CompNet: Complementary network for single-channel speech enhancement

Cunhang Fan\textsuperscript{a}, Hongmei Zhang\textsuperscript{a}, Andong Li\textsuperscript{b}, Wang Xiang\textsuperscript{a}, Chengshi Zheng\textsuperscript{b}, Zhao Lv\textsuperscript{a,*}, Xiaopei Wu\textsuperscript{a,*}

\textsuperscript{a} Anhui Province Key Laboratory of Multimodal Cognitive Computation, School of Computer Science and Technology, Anhui University, Hefei 230601, China
\textsuperscript{b} Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, 100190, Beijing, China

A R T I C L E I N F O

Keywords:
Speech enhancement
Complementary
Filtering and refining
Time–frequency domain
Time-domain

A B S T R A C T

Recent multi-domain processing methods have demonstrated promising performance for monaural speech enhancement tasks. However, few of them explain why they behave better over single-domain approaches. As an attempt to fill this gap, this paper presents a complementary single-channel speech enhancement network (CompNet) that demonstrates promising denoising capabilities and provides a unique perspective to understand the improvements introduced by multi-domain processing. Specifically, the noisy speech is initially enhanced through a time-domain network. However, despite the waveform can be feasibly recovered, the distribution of the time–frequency bins may still be partly different from the target spectrum when we reconsider the problem in the frequency domain. To solve this problem, we design a dedicated dual-path network as a post-processing module to independently filter the magnitude and refine the phase. This further drives the estimated spectrum to closely approximate the target spectrum in the time–frequency domain. We conduct extensive experiments with the WSJ0-SIB4 and VoiceBank + Demand datasets. Objective test results show that the performance of the proposed system is highly competitive with existing systems.

1. Introduction

Speech enhancement aims at removing background noise from the noisy mixture to improve the quality and intelligibility of speech. In real life, speech is often polluted by various noises that can seriously degrade the performance of automatic speech recognition (ASR) (Chan, Jaitley, Le, & Vinylas, 2016; Deng et al., 2022; Zhang, Liu, Huang, & Zhao, 2023), speech coding (Kleijn et al., 2021, 2021; Zhang, Zhang, Zhao, & Du, 2023), and hearing aids (Chen, Feng, Zhang, Lu, & Liu, 2022; Koike-Akino et al., 2023; Van Tasell, 1993). Therefore, speech enhancement is crucial for these back-end tasks. Traditional speech enhancement methods, such as Wiener filtering (Guo & Li, 2023), spectral subtraction (Boll, 1979; Liu, Chen, Yang, & Xie, 2022), and computational auditory scene analysis (CASA) (Rouat, 2008), have been proposed in the past few decades. However, these algorithms suffer from significant performance degradation when the noise is highly nonstationary or the signal-to-noise ratio (SNR) becomes relatively low.

We have witnessed the significant performance improvement of the speech enhancement thanks to the rapid development of deep neural networks (DNNs). Based on DNNs, speech enhancement methods can be broadly categorized into three classes: magnitude, time, and complex domains. Among them, both the magnitude and complex domains belong to the time–frequency (T–F) domain category. For the magnitude domain, the time-domain signal is converted into its T–F representation, and the magnitude-related component is then extracted as both the input and target. Depending on the output of magnitude domain networks, these magnitude domain methods can be divided into mapping-based and masking-based. The mapping-based method uses spectral magnitude or its compressed version as output to estimate clean speech (Han, Wang, & Wang, 2014), while masking-based approaches use masks such as ideal binary mask (IBM) (Kim & Lee, 2023), ideal ratio mask (IRM) (Chen, Chen, Zhang, Hu, & Zhao, 2022; Hu, Fan, & Lu, 2022; Li, Li, Li, & Li, 2021; Wang, Narayanan, & Wang, 2014), and ideal magnitude mask (IAM) (Wang & Chen, 2018a; Xie, Xie, Chen, Zhang, & Li, 2022) as outputs. Paliwal, Wójcicki, and Shannon (2011) have demonstrated the significance of accurate phase estimation in perceptual speech quality improvements. However, in traditional speech enhancement methods, phase recovery is usually not considered as the wrapping effect makes the phase distribution unstructured and difficult to estimate, if not impossible, which inevitably limits the performance upper bound of existing speech enhancement methods. To this end, time domain and complex-domain methods are developed to jointly recover magnitude and phase.

Time-domain methods aim to explicitly or latently model the distribution of waveform samples via feature encoding and decoding.
operations (Abdelaziz, Gong, & Stylianou, 2021; Luo & Megarsani, 2019; Pandey & Wang, 2019; Zhang, Chen, et al., 2022). However, the magnitude and phase are only implicitly handled in the optimization process of the waveform samples, they lack explicit modeling, and thus minimizing the distortion in the time domain cannot guarantee the estimation accuracy of speech spectrum. In contrast, complex-domain based methods (Hu, Liu, et al., 2020; Tan & Wang, 2020; Zhang, Gao, & Liu, 2022) convert the joint optimization toward the magnitude and phase into that of the real and imaginary (RI) components, which exhibit similar spectral structure to the spectral magnitude. Therefore, both the magnitude and phase can be effectively recovered, which partly explains the superiority of the complex-domain methods over magnitude-domain methods. Note that both magnitude-domain and complex-domain methods belong to the T–F domain and here we distinguish between them only for better illustrations.

Studies using time-domain or complex-domain methods consider phase information to some extent, but the magnitude of the speech tends to compensate for inaccurate phase estimation (Wang, Wichern, & Le Roux, 2021), resulting in limited enhancement performance. For this reason, multi-stage strategies are proposed to investigate the performance improvement from progressive or multi-domain perspective. CompNet, as the default choice in our experiments.}

To overcome these limitations, we propose CompNet, a single-channel speech enhancement framework. Unlike previous multi-stage networks, CompNet achieves its ultimate goal from a complementary perspective. CompNet consists of two components: a time-domain pre-processing module and a T–F domain post-processing module, which are trained in an end-to-end manner. The pre-processing module uses the time-domain network to enhance the speech waveform and optimize the sampling points, but the magnitude and phase at this stage may not reach their optimal solution. To solve this problem, we introduce a post-processing module in the T–F domain. This module is devised as a parallel branching structure to optimize magnitude and phase independently (Li, Liu, et al., 2021; Yin, Luo, Xiong, et al., 2020), aiming to obtain the better solution as much as possible. As such, the proposed CompNet can achieve the feasible enhancement performance with the help of the complementary nature between the two modules.

The contributions of this paper are two-fold. First, different from previous works with naive cascading multiple networks without rationality illustrations, this paper attempts to provide a different perspective to understand the improvements caused by multi-domain processing. Specifically, in the time-domain, despite the waveform samples can be feasibly recovered, the distribution of the T–F bins may still be partly different from the target spectrum. Therefore, we specially devise a dual-path network as the post-processing module to further push the estimated spectrum to get close to the target as much as possible in the frequency domain. Second, we follow the “from-coarse-to-fine” concept for pipeline design. After the processing by the first stage, some residual noise can still exist and thus we adopt the second network for further noise suppression. We conduct experiments on the WSJ0-S184 dataset and the VoiceBank + Demand dataset to evaluate the proposed approach. The experimental results demonstrate the highly competitive nature of CompNet compared to existing systems.

This article is structured as follows: Section 2 presents the problem formulation, followed by a detailed description of the proposed framework in Section 3. The experimental setup is outlined in Section 4, while Section 5 covers the results and analysis. Section 6 gives the discussions. Finally, conclusions are presented in Section 7.
3.2. Nested U-structure

The $U$-Net model was first proposed in Olaf et al. (2015) and is now widely used in the field of speech processing (Choi et al., 2021; Guimarães et al., 2020; Xu, Xu, Kong, & Xu, 2022). The $U$-Net consists of a contracting path and an expansive path. Each layer of the contracting path contains a downsampling convolution, a normalization layer, and a PReLU activation function. Similarly, each layer in the expansive path consists of an upsampling convolution, followed by a normalization layer and a PReLU activation function. In $U$-Net, skip connections are used to connect the feature maps of each layer in the contracting path to the feature maps of the same size in the expansive path, facilitating the flow of information through the network.

Recently, a new version of $U$-Net called $U$-Net was introduced in Qin and Zhang (2020). $U$-Net adopts a two-layer nested U-shaped structure, with a similar overall framework as $U$-Net. In contrast, the $U$-Net features residual u-blocks (RSUs) with different receptive field sizes at each stage. This architecture allows for capturing more contextual information at different scales, thereby endowing $U$-Net with rich multi-scale features. Therefore, in this study, we use the $U$-Net architecture for experiments to achieve better performance.

3.3. The proposed framework

The design of the speech enhancement framework is depicted in Fig. 2. The CompNet consists of two fundamental modules: a time domain pre-processing module and a T–F domain post-processing module. The original noisy speech first passes the time-domain preprocessor for initial estimation. The T–F domain post-processing module then filters the magnitude and refines the phase. In the following subsections, the time domain pre-processing module and the T–F domain post-processing module are described in detail.

3.3.1. Preprocessing module in the time domain

We use a temporal convolutional neural network (TCNN) structure as the time domain preprocessing module, whose input is the original noisy speech frames,

\[ \hat{S}_{\text{time}} = \text{TCNN}(y; \Phi_1) \]  

where $\text{TCNN}(\cdot; \Phi_1)$ denotes the time-domain network function and $\hat{S}_{\text{time}}$ denotes the speech processed by the time-domain network. TCNN consists of three components: the encoder, the decoder, and the S-TCMs. The TCNN module is portrayed in Fig. 2. The encoder–decoder structures are symmetric, each of which using a stack of five 2D gated convolutional (2D-ConvGLU) layers. Both the encoder and decoder have the same step size and convolution kernel, and the difference is that the encoder uses convolution to downsample along the frequency axis whereas the decoder uses transposed convolution. Each convolution is followed by Layer Normalization (LN) and PReLU. TCNN uses S-TCMs in the bottleneck to model the time dependency. The skip connection is adopted between the encoder and decoder.

Before feeding into the T–F domain network, we also need to combine the preprocessed magnitude $|\hat{S}_{\text{mag}}|$ with the original phase $\hat{\phi}_y$ as $\hat{S}_{\text{mag}}$, and combine the preprocessed phase $\hat{\phi}_{\text{time}}$ with the original magnitude $|Y|$ as $\hat{S}_{\text{phase}}$.

\[
\hat{S}_{\text{phase}} = |Y| \cos(\hat{\phi}_{\text{time}}) + j|Y| \sin(\hat{\phi}_{\text{time}}) 
\]

where $\hat{S}_{\text{phase}}$ and $\hat{S}_{\text{mag}}$ refer to the complex spectrum after preliminary enhancement of the phase and the complex spectrum after magnitude enhancement, respectively. While $\hat{S}_{\text{mag}}$ estimates the magnitude but requires phase correction, $\hat{S}_{\text{phase}}$ estimates the phase but needs magnitude refinement. Thus, these two components form a complementary pair. The complementary pair encompasses both the information estimated by the time-domain model and the original noisy information, thereby jointly participating in the processing of the second stage.

3.3.2. Postprocessing module in the T-F domain

It is worth noting that the T–F domain post-processing module is not a symmetric structure, consisting mainly of the encoder part of the U-Net ($U^2$-Encoder), the Gain Branch, the Resi Branch, and the decoder. The inputs of the T–F domain post-processing module are $\hat{S}_{\text{phase}}$ and $\hat{S}_{\text{mag}}$, and its main objective in the T–F domain is magnitude filtering and phase refining, as shown in Fig. 2. The $U^2$-Encoder contains four RSUs and a gated convolutional layer, which we use to extract the features,

\[ En_y = \text{Encoder}_{U^2}(\text{Cat}(\hat{S}_{\text{phase}}, \hat{S}_{\text{mag}})) \]

where $\text{Cat}(\cdot)$ denotes the concatenation function and $\text{Encoder}_{U^2}(\cdot)$ is the encoder function with the $U^2$-Net structure. The output obtained through the encoder is denoted as $En_y$. The Gain Branch is responsible for estimating the gain to facilitate magnitude filtering, whereas the Resi Branch is responsible for estimating the residual component for phase refining.

\[
gain = \text{GainBranch}(En_y, \hat{S}_{\text{mag}}; \Phi_2) 
\]

\[ \text{resi} = \text{ResiBranch}(En_y, \hat{S}_{\text{phase}}; \Phi_3) \]

where $\text{GainBranch}(\cdot)$ refers to the Gain Branch function and $\text{ResiBranch}(\cdot)$ refers to the Resi Branch function.

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Input size</th>
<th>Hyperparameters</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convglu 1</td>
<td>1 x T x 320</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 159</td>
</tr>
<tr>
<td>Convglu 2</td>
<td>64 x T x 159</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 79</td>
</tr>
<tr>
<td>Convglu 3</td>
<td>64 x T x 79</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 39</td>
</tr>
<tr>
<td>Convglu 4</td>
<td>64 x T x 79</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 19</td>
</tr>
<tr>
<td>Convglu 5</td>
<td>64 x T x 19</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 9</td>
</tr>
<tr>
<td>Convglu 6</td>
<td>64 x T x 9</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 4</td>
</tr>
<tr>
<td>Reshape_1</td>
<td>64 x T x 4</td>
<td>–</td>
<td>256 x 7</td>
</tr>
<tr>
<td>S-TCMs</td>
<td>256 x T</td>
<td>[1, 2, 5, 9] x 3</td>
<td>256 x 7</td>
</tr>
<tr>
<td>Convglu 2</td>
<td>256 x T</td>
<td>–</td>
<td>64 x T x 4</td>
</tr>
<tr>
<td>Convglu 3</td>
<td>128 x T x 4</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 9</td>
</tr>
<tr>
<td>Convglu 4</td>
<td>128 x T x 9</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 39</td>
</tr>
<tr>
<td>Convglu 5</td>
<td>128 x T x 19</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 79</td>
</tr>
<tr>
<td>Convglu 6</td>
<td>128 x T x 79</td>
<td>(2, 3), (1, 2), 64</td>
<td>64 x T x 320</td>
</tr>
</tbody>
</table>
all use the S-TCMs structure to model the time dependence. The obtained gain and resi are subsequently utilized for collaborative restoration of both \( S_1 \) and \( S_2 \): 
\[
S_1 = \text{gain} \ast S_{\text{phase}} \\
S_2 = \text{resi} + S_{\text{mag}}
\] (11)

We extract the real and imaginary parts of \( S_1 \) and \( S_2 \) separately. Then, combine the real parts of \( S_1 \) and \( S_2 \) to derive the real, and the imaginary parts of \( S_1 \) and \( S_2 \) to derive the imag. Finally, real and imag are fed into a decoder consisting of linear layers, respectively, 
\[
\text{real} = \text{Decoder}_{\text{real}}(\text{GRU}(\text{real}))
\] (13)
\[
\text{imag} = \text{Decoder}_{\text{imag}}(\text{GRU}(\text{imag}))
\] (14)

where GRU(*) is a recurrent neural network (RNN) function (Chung et al., 2014) that we utilize to process the speech information before passing it into the decoder. Decoder\( e_{\text{real}}(*) \) serves as the decoder for the real part, while Decoder\( e_{\text{imag}}(*) \) functions as the decoder for the imaginary part. Finally, we combine the real and imag to obtain the post-processed speech signal in the T–F domain.

Tables 1 and 2 summarize the network design details for each module. Layer names indicate the function and location of the corresponding layer or block. The input is specified with (Channels × TimeSteps × FreqChannels) for 3D-format and (FreqChannels × TimeSteps) for 2D-format. For the U\(^2\)-Encoder, the hyperparameters are specified with (KernelSize1, KernelSize2, Stride, T, Channels) format. Here, KernelSize1 and KernelSize2 respectively denote the kernel size of 2D-ConvGLU and the convolutional kernel size within the U-Net block. For one-dimensional convolutions (1D-conv), the hyperparameters are specified as (KernelSize, Channel). In GRU and Linear layers, the hyperparameters only include the number of output channels. In S-TCMs, the hyperparameters are specified in [DilationRate] format and the kernel size for dilation convolution is 5.

### 3.4. Loss function

The training objective of our CompNet consists of two parts, corresponding to the outputs generated by the two modules. In our time-domain preprocessing module, we train the time-domain network with

\[
\text{SNR loss until convergence,}
\begin{align*}
\hat{s}_{\text{target}} & := \frac{\langle s \rangle}{\|s\|^2} \\
e_{\text{noise}} & := \hat{s} - \hat{s}_{\text{target}} \\
\text{SNR} & := 10 \log_{10} \frac{\|\hat{s}_{\text{target}}\|^2}{\|e_{\text{noise}}\|^2} 
\end{align*}
\] (15)

where \( \hat{s} \) denotes the predicted speech, \( s_{\text{target}} \) is the target clean speech, \( \|s\|^2 = (s, s) \) denotes the signal power.

In the T–F domain post-processing module, we explicitly separate the magnitude and phase. Therefore, the T–F domain loss is defined as:

\[
\begin{align*}
\mathcal{L}_{\text{RI}} & = \left\| \hat{s}_{\text{RI}} - s_{\text{RI}} \right\|_F^2 + \left\| \hat{s}_{\text{RI}} - s_{\text{RI}} \right\|_F^2 \\
\mathcal{L}_{\text{Mag}} & = \left\| \sqrt{|\hat{s}_{\text{RI}}|^2 + |\hat{s}_{\text{RI}}|^2} - \sqrt{|s_{\text{RI}}|^2 + |s_{\text{RI}}|^2} \right\|_F^2 \\
\mathcal{L}_{\text{T–F}} & = \alpha \mathcal{L}_{\text{Mag}} + (1 - \alpha) \mathcal{L}_{\text{RI}} 
\end{align*}
\] (16)

In our CompNet, the time domain preprocessing module and the T–F domain post-processing module are trained jointly. This training method helps to back-propagate the gradient from the T–F domain post-processing module to the time domain preprocessing module. Thus, the total loss can be expressed as:

\[
\mathcal{L}_{\text{total}} = \gamma_1 \text{SNR} + \gamma_2 (\alpha \mathcal{L}_{\text{Mag}} + (1 - \alpha) \mathcal{L}_{\text{RI}}) 
\] (17)

where parameter \( \alpha \) is set to 0.5 by default, parameter \( \gamma_1 \) is set to 0.2 and parameter \( \gamma_2 \) is specified as 1.

### 4. Dataset and experimental setup

#### 4.1. Dataset preparation

To evaluate the performance of CompNet, we conducted extensive experiments using two datasets: WSJ0-SI84 (Paul & Baker, 1992) and VoiceBank + Demand Valentini-Botinhao, Wang, Takaki, and Yamagishi (2016). WSJ0-SI84 consists of 7138 clear speech samples from 83 speakers, with 41 female and 42 male speakers. We selected 77 speakers, and used 5428 and 957 clear speech samples for training and validation, respectively. For testing, we used two sets of 150 speech samples each, from six speakers (three male and three female). In the
first set, the speaker’s speech data was included in the training set, and we referred to these speakers as visible speakers. In the second set, the speaker’s information was not included in the training set, and we referred to these speakers as invisible speakers. The purpose of using invisible speakers was to test the model’s ability to generalize to different voicebanks. In the second set, the speaker’s speech data was included in the training set, and we referred to these speakers as visible speakers. In the second set, the speaker’s speech data was included in the training set, and we referred to these speakers as visible speakers. In the second set, the speaker’s speech data was included in the training set, and we referred to these speakers as visible speakers. In the second set, the speaker’s speech data was included in the training set, and we referred to these speakers as visible speakers.

To ensure stability during training, all speech recordings were sampled at a frequency of 16 kHz. Recordings longer than 8 s were truncated to 8 s, while recordings shorter than 8 s were padded with zeros. We extracted frames using a rectangular window size of 20 ms and an overlap time of 10 ms. For the purpose of STFT operations, we utilized a 20 ms Hamming window with a 50% overlap between adjacent time frames. This resulted in the utilization of a 320-point STFT, which produced a 161-dimensional spectrum. Recently, power compression techniques have been shown to improve performance (Li, Zheng, et al., 2021), and we apply this strategy to the input and target, i.e. $|Y|^{p_{\text{er}}} = |S|^{p_{\text{er}}}$, where $\beta = 0.5$ is considered empirically optimal.

During the training process, we established a batch size of 8 and utilized the Adam optimizer (Kingma & Ba, 2015) for stochastic gradient descent optimization with $\beta_1$ set to 0.9 and $\beta_2$ set to 0.999. To prevent an asymptotic explosion, we applied asymptotic clipping with a maximum value of 5.0. The model was trained for 60 periods with an initial learning rate of 0.0005. If the validation loss did not decrease for two consecutive epochs, the learning rate was halved. If the validation loss did not decrease for three consecutive epochs, the network would be stopped early.

### 4.3. Baseline models

This work conducts experiments on two different datasets, and we do not use the same baselines for our experiments. In the experiments based on the WSJ0-SI84 dataset, we selected eight baselines. Four of these baselines are intended for causal systems, which include LSTM (Chen, Wang, oho, Wang, & Healy, 2016), convolutional recurrent neural network (CRN) (Tan & Wang, 2018), gated convolutional recurrent networks (GCRN) (Tan & Wang, 2020), and ConvTasNet (Luo & Mesgarani, 2019). The LSTM and CRN models operate in the magnitude domain. The CRN model has an encoder-decoder structure with an intermediate LSTM for timing modeling. The GCRN method is an improved version of CRN that replaces regular convolutions in the encoder and decoder with their GLU versions. ConvTasNet is a time domain model, an advanced end-to-end speaker separation system that predicts waveform samples directly rather than using STFT for conversion. It has excellent performance, even for speech enhancement tasks. The other four baselines are for non-causal systems such as BLSTM, BCRN, BGCRN, and the non-causal ConvTasNet. BLSTM, BCRN, and BGCRN are similar to LSTM, CRN, and GCRN, but with bidirectional versions of LSTM replacing all LSTM layers. Noncausal ConvTasNet allows the use of future information. Each individual baseline model operates with the default parameter settings as prescribed in this research paper. Note that ConvTasNet originally used an 8 kHz sampling rate, but was extended to 16 kHz for speech enhancement. All models were trained on a GPU with a Pytorch platform (Paszke et al., 2017).

In the context of experiments conducted on the VoiceBank + Demand dataset, twelve different baseline models were selected for comparative analysis. The first group of models focused on speech enhancement through the use of generative adversarial techniques in the T-F

### Table 2

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Input size</th>
<th>Hyperparameters</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRU</td>
<td>322 x 161</td>
<td></td>
<td>161 x 161</td>
</tr>
<tr>
<td>Linear</td>
<td>161 x 161</td>
<td></td>
<td>161 x 161</td>
</tr>
</tbody>
</table>

### Gain Branch

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Input size</th>
<th>Hyperparameters</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reshape</td>
<td>64 x T x 4</td>
<td></td>
<td>417 x T</td>
</tr>
<tr>
<td>Conv1d</td>
<td>417 x T</td>
<td>1, 256</td>
<td>256 x T</td>
</tr>
<tr>
<td>S-TCMs</td>
<td>256 x [1, 2, 5, 9] x 3</td>
<td>256 x T</td>
<td></td>
</tr>
<tr>
<td>Conv1d</td>
<td>256 x 1, 161</td>
<td>161 x 161</td>
<td></td>
</tr>
<tr>
<td>Magnitude</td>
<td>161 x T</td>
<td></td>
<td>2 x T x 161</td>
</tr>
</tbody>
</table>

### Rest Branch

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Input size</th>
<th>Hyperparameters</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reshape</td>
<td>64 x T x 4</td>
<td></td>
<td>518 x T</td>
</tr>
<tr>
<td>Conv1d</td>
<td>518 x T</td>
<td>1, 256</td>
<td>256 x T</td>
</tr>
<tr>
<td>S-TCMs</td>
<td>256 x [1, 2, 5, 9] x 3</td>
<td>256 x T</td>
<td></td>
</tr>
<tr>
<td>Conv1d</td>
<td>256 x 1, 322</td>
<td>322 x T</td>
<td></td>
</tr>
<tr>
<td>Phase</td>
<td>322 x T</td>
<td></td>
<td>2 x 161 x T</td>
</tr>
</tbody>
</table>

### Real_Decoder

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Input size</th>
<th>Hyperparameters</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRU</td>
<td>322 x 161</td>
<td></td>
<td>161 x 161</td>
</tr>
<tr>
<td>Linear</td>
<td>161 x 161</td>
<td></td>
<td>161 x 161</td>
</tr>
</tbody>
</table>

### Imag_Decoder

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Input size</th>
<th>Hyperparameters</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRU</td>
<td>322 x 161</td>
<td></td>
<td>161 x 161</td>
</tr>
<tr>
<td>Linear</td>
<td>161 x 161</td>
<td></td>
<td>161 x 161</td>
</tr>
</tbody>
</table>
Regarding the ablative study of different post-processing types and encoders in post-processing networks. “vanilla” represents the simultaneous processing of magnitude and phase, while “collaborative” represents the separate processing of magnitude and phase. “u” represents the use of U-Net architecture as the encoder in the post-processing network. “v” represents the use of U-Net architecture as the encoder in the post-processing network.

### Evaluation metrics

Depending on the dataset, we have used different metrics in this paper. The ablation experiments were specifically evaluated on the WSJO-SIB4 dataset (Paul & Ruder, 1992) using perceptual evaluation of speech quality (PESQ) (Rix, Beerends, Hollier, & Hekstra, 2001), extended short-term objective intelligibility (ESTOI) (Jensen & Taal, 2016), and signal-distortion ratio (SDR) (Vincent, Savadjiev, Boffil, Makino, & Rosca, 2007). For comparison with other baselines, the SDR was replaced by DMS-MOS (Naderi & Cutler, 2021), a recently proposed metric that follows ITU-T Rec.P.835 and is highly correlated with subjectively rated human ratings. The PESQ provides a measure of speech quality ranging from −0.5 to 4.5, while the ESTOI measures subjective intelligibility as a percentage between 0 and 1. The SDR is widely recognized for blind source separation and is a valid indicator of the degree of speech distortion.

On the other hand, for the VoiceBank + Demand dataset, we used wideband PESQ (Recommendation ITU-T P ITU, 2007) and three additional target metrics associated with MOS (Hu & Loizou, 2007), namely CSIG, CBAK, and COVL, to assess speech quality. For all metrics used, we report the use of wideband PESQ (Recommendation ITU-T P ITU, 2007) and three additional target metrics associated with MOS (Hu & Loizou, 2007), namely CSIG, CBAK, and COVL, to assess speech quality. For all metrics used, we report the use of wideband PESQ (Recommendation ITU-T P ITU, 2007) and three additional target metrics associated with MOS (Hu & Loizou, 2007), namely CSIG, CBAK, and COVL, to assess speech quality. For all metrics used, we report the use of wideband PESQ (Recommendation ITU-T P ITU, 2007) and three additional target metrics associated with MOS (Hu & Loizou, 2007), namely CSIG, CBAK, and COVL, to assess speech quality.
C. Fan et al.

Fig. 3. PESQ, ESTOI, and SDR under different noisy conditions for different models. The values in each input SNR are averaged from all the seen test set 1. The input SNR value ranges from $-6$ dB to 6 dB with an interval of 3 dB.

Table 4

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>MACs</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>29.04 M</td>
<td>2.94 G/s</td>
</tr>
<tr>
<td>CRN</td>
<td>17.58 M</td>
<td>2.44 G/s</td>
</tr>
<tr>
<td>GCRN</td>
<td>9.77 M</td>
<td>2.42 G/s</td>
</tr>
<tr>
<td>ConvTasNet</td>
<td>5.00 M</td>
<td>2.53 G/s</td>
</tr>
<tr>
<td>CompNet</td>
<td>4.26 M</td>
<td>5.92 G/s</td>
</tr>
</tbody>
</table>

In ablative experiment one, we found that the results were not affected by the noise or test dataset. Therefore, in ablative experiment two, we only enhanced the visible mixed speech. Ablative experiment two investigated the functions of CompNet at different stages. From Fig. 3, we can obtain the following observations. "First_phase" refers to using TCNN to enhance noisy speech, while "Second_phase" means filtering the magnitude and correcting the phase of the noisy speech using a T–F domain network.

- Firstly, CompNet consistently outperforms both the first and second stages in terms of PESQ and ESTOI, highlighting the importance of complementarity. Specifically, compared to the second stage, the speech enhanced by CompNet exhibits better optimization at the sample level, while compared to the first stage, CompNet can supplement the details of the speech.

- Secondly, time-domain networks tend to perform better on SDR, but CompNet outperforms the First_phase on SDR at low SNRs and performs similarly to it at high SNRs. This indicates that CompNet has a very stable and effective speech enhancement capability at low SNRs.

5.2. Model complexity comparison

For the causal speech enhancement system, we need to consider several statistics in a realistic scenario, such as the model size, and the number of multiplicative accumulation operations (MACs) per second. Table 4 lists the four causal baselines as well as the data sizes of the proposed CompNet. It is important to note that input samples are all set to one second of audio, so our experiments are fair. According to Table 4, CompNet has smaller model parameters but is relatively more computationally intensive.

5.3. Comparisons with baselines on WSJ0-SI84 corpus

On the WSJ0-SI84 test set 2, we evaluated the proposed CompNet against some classical baseline models. The test set 2 is divided into a visible speaker dataset and an unseen speaker dataset. Evaluation indicators: PESQ, ESTOI, MOS_OVLR. We have a total of eight baselines, four of which are causal systems and the rest belong to non-causal systems.

In the seen test set 2, as shown in Table 5, both causal and non-causal systems could effectively remove background noise. Our proposed causal CompNet outperforms other causal baselines in all evaluation metrics. The non-causal CompNet performs worse than the non-causal ConvTasNet in the ESTOI and MOS_OVLR metrics but performs best in the PESQ metric.

In the evaluation set with unseen speakers, as shown in Table 6, we observed that all DNN-based speech enhancement models were able to effectively remove noise from untrained speakers under various conditions. Similar to the findings in Table 5, CompNet outperformed the other baseline models in all three evaluation metrics. However, in the context of non-causal systems, CompNet demonstrates the second highest performance in terms of ESTOI and MOS_OVLR.

Based on the analysis of the experimental results mentioned above, we conclude that the proposed CompNet effectively exploits the complementary characteristics to uncover the known information in causal
systems. Therefore, this framework holds significant research significance in the context of causal systems. Furthermore, the complementary network also exhibits remarkable performance in non-causal systems.

5.4. Comparisons with baselines on VoiceBank + demand benchmark

In addition to the above experiments, we also carried out experiments on the VoiceBank + Demand dataset. The CompNet were compared with other baselines, see Table 7. Note that the causal CompNet was used in the training process.

From Table 7, it can be seen that CompNet achieves decent overall performance across these metrics, indicating the effectiveness of our approach. However, the performance of CompNet is not outstanding, which can be explained from two aspects. The first reason may be attributed to the extensive downsampling employed in our modeling, which may result in insufficient modeling of fine details (Yu et al., 2022). The second reason is that we did not incorporate highly complex modules at the specific module level, such as sub-band RNN (Chen, Wang, et al., 2022) or attention mechanisms (Wang, Cornell, et al., 2022). Exploring these module-level considerations is left as future work beyond the scope of this paper.

6. Discussion

The above experimental results indicate that our proposed CompNet, not only improves the performance of speech enhancement but also offers a novel perspective to explain the mechanisms underlying multi-domain networks. The following are some observed results.

1. Compared to time-domain networks and T–F domain methods, using a multi-domain approach yields better speech enhancement performance. While time-domain networks only enhance the speech waveform of the noisy speech, there may still exist residual noise in the distribution of the T–F domain, which differs from the target spectrum. Hence, we found that time-domain networks and time–frequency domain networks can benefit from each other, leading to the proposal of CompNet. At the same time, CompNet showcases a novel perspective to comprehend the improvements brought by multi-domain systems.

2. Furthermore, through ablation experiments, it has been revealed that directly cascading the time-domain network and the T–F domain network is not the optimal approach. Therefore, we adhere to the “from coarse to fine” principle for speech enhancement. Specifically, we first utilize a time-domain network to estimate the speech waveform. Subsequently, we introduce a parallel branch in the T–F domain network to further refine the speech, enabling the attainment of optimal values for both magnitude and phase.

To summarize, our proposed CompNet for single-channel speech enhancement provides an explanatory framework for the mechanisms of multi-domain networks from a complementary perspective. Additionally, it follows the “from coarse to fine” principle, leading to better speech enhancement results.

7. Conclusion

This paper introduces a single-channel speech enhancement network called CompNet. CompNet initially utilizes a time-domain network to comprehensively enhance the waveform of noisy speech. Then, a dual-path network in the time–frequency domain is employed to further refine the estimated spectrum, aiming to approach the target speech more closely in the T–F domain. CompNet not only adheres to the “from coarse to fine” principle but also provides a different perspective to comprehend the improvements brought by multi-domain processing.

We conduct extensive experiments on the WSJ0-SIB4 and VoiceBank + Demand datasets. The experimental results demonstrate the superiority of the network on both causal and non-causal systems. We plan to extend the proposed complementary estimation concept into other speech front-end tasks, e.g., acoustic echo cancellation, and multi-channel speech enhancement. Furthermore, our future research will take a more integrated and comprehensive approach, concentrating not only on further advancing complementary estimation methods but also on developing signal protection strategies. This is due to the negative impact of distortion on downstream tasks, such as ASR.

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.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work is supported by the STI 2030—Major Projects (No. 2021DZD0201500), the National Natural Science Foundation of China (NSFC) (No. 61972437, No. 62201002), Distinguished Youth Foundation of Anhui Scientific Committee, China (No. 2208085J05), Special Fund for Key Program of Science and Technology of Anhui Province, China (No. 202203a070200008), Open Fund of Key Laboratory of Flight Techniques and Flight Safety, China, CACC (No. FZ2022KF15), the Open Research Projects of Zhejiang Lab, China (NO. 2021KH0AB06) and the Open Projects Program of National Laboratory of Pattern Recognition, China (NO. 202200014).

References


Neural Networks 168 (2023) 508–517

C. Fan et al.


IEEE Transactions on Neural Networks and Learning Systems, 19(1), 199.


Springer.


