (Hasannezhad et al., 2020b): Here, a CNN, LSTM, and MLP are integrated to estimate cIRM. The feature extraction is performed by the CNN and LSTM networks while the regression is accomplished by the MLP, which maps the features into the real and imaginary components of cIRM.
5. CS-CNN (Ouyang et al., 2019): A fully-convolutional CNN is utilized to estimate the real and imaginary parts of the clean speech complex spectrogram. The input consists of 13 frames of the noisy speech complex spectrogram presented to the network. The middle frame of the output (frame 7) is considered as the enhanced output frame.
6. DCTCRN (Li et al., 2021): Unlike the previous methods that all perform in the frequency domain, this method accomplishes speech enhancement in the discrete cosine transform (DCT) domain so that the magnitude and phase are simultaneously enhanced. The input is the short-time DCT (STDCT) and the training target is a ratio mask including implicit phase information. The CRN is used as the learning machine to perform the mapping between the input STDCT and the ratio mask.

Table 3
Comparison of different methods with unseen **female** utterances from TIMIT dataset.

SNR	Method	PESQ				STOI				SSNR			
		bble	ftry	rtrt	strt	bble	ftry	rtrt	strt	bble	ftry	rtrt	strt
−6 dB	Unprocessed	0.95	0.87	0.93	0.93	0.512	0.504	0.497	0.588	−9.97	−10.0	−9.51	−9.51
	IRM-MIFD-MLP	1.22	1.23	1.28	1.54	0.568	0.579	0.631	0.693	−3.51	−2.26	−1.19	−0.13
	cIRM-MLP	1.27	1.43	1.31	1.67	0.545	0.549	0.601	0.679	−0.92	**0.40**	0.32	1.97
	MCIRM-CNNGRU	1.28	1.37	1.23	1.60	0.506	0.526	0.544	0.635	−2.15	−0.95	−1.90	0.56
	cIRM-CNNLSTM	1.38	1.45	1.44	1.79	0.551	0.564	0.635	0.690	−0.93	−0.58	0.48	1.90
	CS-CNN	1.27	1.23	1.23	1.45	0.505	0.517	0.497	0.588	−3.85	−3.04	−5.03	−1.58
	DCTCRN	1.36	1.36	1.31	**1.95**	0.561	0.580	**0.656**	0.685	−0.88	−1.18	0.41	1.39
	TCNN	1.34	1.41	**1.46**	1.78	0.564	**0.599**	0.623	0.670	−1.46	−0.30	0.79	1.04
	Proposed	**1.40**	**1.48**	**1.46**	1.80	**0.570**	0.561	0.629	**0.703**	**−0.70**	−0.12	**0.94**	**2.35**
0 dB	Unprocessed	1.36	1.28	1.36	1.48	0.640	0.635	0.641	0.694	−5.58	−5.64	−5.00	−4.89
	IRM-MIFD-MLP	1.77	1.81	1.82	2.09	0.709	0.718	0.743	0.784	1.34	2.13	2.50	3.97
	cIRM-MLP	1.81	1.94	1.88	2.15	0.687	0.690	0.729	0.765	2.46	2.90	3.09	4.40
	MCIRM-CNNGRU	1.83	1.92	1.75	2.10	0.674	0.676	0.699	0.754	1.44	1.83	1.63	3.61
	cIRM-CNNLSTM	1.92	1.98	1.93	2.41	0.710	0.716	0.749	0.787	2.12	2.32	3.15	4.33
	CS-CNN	1.76	1.65	1.67	1.91	0.653	0.649	0.643	0.702	0.46	0.59	−0.73	1.65
	DCTCRN	**2.03**	1.98	1.87	**2.58**	0.719	0.748	0.735	0.752	2.05	2.91	3.34	4.13
	TCNN	1.90	1.95	1.85	2.30	**0.767**	**0.765**	0.748	0.786	2.52	2.93	3.61	**5.75**
	Proposed	1.94	**2.06**	**1.96**	2.30	0.721	0.711	**0.750**	**0.798**	**3.00**	**3.50**	**3.80**	5.35
6 dB	Unprocessed	1.85	1.77	1.84	2.01	0.766	0.763	0.777	0.798	−0.18	−0.30	0.16	0.49
	IRM-MIFD-MLP	2.34	2.37	2.35	2.63	0.822	0.824	0.838	0.849	5.61	6.00	6.06	7.31
	cIRM-MLP	2.34	2.42	2.36	2.63	0.805	0.802	0.823	0.836	5.31	5.41	5.68	6.94
	MCIRM-CNNGRU	2.31	2.37	2.25	2.54	0.795	0.788	0.810	0.824	4.34	4.23	4.59	5.75
	cIRM-CNNLSTM	2.47	2.52	**2.52**	2.85	0.824	0.830	0.845	0.857	5.67	5.59	5.76	7.08
	CS-CNN	2.17	2.11	2.09	2.31	0.768	0.763	0.770	0.790	3.70	3.84	3.13	4.41
	DCTCRN	2.32	2.46	2.48	**2.93**	0.792	0.802	0.805	0.829	6.23	7.29	7.00	8.19
	TCNN	2.41	2.38	2.47	2.75	**0.882**	0.794	0.815	0.830	6.27	**7.41**	7.00	8.47
	Proposed	**2.50**	**2.58**	2.47	2.78	0.836	**0.837**	**0.849**	**0.869**	**6.82**	7.09	**7.06**	**8.75**
12 dB	Unprocessed	2.34	2.30	2.31	2.50	0.869	0.883	0.888	0.882	5.53	5.50	5.91	6.11
	IRM-MIFD-MLP	2.90	2.92	2.85	3.09	0.896	0.900	0.905	0.899	8.61	8.67	8.42	9.24
	cIRM-MLP	2.86	2.96	2.84	3.05	0.882	0.887	0.892	0.891	7.67	7.65	7.71	8.62
	MCIRM-CNNGRU	2.82	2.85	2.76	2.99	0.875	0.873	0.887	0.881	7.23	6.69	7.15	7.74
	cIRM-CNNLSTM	3.00	2.99	2.95	3.25	0.898	0.903	0.909	0.907	8.25	8.4	8.52	9.46
	CS-CNN	2.56	2.47	2.52	2.70	0.843	0.840	0.847	0.853	6.16	6.07	5.95	6.52
	DCTCRN	2.90	3.00	2.85	**3.37**	0.877	0.861	0.901	0.894	10.68	10.81	10.80	11.85
	TCNN	2.93	2.79	2.91	3.12	0.897	0.882	0.917	0.855	10.32	9.74	10.16	11.31
	Proposed	**3.02**	**3.07**	**2.97**	3.23	**0.915**	**0.916**	**0.922**	**0.920**	**10.89**	**11.04**	**10.89**	**12.16**

Table 4
Comparison of different methods with unseen utterances from IEEE corpus and 20 different noises.

Method	PESQ				STOI				SSNR			
	−6	0	6	12	−6	0	6	12	−6	0	6	12
Unprocessed	1.40	1.76	2.13	2.54	0.588	0.708	0.825	0.913	−8.99	−5.17	0.01	5.75
IRM-MIFD-MLP	1.83	2.36	2.88	3.30	0.711	0.824	0.898	0.942	−1.77	3.19	7.30	10.16
cIRM-MLP	1.85	2.37	2.86	3.27	0.690	0.810	0.889	0.938	1.03	4.35	7.26	10.02
MCIRM-CNNGRU	1.85	2.34	2.78	3.15	0.658	0.782	0.869	0.922	0.22	3.45	6.14	8.26
cIRM-CNNLSTM	2.06	2.58	3.06	3.44	0.720	0.832	0.907	0.949	1.02	4.54	8.07	10.88
CS-CNN	1.98	2.46	2.82	3.09	0.685	0.817	0.896	0.939	**2.11**	4.98	9.23	11.01
DCTCRN	1.91	2.39	2.83	3.21	0.704	**0.840**	0.891	0.937	1.42	4.96	**9.43**	11.47
TCNN	1.85	2.30	2.64	2.93	0.690	0.811	0.877	0.912	1.52	5.06	8.05	10.36
Proposed	**2.07**	**2.60**	**3.08**	**3.46**	**0.724**	0.838	**0.911**	**0.955**	1.63	**5.08**	8.34	**11.76**

7. TCNN (Pandey and Wang, 2019): This method is designed to perform in the time domain where a temporal convolutional neural network (TCNN), along with an embedded encoder–decoder architecture with a temporal convolutional network, is employed to directly map the noisy speech to the clean one.

Fig. 10 illustrates the number of trainable parameters of each method along with the average PESQ score of the processed speech over different noises and SNR levels evaluated with the TIMIT dataset. As shown, TCNN and DCTCRN are of high numbers of model parameters; thus, they have high computational complexity. As expected, the MLP-based models, i.e., IRM-MIFD-MLP and cIRM-MLP, also contain high numbers of model parameters and, consequently, require a large memory to store them. It is worth mentioning that the computations of TCNN and DCTCRN are much higher than MLP-based models since the formers contain many convolutional operations. Two other hybrid models, i.e., MCIRM-CNNGRU and cIRM-CNNLSTM, have a fair number of parameters, each around 1 million. The lowest number of parameters belongs to CS-CNN and the proposed model, with the latter requiring slightly fewer parameters. Although the number of model parameters of PACDNN is only 3% of TCNN and 6% of DCTCRN, it outperforms all these aforementioned models in the SE task, as shown in the figure and further discussed below.

Since speech characteristics differ between males and females, we evaluate different models separately to show the generalization capability of the desired models to different genders. The comparison results for male test utterances from the TIMIT dataset are shown in Table 2 where bble, ftry, rtrt, and strt denote babble, factory, restaurant, and street noises, respectively. As shown, the proposed model outperforms all the other ones in terms of the various objective quality metrics, except for a few cases, including PESQ at SNR levels of −6 and 0 dB for street noise where DCTCRN give slightly better scores and SNR levels of 6 and 12 dB where cIRM-CNNLSTM achieves slightly better scores. Also, at the SNR level of 0 and 12 dB, TCNN yields marginally better

Table 5
Comparison of different methods with unseen utterances form IEEE corpus mixed with **unseen** noises at unmatched SNR levels.

SNR	Method	PESQ				STOI				SSNR			
		bscs	cair	cfsp	sttc	bscs	cair	cfsp	sttc	bscs	cair	cfsp	sttc
−6 dB	Unprocessed	1.30	1.12	1.71	1.18	0.587	0.506	0.715	0.497	−7.06	−8.66	−9.16	−9.26
	IRM-MIFD-MLP	1.76	1.37	2.20	1.42	0.693	0.579	0.817	0.584	−2.01	−3.35	−1.28	−3.50
	cIRM-MLP	1.73	1.25	2.23	1.34	0.678	0.557	0.814	0.544	0.58	−0.97	2.14	−1.00
	MCIRM-CNNGRU	1.69	1.42	2.09	1.37	0.632	0.539	0.756	0.518	−0.71	−2.21	1.03	−2.85
	cIRM-CNNLSTM	1.74	1.30	2.37	1.30	0.677	0.562	0.815	0.542	0.23	−1.18	2.73	−1.77
	CS-CNN	1.64	1.46	2.14	1.53	0.657	0.554	0.786	0.535	0.99	−0.95	3.12	−2.08
	DCTCRN	1.62	1.27	2.16	1.13	0.651	0.543	0.799	0.491	0.24	−1.09	3.05	−1.62
	TCNN	1.59	1.46	2.15	1.47	0.661	0.586	0.748	**0.608**	**1.00**	−0.89	2.91	**−0.93**
	Proposed	**1.96**	**1.51**	**2.59**	**1.55**	**0.701**	**0.601**	**0.845**	0.578	0.90	**−0.65**	**3.86**	−1.08
0 dB	Unprocessed	1.81	1.60	2.17	1.52	0.727	0.632	0.807	0.629	−4.33	−4.97	−4.96	−5.38
	IRM-MIFD-MLP	2.27	1.96	2.70	1.93	0.815	0.742	0.876	0.728	2.68	1.35	4.05	1.36
	cIRM-MLP	2.28	1.91	2.80	1.91	0.803	0.736	0.873	0.722	3.85	2.77	5.23	2.81
	MCIRM-CNNGRU	2.18	1.89	2.58	1.84	0.777	0.686	0.852	0.682	2.55	1.36	3.90	1.36
	cIRM-CNNLSTM	2.34	1.99	2.84	1.97	0.815	0.733	0.882	0.728	4.04	2.38	5.68	2.76
	CS-CNN	2.29	1.96	2.63	2.05	0.827	0.731	0.882	0.725	**5.66**	3.35	6.09	2.65
	DCTCRN	2.18	1.89	2.61	1.77	0.805	0.719	0.877	0.708	4.17	2.69	6.14	2.87
	TCNN	2.15	2.09	2.60	2.06	0.815	0.723	0.853	0.710	5.04	**3.78**	6.16	3.00
	Proposed	**2.49**	**2.10**	**3.02**	**2.14**	**0.837**	**0.758**	**0.897**	**0.753**	4.62	3.41	**6.33**	**3.05**
6 dB	Unprocessed	2.15	2.00	2.58	1.87	0.837	0.774	0.878	0.761	0.63	0.17	0.29	−0.18
	IRM-MIFD-MLP	2.76	2.54	3.17	2.51	0.897	0.853	0.916	0.849	6.85	5.99	7.75	6.16
	cIRM-MLP	2.76	2.52	3.24	2.50	0.885	0.853	0.915	0.848	6.89	6.08	7.75	6.04
	MCIRM-CNNGRU	2.64	2.42	3.06	2.37	0.872	0.822	0.904	0.819	5.74	4.71	6.68	4.84
	cIRM-CNNLSTM	2.86	2.61	3.30	2.55	0.903	0.858	0.928	0.854	7.13	6.16	8.90	6.68
	CS-CNN	2.66	2.48	3.05	2.49	0.902	0.860	**0.937**	0.848	**9.17**	6.91	**10.1**	6.54
	DCTCRN	2.67	2.42	3.04	2.36	0.901	0.854	0.930	0.846	7.99	6.89	8.96	6.82
	TCNN	2.56	2.53	2.95	2.43	0.884	0.868	0.902	0.852	8.08	**7.32**	8.82	6.65
	Proposed	**2.92**	**2.67**	**3.41**	**2.67**	**0.906**	**0.873**	0.935	**0.866**	7.86	7.12	8.92	**6.97**
12 dB	Unprocessed	2.54	2.44	2.98	2.30	0.927	0.889	0.933	0.883	6.56	6.10	6.15	5.49
	IRM-MIFD-MLP	3.29	3.04	3.59	2.99	0.947	0.922	0.950	0.922	10.07	9.37	10.34	9.62
	cIRM-MLP	3.21	3.02	3.63	3.01	0.940	0.920	0.949	0.922	10.00	9.27	10.21	9.25
	MCIRM-CNNGRU	3.05	2.89	3.48	2.87	0.931	0.905	0.939	0.902	8.24	7.80	9.09	7.34
	cIRM-CNNLSTM	3.26	3.10	3.63	3.07	0.950	0.932	0.954	0.923	10.22	9.12	10.15	9.91
	CS-CNN	2.98	2.87	3.38	2.87	0.948	0.930	**0.964**	0.924	**12.04**	10.17	**12.58**	10.64
	DCTCRN	3.02	2.96	3.42	2.90	0.943	0.934	0.960	0.930	11.48	10.19	11.80	10.83
	TCNN	2.81	2.88	3.20	2.85	0.914	0.911	0.925	0.903	10.39	10.07	10.74	9.59
	Proposed	**3.36**	**3.19**	**3.70**	**3.17**	**0.955**	**0.936**	0.962	**0.936**	11.45	**10.79**	11.80	**10.93**

Table 6
Cross-corpus evaluation, where the training and testing are accomplished with TIMIT dataset and IEEE corpus, respectively.

Method	PESQ				STOI				SSNR			
	−6	0	6	12	−6	0	6	12	−6	0	6	12
Unprocessed	1.29	1.70	2.10	2.52	0.541	0.676	0.814	0.913	−8.27	−4.64	0.61	6.34
IRM-MIFD-MLP	1.57	2.05	2.53	3.01	**0.609**	0.741	0.837	0.922	−2.44	1.63	4.76	6.51
cIRM-MLP	1.53	2.03	2.51	2.97	0.591	0.731	0.840	0.907	**−0.17**	2.21	4.22	5.77
MCIRM-CNNGRU	1.55	2.00	2.44	2.87	0.531	0.703	0.822	0.891	−1.69	1.50	3.45	4.57
cIRM-CNNLSTM	**1.66**	2.09	2.58	3.01	0.598	0.740	0.843	0.907	−0.72	2.57	4.35	5.96
CS-CNN	1.48	1.84	2.24	2.57	0.513	0.658	0.764	0.827	−3.99	−0.32	2.84	4.88
DCTCRN	**1.66**	1.99	2.46	3.00	0.589	0.745	0.850	0.920	−0.93	2.04	5.16	7.00
TCNN	1.47	1.97	2.43	2.82	0.592	0.751	0.857	0.909	−0.06	2.24	5.18	6.97
Proposed	1.64	**2.12**	**2.60**	**3.05**	0.604	**0.752**	**0.858**	**0.926**	−0.30	**2.62**	**5.37**	**7.96**

STOI for street and factory noises. Furthermore, TCNN gives better SSNR scores for babble and street noises at SNR levels of 0 and 12 dB. Table 3 illustrates results for the female utterances from the TIMIT dataset. Again, we can see that the proposed model outperforms the others in nearly all cases, except for a few cases of STOI at SNR level of −6 and 0 where DCTCRN and TCNN gives better results.

In another experiment, we compare the different methods on the IEEE corpus where 20 noises are mixed with the selected utterances, with unmatched SNR levels between the training and testing stages. As can be seen from Table 4, which presents the average scores for the PESQ, STOI and SSNR metrics, the proposed model clearly outperforms all the other methods in all cases, except for the SSNR scores at SNR levels of −6 and 6 dB, where CS-CNN and DCTCRN give slightly better results. This experiment demonstrates that although the proposed model has a very small number of parameters, it can perform well under different noise conditions.

Under the same training conditions as for Table 4, we tested the different methods with unseen highly-nonstationary noises mixed with unseen utterances from IEEE corpus at unmatched SNR levels to evaluate their generalization capability in unseen conditions. The comparison results are shown in Table 5 where bscs, cair, cfsp, and sttc denote *Coffee Shop, Busy City Street, Car Interior,* and *Street Traffic*. It can be seen that the proposed model generally outperforms all the other methods, except for a few cases. This experiment demonstrates that the proposed model has very good generalization capability thanks to its careful design and the small number of parameters making it not learn specific patterns of the training dataset but instead rely on the general information of speech and noise.

As shown in Pandey and Wang (2020), there can be a considerable performance degradation with DNN methods when the training and testing datasets are different, especially at low SNR levels. This study reveals that some well-known but highly complex SE methods do not

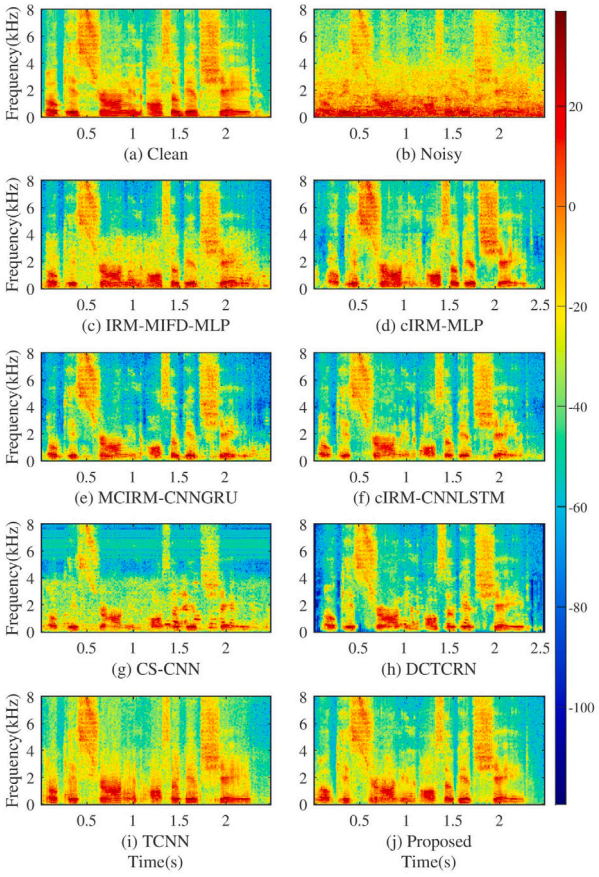

Fig. 11. Illustration of the STFT magnitudes (log scale) of a enhanced speech using different models. The models are trained with TIMIT dataset and tested with an utterance from IEEE corpus. The utterance is mixed with the *street* noise at the SNR level of 0 dB.

perform well on untrained corpora. In this last experiment, we compare the cross-corpus generalization capability of different methods. To this end, we trained different models with the TIMIT dataset and tested them with the IEEE corpus. The results, shown in Table 6 for different SNR levels, reveal that the proposed model outperforms the other ones when the training and testing datasets are different, except at SNR −6 dB, where other methods yield somehow better results. Furthermore, a sample spectrograms is illustrated in Fig. 11 showing the differences of different methods. Hence, we can conclude that the proposed PACDNN model offers very good generalization capability to unseen datasets.

4. Conclusion

This paper proposed a phase-aware composite deep neural network called PACDNN for speech enhancement where both speech magnitude and phase are enhanced. Specifically, we designed a masking-based method to enhance the magnitude and employed phase derivative to reconstruct the clean speech phase. Due to the structural similarity of the spectral mask and phase derivative, a single neural network was used to estimate both information types through simultaneous parameter sharing. The proposed network integrates improved LSTM and CNN, which perform in parallel to exploit a complementary set of features. Different potential DNN solutions were investigated and compared in terms of objective speech quality and computational complexity measures in order to optimize the final regression between the features and the desired targets. Through extensive series of experiments, the resulting PACDNN model was evaluated and compared with several known DNN-based SE methods using different datasets and objective measures. In particular, the capability of the proposed model in dealing with unseen noisy conditions, cross-corpus generalization, and unmatched SNR levels in testing and training were investigated, demonstrating the advantages of PACDNN over other methods in SE applications, in spite of its lower complexity.

CRediT authorship contribution statement

Mojtaba Hasannezhad: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft. **Hongjiang Yu:** Methodology, Software, Formal analysis, Writing – original draft. **Wei-Ping Zhu:** Validation, Formal analysis, Writing – review & editing, Supervision, Resources. **Benoit Champagne:** Validation, Formal analysis, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under a CRD grant from NSERC (Govt. of Canada) with industrial sponsor Microchip (Ottawa, Canada).

References

Abbaszadeh, P., 2016. Improving hydrological process modeling using optimized threshold-based wavelet de-noising technique. Water Resour. Manag. 30 (5), 1701–1721.

Abd El-Fattah, M., Dessouky, M.I., Diab, S.M., Abd El-Samie, F.E.-S., 2008. Speech enhancement using an adaptive wiener filtering approach. Prog. Electromagn. Res. 4, 167–184.

Agnew, J., Thornton, J.M., 2000. Just noticeable and objectionable group delays in digital hearing aids. J. Am. Acad. Audiol. 11 (6), 330–336.

Chen, J., Wang, D., 2017. Long short-term memory for speaker generalization in supervised speech separation. J. Acoust. Soc. Am. 141 (6), 4705–4714.

Cui, X., Chen, Z., Yin, F., 2020. Speech enhancement based on simple recurrent unit network. Appl. Acoust. 157, 107019.

Dey, R., Salemt, F.M., 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In: Int. Midwest Symposium on Circuits and Systems. MWSCAS, IEEE, pp. 1597–1600.

Erdogan, H., Hershey, J.R., Watanabe, S., Le Roux, J., 2015. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing. ICASSP, pp. 708–712.

Fu, S.-W., Hu, T.-y., Tsao, Y., Lu, X., 2017a. Complex spectrogram enhancement by convolutional neural network with multi-metrics learning. In: Int. Workshop on Machine Learning for Signal Processing. MLSP, IEEE, pp. 1–6.

Fu, S.-W., Tsao, Y., Lu, X., Kawai, H., 2017. Raw waveform-based speech enhancement by fully convolutional networks. In: IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. APSIPA ASC, pp. 006–012.

Gao, F., Wu, L., Zhao, L., Qin, T., Cheng, X., Liu, T.-Y., 2018. Efficient sequence learning with group recurrent networks. In: Proc. of Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1. pp. 799–808, Long Papers.

Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., 1993. DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon Technical Report, vol. 93.

Hasannezhad, M., Ouyang, Z., Zhu, W.-P., Champagne, B., 2020a. An integrated CNN-gru framework for complex ratio mask estimation in speech enhancement. In: IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. APSIPA ASC, pp. 764–768.

Hasannezhad, M., Ouyang, Z., Zhu, W.-P., Champagne, B., 2020b. Speech separation using a composite model for complex mask estimation. In: Int. Midwest Symposium on Circuits and Systems. MWSCAS, IEEE, pp. 578–581.

Hasannezhad, M., Zhu, W.-P., Champagne, B., 2021. A novel low-complexity attention-driven composite model for speech enhancement. In: International Symposium on Circuits and Systems. ISCAS, IEEE, pp. 1–5.

Hegde, R.M., Murthy, H.A., Gadde, V.R.R., 2007. Significance of the modified group delay feature in speech recognition. IEEE Trans. Audio, Speech, Lang. Process. 15 (1), 190–202.

Hsieh, T.-A., Wang, H.-M., Lu, X., Tsao, Y., 2020. WaveCRN: An efficient convolutional recurrent neural network for end-to-end speech enhancement. arXiv preprint arXiv:2004.04098.

Hu, Y., Liu, Y., Lv, S., Xing, M., Zhang, S., Fu, Y., Wu, J., Zhang, B., Xie, L., 2020. DCCRN: deep complex convolution recurrent network for phase-aware speech enhancement. arXiv preprint arXiv:2008.00264.

Hu, Y., Loizou, P.C., 2007. Evaluation of objective quality measures for speech enhancement. IEEE/ACM Trans. Audio, Speech, Lang. Process. 16 (1), 229–238.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 7132–7141.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krawczyk, M., Gerkmann, T., 2014. STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement. IEEE/ACM Trans. Audio, Speech, Lang. Process. 22 (12), 1931–1940.

Li, Q., Gao, F., Guan, H., Ma, K., 2021. Real-time monaural speech enhancement with short-time discrete cosine transform. arXiv preprint arXiv:2102.04629.

Liang, S., Liu, W., Jiang, W., Xue, W., 2013. The optimal ratio time-frequency mask for speech separation in terms of the signal-to-noise ratio. J. Acoust. Soc. Am. 134 (5), EL452–EL458.

Martin, R., May 2002. Speech enhancement using MMSE short time spectral estimation with gamma distributed speech priors. In: IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Vol. 1. pp. I–253.

Mowlaee, P., Saeidi, R., 2014. Time-frequency constraints for phase estimation in single-channel speech enhancement. In: Int. Workshop on Acoustic Signal Enhancement. IWAENC, IEEE, pp. 337–341.

Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. Wavenet: a generative model for raw audio. arXiv preprint arXiv:1609.03499.

Ouyang, Z., Yu, H., Zhu, W.-P., Champagne, B., 2019. A fully convolutional neural network for complex spectrogram processing in speech enhancement. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing. ICASSP, pp. 5756–5760.

Pandey, A., Wang, D., 2019. TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6875–6879.

Pandey, A., Wang, D., 2020. Learning complex spectral mapping for speech enhancement with improved cross-corpus generalization. Proc. Interspeech 2020 4511–4515.

Parchami, M., Zhu, W.-P., Champagne, B., Plourde, E., 2016. Recent developments in speech enhancement in the short-time Fourier transform domain. IEEE Circuits Syst. Mag. 16 (3), 45–77.

Park, S.R., Lee, J., 2016. A fully convolutional neural network for speech enhancement. arXiv preprint arXiv:1609.07132.

Prasad, V.K., Nagarajan, T., Murthy, H.A., 2004. Automatic segmentation of continuous speech using minimum phase group delay functions. Speech Commun. 42 (3–4), 429–446.

Premium Beat, www.premiumbeat.com.

Rothauser, E., 1969. IEEE Recommended practice for speech quality measurements. IEEE Trans. Audio Electroacoust. 17, 225–246.

Roy, A.G., Navab, N., Wachinger, C., 2018. Concurrent Spatial and Channel 'Squeeze & Excitation' in Fully Convolutional Networks. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, pp. 421–429.

Shifas, M.P., Claudio, S., Stylianou, Y., et al., 2020. A fully recurrent feature extraction for single channel speech enhancement. arXiv preprint arXiv:2006.05233.

Srinivasan, S., Roman, N., Wang, D., 2006. Binary and ratio time-frequency masks for robust speech recognition. Speech Commun. 48 (11), 1486–1501.

Stark, A.P., Paliwal, K.K., 2008. Speech analysis using instantaneous frequency deviation. In: INTERSPEECH.

Strake, M., Defraene, B., Fluyt, K., Tirry, W., Fingscheidt, T., 2020. Fully convolutional recurrent networks for speech enhancement. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing. ICASSP, pp. 6674–6678.

Takamichi, S., Saito, Y., Takamune, N., Kitamura, D., Saruwatari, H., 2018. Phase reconstruction from amplitude spectrograms based on von-Mises-distribution deep neural network. In: 2018 16th International Workshop on Acoustic Signal Enhancement. IWAENC, IEEE, pp. 286–290.

Takamichi, S., Saito, Y., Takamune, N., Kitamura, D., Saruwatari, H., 2020. Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks. Signal Process. 169, 107368.

Tan, K., Wang, D., 2018. A convolutional recurrent neural network for real-time speech enhancement.. In: INTERSPEECH. pp. 3229–3233.

Tan, K., Wang, D., 2019. Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement. IEEE/ACM Trans. Audio, Speech, Lang. Process. 28, 380–390.

Varga, A., Steeneken, H.J., 1993. 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 Commun. 12 (3), 247–251.

Wang, D., Chen, J., 2018. Supervised speech separation based on deep learning: An overview. IEEE/ACM Trans. Audio, Speech, Lang. Process. 26 (10), 1702–1726.

Wang, Y., Narayanan, A., Wang, D., 2014. On training targets for supervised speech separation. IEEE/ACM Trans. Audio, Speech, Lang. Process. 22 (12), 1849–1858.

Williamson, D.S., Wang, Y., Wang, D., 2015. Complex ratio masking for monaural speech separation. IEEE/ACM Trans. Audio, Speech, Lang. Process. 24 (3), 483–492.

Woo, S., Park, J., Lee, J.-Y., So Kweon, I., 2018. CBAM: Convolutional block attention module. In: Proc. of the European Conf. on Computer Vision. ECCV, pp. 3–19.

Xu, Y., Du, J., Dai, L.-R., Lee, C.-H., 2014. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio, Speech, Lang. Process. 23 (1), 7–19.

Yin, D., Luo, C., Xiong, Z., Zeng, W., 2020. PHASEN: A phase-and-harmonics-aware speech enhancement network.. In: Association for the Advancement of Artificial Intelligence. AAAI, pp. 9458–9465.

Yu, F., Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Zhao, H., Zarar, S., Tashev, I., Lee, C.-H., 2018. Convolutional-recurrent neural networks for speech enhancement. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing. ICASSP, pp. 2401–2405.

Zheng, N., Zhang, X.-L., 2018. Phase-aware speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio, Speech, Lang. Process. 27 (1), 63–76.