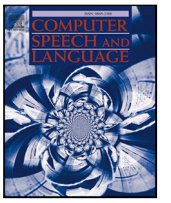
Computer Speech and Language 89 (2025) 101677

Contents lists available at ScienceDirect

Computer Speech & Language

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



A systematic study of DNN based speech enhancement in reverberant and reverberant-noisy environments

Heming Wang [a,*], Ashutosh Pandey [a], DeLiang Wang [a,b]

[a] The Ohio State University, 281 W Lane Ave, Columbus, 43210 OH, United States
[b] Center for Cognitive and Brain Science, 1835 Neil Ave, Columbus, 43210 OH, United States

ARTICLE INFO

Keywords:
Speech enhancement
Speech dereverberation
Self-attention
ARN
DC-CRN

ABSTRACT

Deep learning has led to dramatic performance improvements for the task of speech enhancement, where deep neural networks (DNNs) are trained to recover clean speech from noisy and reverberant mixtures. Most of the existing DNN-based algorithms operate in the frequency domain, as time-domain approaches are believed to be less effective for speech dereverberation. In this study, we employ two DNNs: ARN (attentive recurrent network) and DC-CRN (densely-connected convolutional recurrent network), and systematically investigate the effects of different components on enhancement performance, such as window sizes, loss functions, and feature representations. We conduct evaluation experiments in two main conditions: reverberant-only and reverberant-noisy. Our findings suggest that incorporating larger window sizes is helpful for dereverberation, and adding transform operations (either convolutional or linear) to encode and decode waveform features improves the sparsity of the learned representations, and boosts the performance of time-domain models. Experimental results demonstrate that ARN and DC-CRN with proposed techniques achieve superior performance compared with other strong enhancement baselines.

1. Introduction

In real-world environments, speech is corrupted by both background noise and room reverberation, which degrades speech quality and intelligibility. Speech enhancement aims to remove background noise and recover clean anechoic speech from reverberant-noisy mixtures. Reverberation is generated by sound reflections by surfaces including walls, floors and ceilings, and other objects. The speech signal received by a speaker or a far-field microphone is a summation of the direct sound and an infinite number of delayed and decayed copies of the original speech signal. Noisy signals refer to the addition of clean speech signals and noises, and are challenging for speech enhancement when the signal-to-noise (SNR) ratio is low. The task of speech dereverberation aims to recover the clean signal from reverberant speech. The distortion caused by both noise and reverberation creates difficulties for many speech-related tasks (Wang and Chen, 2018), including speaker separation (Aralikatti et al., 2021), automatic speech recognition (Al-Karawi et al., 2015; Heymann et al., 2019) and speaker identification (Zhao et al., 2014; Bai and Zhang, 2021).

Many studies have been published to tackle the speech dereverberation problem. A popular conventional approach to remove or attenuate room reverberation is the weighted prediction error (WPE) algorithm (Nakatani et al., 2010). WPE computes a filter based on delayed linear prediction to estimate room reverberation. Other conventional signal processing approaches include a relative transfer function identification method (Talmon et al., 2009), a Wiener-like filter based on estimated reverberation time (Habets et al., 2009), and employing the estimated power spectral density to estimate late reverberation (Braun et al., 2018).

* Corresponding author.
E-mail addresses: wang.11401@osu.edu (H. Wang), pandey.99@osu.edu (A. Pandey), dwang@cse.ohio-state.edu (D. Wang).

https://doi.org/10.1016/j.csl.2024.101677
Received 22 November 2023; Received in revised form 14 April 2024; Accepted 5 June 2024
Available online 6 June 2024

Since the introduction of deep learning, data-driven approaches are widely used to directly restore target speech (Kinoshita et al., 2007; Han et al., 2015; Zhao et al., 2018; Defossez et al., 2020; Wang and Wang, 2020a; Schroter et al., 2022; Li et al., 2023; Neri and Braun, 2023), and they substantially improve the enhancement performance compared with conventional methods. Many recent studies investigated the use of DNN for joint dereverberation and denoising. Yan and Wang (Zhao and Wang, 2020) proposed to perform speech enhancement in the reverberant-noisy conditions using a densely connected UNet (Huang et al., 2017). The network operates in the complex domain, and a time–frequency attention module is incorporated to aggregate contextual information across time–frequency units in complex feature maps. Li et al. (2021) proposed a Simultaneous Denoising and Dereverberation network (SDDNet) that utilizes a multi-stage model to progressively remove reverberation and then suppress background noise. The core idea of their network design is to decouple the training targets. SDDNet shows considerable improvement for real-time enhancement and is ranked the top in the 2021 Deep Noise Suppression (DNS) challenge (Reddy et al., 2021). Choi et al. (2021) employed a lightweight UNet model for speech enhancement, and achieved strong performance with a relatively small computational cost. During training, they used a composite loss function that consists of waveform cosine similarity and the mean absolute error between complex spectrograms. Fu et al. (2022) proposed UFormer with two branches: one operates in the magnitude spectrogram and the other in the complex domain. Attention modules and dual-path conformers are incorporated to model local and global temporal dependencies.

This study focuses on monaural speech enhancement and performs joint denoising and dereverberation. When reverberation is present, the highly varying nature of room impulse responses makes it difficult to separate the direct sound from its reflections in the time domain. Most of the existing deep learning approaches operate in the frequency domain (Zhao et al., 2020; Li et al., 2021; Schroter et al., 2022; Purushothaman et al., 2023). On the other hand, time-domain processing has been shown to be competitive in other speech processing tasks, for instance, automatic speech recognition (Kinoshita et al., 2020), speaker identification (Delcroix et al., 2020; Salvati et al., 2023) and speaker separation (Ravenscroft et al., 2023). Although existing studies show good performance on tailored evaluation datasets, comprehensive understanding is lacking in terms of how specific techniques contribute to enhancement performance, especially dereverberation performance. Our paper aims to fill this void. A related study by Cord-Landwehr et al. (2022) focuses on mask-based source separation. They employed the SepFormer model (Subakan et al., 2021) as their starting point and optimized its performance on reverberant mixtures. Surprisingly, SepFormer only marginally improves over a carefully optimized bidirectional long-short term memory (LSTM) baseline. Rather than advanced DNN architecture, the authors found that the effects of objective functions, window lengths and feature domains are more significant for reverberant speaker separation. Our work is different as it is not limited to mask-based models, and we focus on speech enhancement rather than speaker separation.

This paper aims to systematically investigate the effects of various factors on speech dereverberation and denoising performance. We examine two strong baseline models. One is ARN (attentive recurrent network) (Pandey and Wang, 2022), which is a time-domain model composed of recurrent neural networks (RNNs) with self-attention. The other is DC-CRN (densely-connected convolutional recurrent neural network) (Tan et al., 2021), which is a frequency-domain model consisting of convolutional operations and conducts complex spectral mapping. The recent ARN and DC-CRN architectures are representative speech enhancement models, employing two primary DNN architectures: RNN and CNN (convolutional neural network). Since most existing speech enhancement models utilize RNN or CNN, we believe that our findings can be extended to other DNN models for speech enhancement. To study the effect of input representation domains, we create the corresponding time-domain or frequency-domain variants for the two DNN architectures. Additionally, we investigate factors such as window lengths, objective functions, and transform layers. Our evaluation results suggest that, unlike the common wisdom that frequency domain approaches work better, the representation domain does not play a major role in speech dereverberation. For time-domain models, adding linear layers to encode and decode waveform features improves learned representations, which serves a similar function to short-time Fourier Transform (STFT) employed in frequency-domain networks. In summary, the main contributions of our work are three-fold. First, we show that a larger window size helps dereverberation performance, which is consistent with the conclusion in Cord-Landwehr et al. (2022). Second, we reveal that, for time-domain models, a transform layer is crucial in converting feature maps into an embedding space that is sparser and more separable, and benefits dereverberation. Lastly, we compare an optimized ARN architecture to other advanced baselines and achieve a significant performance advantage in terms of objective speech intelligibility and quality metrics.

The rest of the paper is organized as follows. In Section 2, we formulate the dereverberation and denoising problem. Section 3 describes DNN models and designs in detail. In Section 4, we present datasets and experimental setups used in our experiments. Experimental results, analyses, and comparisons are provided in Section 5. Section 6 concludes the paper.

2. Formulation

A reverberant-noisy speech signal $\mathbf{y}$ can be modeled as a combination of reverberant speech and noise $\mathbf{n}$. Reverberant speech is produced by a clean target speech signal $\mathbf{s}$ convolved with a room impulse response $\mathbf{h}$, which can be expressed as,

$$\mathbf{y}(k) = (\mathbf{s} * \mathbf{h})(k) + \mathbf{n}(k), \tag{1}$$

where $\{\mathbf{y}, \mathbf{s}, \mathbf{n}\} \in \mathbb{R}^{K\times 1}$, and K is the number of samples in the speech signal. We use k to index time samples, and $*$ to represent the convolution operator. From another perspective, we can divide $\mathbf{y}(k)$ into the target speech $\mathbf{x}(k)$, reverberant speech $\mathbf{r}(k)$ and background noise $\mathbf{n}(k)$, i.e.,

$$\mathbf{y}(k) = \mathbf{x}(k) + \mathbf{r}(k) + \mathbf{n}(k). \tag{2}$$

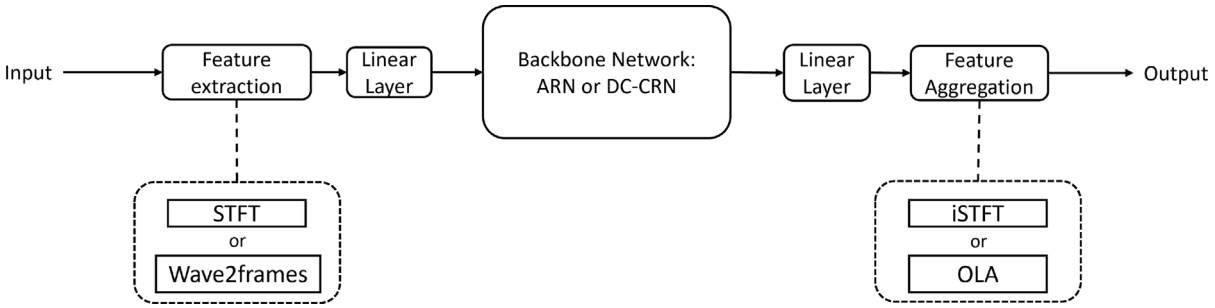


Fig. 1. Diagram of the proposed enhancement pipeline. The input and output are both waveform signals, and the features extracted are either STFT-based or time-domain overlapped frames depending on the domain of operation. A linear layer is used before and after the backbone network to transform features to a fixed embedding dimension. Feature aggregation is applied to transform the feature dimension back to the original input size.

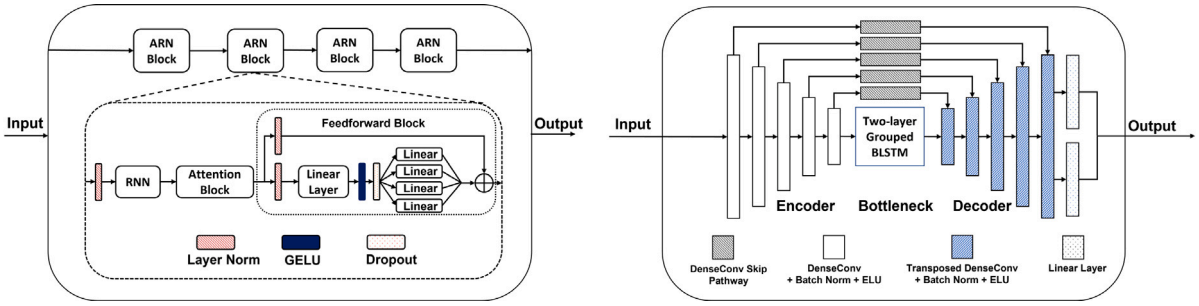


Fig. 2. Illustration of two backbone networks: (a). ARN is composed of RNN based blocks plus self-attention, (b). DC-CRN is built upon the UNet architecture and enhanced with densely connected convolutions. Input features have the same dimension as that of output features.

Note that the target speech $\mathbf{x}(k)$ is different from the dry clean signal, as it incorporates a small time shift and energy decay due to sound transmission, and may also include early reflections. Converting to the frequency domain by performing STFT, we have,

$$\mathbf{Y}[t, f] = \mathbf{X}[t, f] + \mathbf{R}[t, f] + \mathbf{N}[t, f], \tag{3}$$

where t and f index time frame and frequency bin, respectively. In this paper, our goal is to recover the target speech signal from a reverberant-noisy mixture, hence removing reverberation and background noise. With DNN model denoted as f and its parameters as θ, reverberant-noisy enhancement can be expressed as,

$$\hat{\mathbf{x}}(k) = f(\mathbf{y}(k), \theta). \tag{4}$$

$\hat{\mathbf{x}}(k)$ is the estimated target signal. For metric computation, we use $\mathbf{x}(k)$ as the reference target. For reverberant-only conditions, we do not take the background noise $\mathbf{n}$ into consideration.

3. Model description

3.1. DNN architectures

We adopt a general architecture for DNN based speech enhancement, and the overall pipeline is depicted in Fig. 1. For a given input signal, we first perform feature extraction through transform operations to convert the input to the domain of interest. If speech enhancement is performed in the time domain, input signals are chunked into overlapping frames (denoted as Wave2frame). Otherwise, we conduct STFT to convert signals to complex-domain spectrograms. Extracted feature maps are first encoded by a linear layer that transforms the feature dimension into a fixed embedding dimension, and then passed to a DNN block. We employ two DNN architectures, ARN (Pandey and Wang, 2022) and DC-CRN (Tan et al., 2021) as backbone models. Their detailed designs are described in the following subsections. The output of the DNN block is fed to another linear layer (decoder layer) that transforms the feature dimension back to its original size. Finally, we perform feature aggregation to reconstruct target speech signals, where we perform either inverse STFT (iSTFT) to complex-domain spectrograms or overlap-and-add (OLA) to waveform segments, depending on input type.

3.1.1. ARN

Fig. 2(a) illustrates the design of ARN, which is a time-domain network consisting of RNNs augmented by self-attention blocks and feedforward blocks. In recent years, the attention mechanism has been widely utilized in sequence-to-sequence tasks, such as automatic speech recognition (Povey et al., 2018), natural language processing (Vaswani et al., 2017; Galassi et al., 2020) and computer vision (Guo et al., 2022). The concept of attention is to create a context vector that highlights the salient part of a sequence and suppresses irrelevant information. Self-attention operates on the same sequence, and models the dependencies of different parts of the sequence (Shaw et al., 2018). Recent speech enhancement studies (Giri et al., 2019; Zhao et al., 2020) also report substantial performance gain by incorporating attention modules. The default network input is waveform signals. When input features are STFT complex vectors, the real and the imaginary parts of complex spectrograms are concatenated along the feature dimension to obtain real-valued vectors. We perform layer normalization on input features, which is an alternative to batch normalization and has shown to improve convergence and performance (Ba et al., 2016). Then normalized features pass through an RNN block, specifically, an LSTM network to model temporal dependencies. The attention block computes the self-attention of a given sequence. The output from the attention block passes through a feedforward block, which consists of linear layers, Gaussian error linear units (GELUs) (Hendrycks and Gimpel, 2016), and dropout (Srivastava et al., 2014). The linear layer output is split into four vectors and are then summated. With residual connections added, we finally obtain the output of an ARN block.

More details can be found in Pandey and Wang (2022). Following the original implementation, we use four ARN blocks and set the embedding dimension to 1024. A dropout rate of 5% is used in the feedforward module. The number of trainable parameters of ARN is 55.7 millions.

3.1.2. DC-CRN

DC-CRN performs complex spectral mapping and is developed originally for the convolutional recurrent network (CRN) architecture, which is a strong enhancement model that consists of a convolutional encoder and decoder. In the bottleneck of CRN, a recurrent network is employed to model temporal dependencies. Specifically, a grouped BLSTM (Gao et al., 2018) is applied to improve memory and computational efficiency. To capture the benefits of dense connections (Huang et al., 2017; Pandey and Wang, 2020), a densely-connected (DC) convolutional block is adopted in place of a standard convolution in both the encoder and decoder, as illustrated in Fig. 2(b). The core idea of a DC block is to reuse feature maps by splitting a convolution layer into numerous layers with fewer channels, and each layer receives the output from all preceding layers. The information flow is improved by this design since all convolution layers are directly connected. A 2D convolution, a batch normalization layer (Ioffe and Szegedy, 2015), and an exponential linear unit (ELU) activation function (Clevert et al., 2016) are the components of a DC block. Within a DC block, the number of the output channels of the first four convolutional layers is set to 8. All previous outputs are received by the final layer and are aggregated with a gated convolution. Unlike CRN that directly concatenates the output of a decoder layer and the corresponding encoder layer as skip connections (Tan and Wang, 2018; Hu et al., 2020), DC-CRN employs a DC block to the encoder output prior to concatenating it with the output of the decoder layer. Such a design helps the fusion of feature maps from both the encoder and the decoder and refines the intermediate representations gradually. For the network input of STFT complex vectors, we stack the real and imaginary parts as separate convolutional channels. The output of DC-CRN is split into two halves and fed to two linear layers to produce real and imaginary estimates separately. When input features are waveform signals, we only use one convolution channel and adopt one linear layer to the DC-CRN output before reshaping it to the waveform. Following the implementation in Tan et al. (2021), we adopt five DC blocks in both encoder and decoder parts and set the embedding dimension to 161. Within the bottleneck, each BLSTM layer has 640 units in either direction. The original numbers of convolution channels are 2, 16, 32, 64, 128, 256 for the encoder, and mirrored for the decoder (see Tan et al. (2021)). To obtain better enhancement performance and meet GPU memory constraints, we increase convolution channels to 2, 24, 48, 96, 192, 384, respectively, resulting in 18.12 million trainable parameters for DC-CRN.

3.2. Loss functions

We examine commonly used loss functions in speech enhancement studies, which include training objectives in both time and spectral domains. The default loss employed in our experiments is the phase-constrained magnitude (PCM) loss introduced in Pandey and Wang (2021), which takes into account both target speech and residual noise. Specifically, the PCM loss $\mathcal{L}_{PCM}$ requires a reasonable estimate of both speech and residual noise with respect to STFT magnitudes in $\mathcal{L}_1$ norm, which is defined as,

$$\begin{aligned}\mathcal{L}_{PCM} =& \frac{1}{TF}\sum_{t,f}||\hat{\mathbf{X}}_r(t,f)+\hat{\mathbf{X}}_i(t,f)|-|\mathbf{X}_r(t,f)+\mathbf{X}_i(t,f)||\\ &+\frac{1}{TF}\sum_{t,f}||(\mathbf{Y}_r(t,f)-\hat{\mathbf{X}}_r(t,f))+(\mathbf{Y}_i(t,f)-\hat{\mathbf{X}}_i(t,f))|-|(\mathbf{Y}_r(t,f)-\mathbf{X}_r(t,f))+(\mathbf{Y}_i(t,f)-\mathbf{X}_i(t,f))||,\end{aligned} \tag{5}$$

where $\hat{\mathbf{X}}$ denotes the STFT of a predicted target speech signal, and $\mathbf{Y}-\hat{\mathbf{X}}$ is the STFT of the estimated residual noise. Subscripts r and i denote the real and imaginary parts of a complex vector, respectively. Note that, in this context, a residual signal $\mathbf{Y}-\mathbf{X}$ contains both noise and speech reverberation. T and F are the total number of time frames and frequency bins, respectively. This loss is demonstrated to be effective in imposing a phase constraint on speech enhancement. By optimizing the magnitudes of target speech and residual simultaneously, the infinite number of candidates for the estimated speech complex vector is reduced to two due to a triangular magnitude relation (Pandey and Wang, 2021). It is also worth noting that the loss is measured in the spectral domain. For loss calculation, we divide a waveform signal into frames or segments with a frame size of 512 samples and a frame

shift of 128 samples, corresponding to an analysis window of 32 ms with a 25% overlap for the sampling frequency of 16 kHz. Then we multiply these frames with a Hanning window. The STFT vectors are calculated on windowed frames to define related terms in the frequency domain.

Another loss function $\mathcal{L}_{RI+MAG}$ proposed in Wang et al. (2020) measures the spectrogram similarity in the complex domain. Defined in terms of real and imaginary spectrograms, it measures the differences of the real and imaginary parts $\mathcal{L}_{RI}$ and the magnitude difference $\mathcal{L}_{MAG}$. $\mathcal{L}_{RI+MAG}$ is calculated by,

$$\mathcal{L}_{RI+MAG} = \mathcal{L}_{RI} + \mathcal{L}_{MAG} \tag{6}$$

$$\mathcal{L}_{RI} = \frac{1}{TF}\sum_{t,f} |\hat{\mathbf{X}}_r(t,f) - \mathbf{X}_r(t,f)| + |\hat{\mathbf{X}}_i(t,f) - \mathbf{X}_i(t,f)| \tag{7}$$

$$\mathcal{L}_{MAG} = \frac{1}{TF}\sum_{t,f} \|\hat{\mathbf{X}}(t,f)| - |\mathbf{X}(t,f)\|, \tag{8}$$

where $|\cdot|$ measures the STFT magnitude of a given vector. The rationale for incorporating a magnitude loss is the relative importance of magnitude over phase (Wang and Wang, 2020b; Wang et al., 2021b; Zhang et al., 2021). We use the same computation as PCM to obtain related complex vectors in the spectral domain.

For time-domain speech enhancement, we investigate two other popular loss functions. One is the mean absolute error (MAE) loss, defined as,

$$\mathcal{L}_{MAE} = \frac{1}{K}\sum_{k} |\hat{\mathbf{x}}(k) - \mathbf{x}(k)|. \tag{9}$$

The other one calculates the scale-invariant signal-to-noise ratio (SI-SNR) using the target waveform and the estimated waveform (Le Roux et al., 2019),

$$\mathbf{x}_{target} = \frac{\langle \hat{\mathbf{x}}, \mathbf{x}\rangle \mathbf{x}}{\|\mathbf{x}\|^2} \tag{10}$$

$$\mathbf{e}_{noise} = \hat{\mathbf{x}} - \mathbf{x}_{target} \tag{11}$$

$$\mathcal{L}_{SI\text{-}SNR} = -10\log_{10}\frac{\|\mathbf{x}_{target}\|^2}{\|\mathbf{e}_{noise}\|^2}, \tag{12}$$

where both the estimated signal $\hat{\mathbf{x}}$ and the target signal $\mathbf{x}$ are normalized to zero mean before loss computation, and $\langle \cdot, \cdot \rangle$ computes the dot product of two vectors. Note that the noise energy term in Eq. (12) refers to both background noise and reverberation.

4. Experimental setup

4.1. Datasets

We use the DNS `read_speech` dataset (Reddy et al., 2021) as our speech corpus, which consists of around 550 h of speech utterances. We choose utterances with low estimated reverberation time (i.e. less than 0.17 s) for our experiments, and set aside 10% speakers for validation and 10% speakers for testing, resulting in 39 225 utterances for training, 5176 utterances for validation and 5314 utterances for testing. For both training and validation, each reverberant-noisy mixture is generated by mixing a clean utterance convolved with a randomly picked RIR (room impulse response) and a noise segment. The clean utterance is convolved with the first 50 ms of the RIR in order to create the target signal. Data simulation for reverberant-only experiments is similar except that no noise is added. Using the `RIR-Generator` package,[1] we artificially generate 12 000 RIRs with a T_{60} (reverberation time) value randomly sampled from 0.3 to 1.2 s using the image method (Allen and Berkley, 1979), as similarly done in Wang and Wang (2020a). For each RIR generation, we randomly sample a room length and width from [5, 10] m, and height from [3, 4] m. The microphone is placed at the height of [0, 1] m, and within 0.5 m of the center of the room. The target speaker is placed at the same height as the microphone, its distance from the microphone is within [0.75, 2.5] m, and it is at least 0.5 m from each wall. We split RIRs and assign 10k files to training, 1k files for validation, and 1k files for testing. For background noises, we randomly select 22 000 noises from the DNS challenge[2] and split them into 20k/1k/1k for training/validation/testing purposes. A noise segment is produced by randomly picking a noise file and cutting a segment of the same length as the speech utterance. If the noise file is not long enough, we extend its length by concatenating with another noise file until reaching sufficient length. For noise mixing in training and validation, we uniformly sample an SNR value from $\{-5, -4, \ldots, 5\}$ dB, which is calculated with respect to a reverberant speech signal and a noisy signal.

[1] https://github.com/ehabets/RIR-Generator.

[2] https://github.com/microsoft/DNS-Challenge.

Table 1
Enhancement (dereverberation) performance of two backbone DNNs.

Model	GFLOPs	Win (ms)	SI-SNR	STOI	PESQ
Unprocessed			3.920	0.830	2.324
ARN-WM	11.838	32	**11.978**	**0.962**	**3.494**
ARN-WM	11.707	16	11.372	0.959	3.465
ARN-WM	11.528	8	11.738	0.956	3.454
ARN-CSM	11.841	32	**11.967**	**0.961**	**3.498**
ARN-CSM	11.709	16	11.604	0.960	3.478
ARN-CSM	11.528	8	11.001	0.949	3.367
DC-CRN-WM	24.303	32	**9.662**	**0.944**	**3.257**
DC-CRN-WM	24.283	16	9.454	0.943	3.247
DC-CRN-WM	24.273	8	9.376	0.941	3.229
DC-CRN-CSM	24.376	32	**9.810**	**0.945**	**3.290**
DC-CRN-CSM	24.356	16	9.735	0.944	3.285
DC-CRN-CSM	24.345	8	9.572	0.943	3.242

4.2. Setup

For all experiments, speech utterances are sampled at 16 kHz. We apply root-mean squared (RMS) normalization to each reverberant/reverberant-noisy utterance, and scale the corresponding target utterance accordingly. For ARN, we use an Adam optimizer (Kingma and Ba, 2015) and train with a batch size of 32 utterances for 100 epochs. The initial learning rate is set to 0.0006, and is halved if the validation loss has not improved for three consecutive epochs. We also employ gradient clipping with a maximum value of 5.0 to avoid gradient explosion. The training procedure of the DC-CRN model is similar, except that we train with a batch size of 16 to satisfy the GPU memory constraint. Within each batch, we randomly cut 4 s of each utterance. Shorter utterances in a batch are padded with zeros so that all input features have the same size. During loss calculations, zero-padded regions are ignored. All models are trained using the automatic mixed precision to expedite training (Micikevicius et al., 2018). Two NVIDIA Volta V100 32 GB GPUs are utilized for training, and the batch is evenly distributed to two GPUs using the DataParallel module from PyTorch (Paszke et al., 2019) for each training step.

We evaluate enhancement performance using three metrics, scale-invariant SNR (SI-SNR), short-time objective intelligibility (STOI) (Taal et al., 2011) and perceptual evaluation of speech quality (PESQ) (Rix et al., 2001), which are standard objective evaluation metrics in speech enhancement studies. STOI has a typical value range of [0, 1], and we report the narrow-band PESQ that is based on the ITU P.862.1 standard (Recommendation, 2001), which ranges from [−0.5, 4.5]. For all these metrics, higher values indicate better performance.

4.3. Causality

We implement both causal and non-causal versions of two backbone DNNs, in part to facilitate comparisons with other baseline algorithms. The default settings are non-causal. For causal implementation, we use uni-directional LSTMs in ARN, and apply a mask to the attention matrix in an attention block such that the entries above the main diagonal are negative infinity to remove the contributions of future frames. For DC-CRN, the temporal dependencies are modeled by its RNN bottleneck. We easily convert DC-CRN to the causal version by converting grouped BLSTMs to grouped uni-directional LSTMs. We set the group number to 2 for the causal version and 4 for the non-causal version, so that their numbers of trainable parameters are close.

5. Experimental results

5.1. Reverberant-only experiments

To study factors impacting dereverberation performance, we conduct a series of experiments to enhance reverberant-only speech signals or dereverberate the speech signals. We fix the window shift to 2 ms, and experiment with window sizes of 32 ms, 16 ms, and 8 ms. We set the embedding dimension to be the same for all window sizes for a fair comparison, which is 1024 for ARN and 161 for DC-CRN as described in the respective papers. We also document giga floating-point operations per second (GFLOPs) by taking the average of enhancing 50 test utterances using an open-source package.[3] While ARN is computationally more efficient than DC-CRN, within each architecture, the computational complexities are close. The results are displayed in Table 1. We denote time domain models with **WM** standing for waveform mapping, and frequency domain models with **CSM** for complex spectral mapping. Overall, the ARN architecture consistently outperforms DC-CRN across all three metrics, two different representation domains, and three window sizes. We observe that a larger window size yields a small but consistent benefit for speech dereverberation, and the baselines with a 32 ms window size perform the best. This is to be expected, especially for longer T_{60} values. With shorter STFT

[3] https://github.com/Lyken17/pytorch-OpCounter.

Table 2
Enhancement performance of two backbone DNNs on reverberant-noisy DNS speech.

Models	Win (ms)	−5 dB			0 dB			5 dB		
		SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ
Unprocessed		−7.236	0.579	1.326	−3.002	0.652	1.558	0.276	0.714	1.780
ARN-WM	32	7.250	0.831	2.602	8.492	0.878	2.843	9.377	0.907	3.017
ARN-WM	16	**7.576**	**0.836**	**2.619**	**8.841**	**0.883**	**2.863**	**9.726**	**0.912**	**3.034**
ARN-WM	8	7.399	0.832	2.593	8.683	0.880	2.841	9.569	0.909	3.015
ARN-CSM	32	7.604	0.839	2.647	8.858	0.884	2.883	9.744	0.912	3.053
ARN-CSM	16	**7.704**	**0.843**	**2.659**	**8.973**	**0.888**	**2.897**	**9.865**	**0.915**	**3.064**
ARN-CSM	8	7.382	0.836	2.619	8.634	0.882	2.859	9.513	0.911	3.031
DC-CRN-WM	32	**5.784**	**0.779**	**2.331**	**6.953**	**0.839**	**2.588**	**7.800**	**0.877**	**2.776**
DC-CRN-WM	16	5.523	0.778	2.335	6.694	0.837	2.587	7.539	0.875	2.771
DC-CRN-WM	8	5.490	0.773	2.302	6.689	0.834	2.566	7.532	0.873	2.757
DC-CRN-CSM	32	**5.915**	**0.794**	**2.416**	**7.096**	**0.850**	**2.667**	**7.941**	**0.885**	**2.843**
DC-CRN-CSM	16	5.491	0.780	2.327	6.683	0.839	2.586	7.528	0.877	2.772
DC-CRN-CSM	8	5.134	0.764	2.234	6.295	0.827	2.505	7.122	0.867	2.700

Table 3
Enhancement performance comparison of strong baselines and proposed ARN-CSM in reverberant-noisy conditions. Win len/win shift denotes window size and window shift respectively.

	Win len/	−5 dB			0 dB			5 dB		
	win shift (ms)	SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ
Unprocessed		−7.236	0.579	1.326	−3.002	0.652	1.558	0.276	0.714	1.780
ARN-CSM (Pandey and Wang, 2022)	16/2	**7.091**	**0.811**	**2.446**	**8.401**	**2.865**	**2.720**	**9.322**	**0.899**	**2.913**
DC-CRN (Tan et al., 2021)	20/10	5.907	0.757	2.139	7.212	0.823	2.437	8.152	0.866	2.654
TFAUNet (Zhao et al., 2020)	20/10	4.826	0.679	1.811	6.034	0.753	2.042	6.955	0.806	2.225
SDDNet (Li et al., 2021)	20/10	5.438	0.745	2.101	6.802	0.810	2.353	7.758	0.854	2.547
TRUNet (Choi et al., 2021)	16/4	5.104	0.729	2.057	6.347	0.795	2.310	7.215	0.842	2.504
UFormer (Fu et al., 2022)	25/10	4.803	0.717	2.021	6.508	0.796	2.322	7.431	0.843	2.522

analysis windows, reverberation modeling within each frame is less accurate, causing worse estimation. The same reasoning applies to time-domain models. Another finding is that STFT features have a little better performance compared with waveform segments for DC-CRN. With a large enough window size, time-domain models perform competitively with frequency-domain counterparts. For example, with a 32 ms window size, ARN-WM has almost the same performance as ARN-CSM, especially for longer T_{60} values.

5.2. Reverberant-noisy experiments

We employ the two backbone DNNs to perform non-causal speech enhancement in reverberant-noisy conditions. All the models are trained from scratch. Similar to the reverberation setting, we control the window shift to 2 ms. The test data is created by first convolving test utterances with test RIRs, and then mixing them with test noises at three SNR levels: −5 dB, 0 dB and 5 dB. We display enhancement results in Table 2. Overall, the ARN architecture still achieves better enhancement performance. Compared with the best result obtained by DC-CRN, we observe over 1 dB SI-SNR improvement, over 3% higher in STOI and 0.2 in PESQ. Window size has a noticeable impact on enhancement performance. For DC-CRN, large window sizes are always beneficial in terms of SI-SNR, STOI, and PESQ. When the window size is 8 ms, we observe a noticeable drop in enhancement performance, especially in PESQ. The drop is significant for complex spectral mapping: for DC-CRN-CSM at −5 dB SNR, the window size of 8 ms has a 0.03 STOI drop and 0.18 PESQ degradation compared to the window size configuration of 32 ms. In terms of ARN, the 16 ms window size produces the best enhancement performance. This could result from a compromise between denoising and dereverberation, as removing background noise would not require a large window size. The effect of feature domains is consistent with the finding in Section 5.1: there is a slight advantage of CSM over WM, but the overall performance is close. Additionally, for both ARN and DC-CRN, the scores of waveform mapping variants are less affected by the window size compared with CSM counterparts. This could be because the time difference between neighboring waveform samples is fixed, while the frequency spacing changes, as window size varies. In other words, window size does not impact the time resolution of WM, but alters the frequency resolution of CSM.

5.3. Comparison with other baselines

To demonstrate the effectiveness of the proposed architectures, we compare different models under the same experimental setting in 5.2 for speech enhancement in reverberant-noisy conditions. For our ARN-CSM baseline, we employ a window shift of 2 ms and the window size of 16 ms. The DC-CRN implementation follows the original configuration in Tan et al. (2021), where we do not use linear layers to transform input features and adopt a window size of 20 ms with a 50% overlap. We compare with several strong baselines introduced in Section 1. All baselines are implemented in the causal settings for fair comparison.

Table 4
Enhancement performance comparison of different training targets.

	−5 dB			0 dB			5 dB		
	SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ
Unprocessed	−7.236	0.579	1.326	−3.002	0.652	1.558	0.276	0.714	1.780
Early reverberation	**7.604**	**0.839**	**2.647**	**8.858**	**0.884**	**2.883**	**9.744**	**0.912**	**3.053**
Direct-path	7.415	0.835	2.638	8.540	0.884	2.880	9.750	0.909	3.049
Dry clean	6.897	0.821	2.426	7.995	0.872	2.674	9.235	0.891	2.968

Table 5
Enhancement performance comparison of different loss functions.

	−5 dB			0 dB			5 dB		
	SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ	SI-SNR	STOI	PESQ
Unprocessed	−7.236	0.579	1.326	−3.002	0.652	1.558	0.276	0.714	1.780
PCM	7.604	**0.839**	2.647	8.858	**0.884**	2.883	9.744	**0.912**	3.053
RI+MAG	7.951	0.838	**2.649**	9.239	**0.884**	**2.890**	10.140	**0.912**	**3.062**
MAE	**8.191**	0.820	2.507	**9.489**	0.869	2.732	**10.387**	0.898	2.904
SI-SNR	7.864	0.811	2.470	8.995	0.861	2.696	9.892	0.893	2.864

The first baseline is TFAUNet (time–frequency attention UNet) by Zhao and Wang (2020). We convert the original implementation to a causal setting by eliminating future frames in attention computation, and also modify dilated convolutions to fit the causal requirement. The second baseline is SDDNet (Li et al., 2021), and we follow the original description by using a 320-point STFT with a 50% overlap and also compress spectral magnitudes. Additionally, we adopt the three-stage implementation as it leads to the best objective metrics. The third model for comparison is the tiny recurrent UNet by Choi et al. (2021), denoted as TRUNet. Following their description, we use a 16 ms window length and 4 ms window shift. Our implementation uses an open-source repository[4] as reference. The last baseline is UFormer (Fu et al., 2022). Following their default setting in the publicly available code repository,[5] we adopt a window size of 25 ms and a 10 ms window shift for a 512-point STFT computation.

We present comparison results in Table 3. Our ARN-CSM shows a clear advantage over other baselines in all metrics. Especially at −5 dB SNR, ARN-CSM has 1.184 dB SI-SNR improvement over the second best baseline of DC-CRN, and STOI reaches 0.811, which is 6.6% higher than SDDNet. In addition, the PESQ value of 2.446 is much higher than the next two best baselines of DC-CRN and SDDNet, demonstrating its strong ability in speech dereverberation and denoising.

5.4. Training targets

In this study, we adopt a training target obtained by convolving dry clean signals with the first 50 ms of a simulated RIR, denoted as "early reverberation". In addition, we conduct a set of enhancement experiments with different training targets using ARN-CSM for joint denoising and dereverberation, and present the results in Table 4. This evaluation includes the direct-path signal, obtained by first setting the T_{60} parameter of the RIR generator to zero and then convolving with the dry clean signal (Wang et al., 2021a), and the dry clean signal. As shown in the table, training with the proposed early-reverberation target produces slightly better results than the direct-path signal. Training with the dry-clean signal consistently performs worse across different SNR levels. Using the dry-clean signal as the target increases training difficulty and may lead to artifacts in the enhanced speech (Valin et al., 2022).

5.5. Loss functions

To examine the effect of loss functions, we perform speech enhancement using the ARN-CSM variant in reverberant-noisy conditions. The results of the loss functions described in Section 3.2 are shown in Table 5. We find that the frequency-domain loss functions (PCM and RI+MAG) show a clear advantage over the time-domain losses (MAE and SI-SNR) in STOI and PESQ. This advantage could be explained as follows. In the time domain, reverberation is produced by convolving a speech signal with an RIR, which is non-additive, and room impulse responses are highly sensitive. In the frequency domain, reverberant speech amounts to the product of the frequency response of an RIR and a complex spectrogram,[6] which should be easier to separate. Also, both PCM and RI+MAG regulate STFT magnitude estimation, which benefits STOI and PESQ metrics that are measured in magnitudes. PCM and RI+MAG perform similarly in terms of STOI and PESQ, although RI+MAG has a small improvement over PCM in SI-SNR. Among time-domain losses, MAE performs slightly better than SI-SNR, which is very sensitive to sudden changes of speech samples.

[4] https://github.com/YangangCao/TRUNet.
[5] https://github.com/felixfuyihui/Uformer.
[6] Strictly speaking the statement is only valid when the STFT analysis window is greater than the length of the RIR filter, and it is otherwise an approximation.

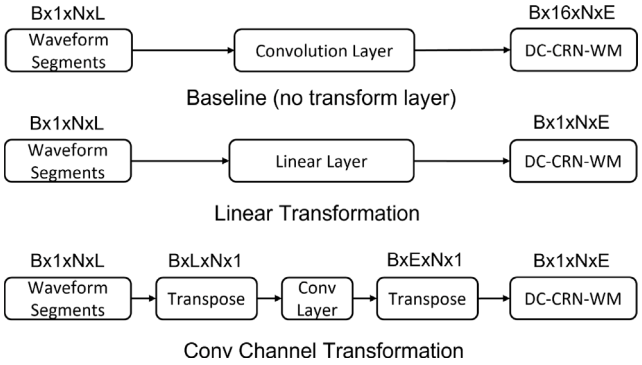


Fig. 3. Illustration of the baseline, and two transforms performed in DC-CRN-WM corresponding to the "Linear Transformation" and "Conv Channel Transformation". Also shown are feature dimensions in the format of $(BatchSize, ConvolutionChannels, NumOfFrames, FrameLength)$ with L denoting frame length and E embedding dimension.

Table 6
The effect of transform layers on DC-CRN-WM for speech dereverberation.

Model	SI-SNR	STOI	PESQ
Unprocessed	3.920	0.830	2.324
Baseline (no transform layer)	7.865	0.920	3.038
Linear transformation	9.830	**0.939**	3.169
Conv channel transformation	**9.880**	**0.939**	**3.197**

5.6. Transform layer

Previous studies suggest that the initial layers of CNN-based time-domain networks can emulate STFT-like transformations (Luo and Mesgarani, 2018; Borgström and Brandstein, 2020). In this section, we investigate and evaluate the effect of different transforms. We argue that the incorporation of a transform layer is crucial for time-domain networks for speech dereverberation, and is a major factor contributing to their competitive performance compared with frequency-domain counterparts. A transform layer in this context refers to a learnable layer that operates on input vectors, which works like STFT operations. It is typically a linear layer, but it can be other learnable operations like convolution. To validate this argument, we employ a causal DC-CRN-WM network as our baseline and compare the speech dereverberation results of different transforms. For the baseline network, instead of a linear encoder, we add a convolutional layer with a stride of (1,2) and a kernel size of (1,4), corresponding to the number of frames and frame length, respectively. The number of channels is set to 16, and the subsequent convolution layer is changed accordingly. Unlike traditional approaches that use a linear encoder, our baseline integrates a convolutional layer with a stride of (1,2) and a kernel size of (1,4), corresponding to the number of frames and frame length, respectively. We regard the frame length as the feature dimension. The added convolution operation is applied locally and does not project the feature dimension to the embedding dimension. This setup leads to a fully convolutional baseline without any linear pre- or post-processing layers, distinguishing it from models with transform layers. Fig. 3 illustrates this baseline and two kinds of transform layer. The first transform layer (Linear Transform) is the proposed design, where we perform linear transformation before the convolutional encoders on the feature dimension directly. The second transform layer (Conv Channel) employs a convolution layer to transform the feature dimension into the embedding dimension, which treats the feature dimension as convolutional channels and serves as the transform layer.

The speech dereverberation results are listed in Table 6. As shown in the table, the baseline with no transform layer clearly underperforms the others. Compared with the two transforms, STOI drops by 1.9%, and PESQ by over 0.13. Using a linear layer or a convolution operation to encode features to an embedding space results in similar dereverberation performance. Fig. 4 visualizes the magnitude spectrograms of a target utterance and the corresponding reverberant utterance, along with the learned representations obtained by the baseline and two transforms (obtained before the backbone network in Fig. 1). Note that we only plot the first channel for the baseline, as visualizations of other channels are similar. The learnable representations of "Linear Transform" and "Conv Channel" show similar patterns, which resemble globally the magnitude spectrogram of the target utterance, hence beneficial for dereverberation. Without a transform layer, the convolutional encoder operates locally and cannot capture the global pattern of reverberation.

6. Conclusion

We have investigated various techniques and designs for speech dereverberation and speech enhancement in reverberant-noisy conditions. Using two backbone DNNs, we contrast time-domain and frequency-domain models. Our investigation has led to the following observations and insights. A proper transform layer is crucial for speech dereverberation of time-domain models, as it

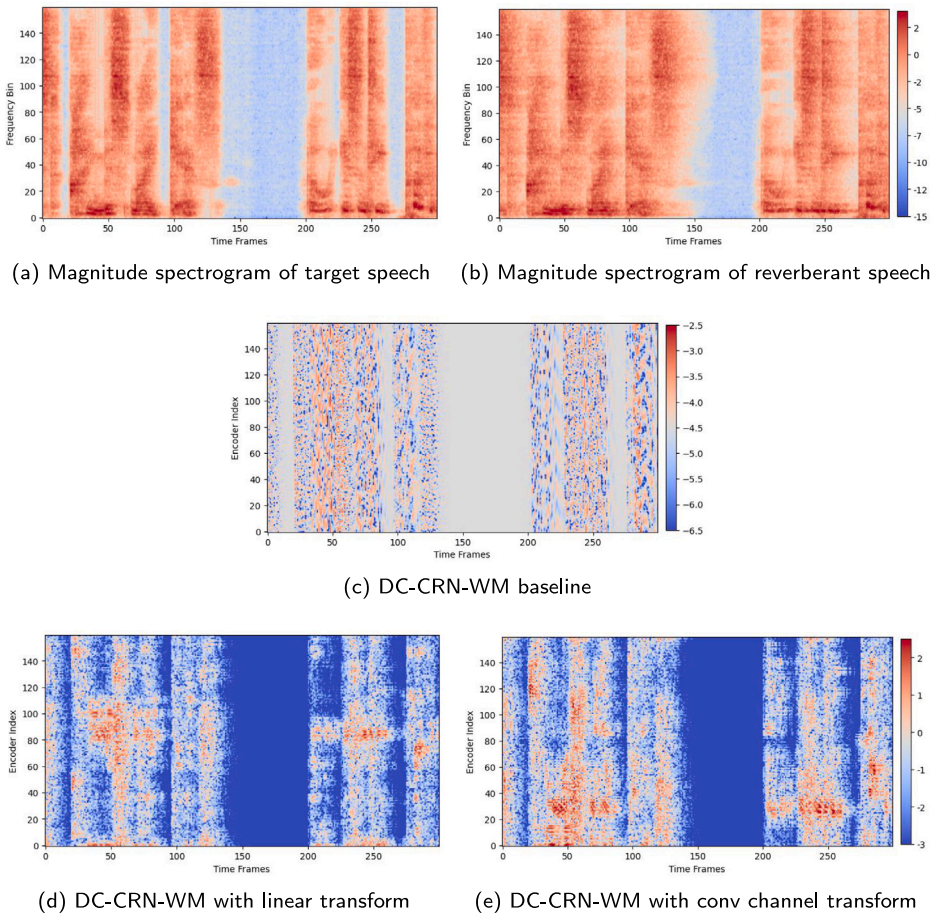


Fig. 4. Effects of transform layers: (a). The magnitude spectrogram of a target utterance. (b). The magnitude spectrogram of the corresponding reverberant utterance. (c). The visualization of the encoded output obtained for the fully convolutional baseline. (d). Visualization of the encoded output by performing convolutional transform on the feature dimension. (e). Visualization of the encoded output by performing a linear transform on the feature dimension.

enables learning meaningful and sparse representations that are similar to STFT magnitudes. Although STFT features show a slight advantage over waveform signals, with a large enough window size and a simple linear transform layer, one can obtain equally strong dereverberation performance using time-domain models. We also show that window sizes play a significant role in enhancement models, and larger window sizes tend to result in better dereverberation performance. Using an optimized loss function and the above insights, a frequency-domain ARN outperforms other strong baselines and achieves state-of-the-art enhancement performance on the DNS dataset.

CRediT authorship contribution statement

Heming Wang: Writing – original draft. **Ashutosh Pandey:** Writing – review & editing. **DeLiang Wang:** Writing – review & editing, Supervision.

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

Data will be made available on request.

Acknowledgments

The authors would like to thank Yihui Fu, Andong Li, Ke Tan and Yan Zhao for and assistance in model comparisons. This research was supported in part by an NIDCD (R01 DC012048) grant, the Ohio Supercomputer Center, and the Pittsburgh Supercomputing Center (under NSF grant ACI-1928147).

References

Al-Karawi, K.A., Al-Noori, A.H., Li, F.F., Ritchings, T., et al., 2015. Automatic speaker recognition system in adverse conditions—implication of noise and reverberation on system performance. Int. J. Inf. Electron. Eng. 5, 423–427.
Allen, J., Berkley, D., 1979. Image method for efficiently simulating small-room acoustics. J. Acoust. Soc. Am. 65, 943–950.
Aralikatti, R., Ratnarajah, A., Tang, Z., Manocha, D., 2021. Improving reverberant speech separation with synthetic room impulse responses. In: Proceedings of ASRU. pp. 900–906.
Ba, J.L., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. arXiv:1607.06450.
Bai, Z., Zhang, X.-L., 2021. Speaker recognition based on deep learning: An overview. Neural Netw. 140, 65–99.
Borgström, B.J., Brandstein, M.S., 2020. Speech enhancement via attention masking network (SEAMNET): An end-to-end system for joint suppression of noise and reverberation. IEEE/ACM Trans. Audio Speech Lang. Process. 29, 515–526.
Braun, S., Kuklasiński, A., Schwartz, O., Thiergart, O., Habets, E.A., Gannot, S., Doclo, S., Jensen, J., 2018. Evaluation and comparison of late reverberation power spectral density estimators. IEEE/ACM Trans. Audio Speech Lang. Process. 26, 1056–1071.
Choi, H.-S., Park, S., Lee, J.H., Heo, H., Jeon, D., Lee, K., 2021. Real-time denoising and dereverberation with tiny recurrent U-Net. In: Proceedings of ICASSP. pp. 5789–5793.
Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In: Proceedings of ICLR.
Cord-Landwehr, T., Boeddeker, C., Von Neumann, T., Zorilă, C., Doddipatla, R., Haeb-Umbach, R., 2022. Monaural source separation: From anechoic to reverberant environments. In: Proceedings of IWAENC. p. 5.
Defossez, A., Synnaeve, G., Adi, Y., 2020. Real time speech enhancement in the waveform domain. In: Proceedings of INTERSPEECH. pp. 545–549.
Delcroix, M., Ochiai, T., Zmolikova, K., Kinoshita, K., Tawara, N., Nakatani, T., Araki, S., 2020. Improving speaker discrimination of target speech extraction with time-domain speakerbeam. In: Proceedings of ICASSP. pp. 691–695.
Fu, Y., Liu, Y., Li, J., Luo, D., Lv, S., Jv, Y., Xie, L., 2022. UFormer: A UNet based dilated complex & real dual-path conformer network for simultaneous speech enhancement and dereverberation. In: Proceedings of ICASSP. pp. 7417–7421.
Galassi, A., Lippi, M., Torroni, P., 2020. Attention in natural language processing. IEEE Trans. Neural Netw. Learn. Syst. 32, 4291–4308.
Gao, F., Wu, L., Zhao, L., Qin, T., Cheng, X., Liu, T.-Y., 2018. Efficient sequence learning with group recurrent networks. In: Proceedings of NAACL. pp. 799–808.
Giri, R., Isik, U., Krishnaswamy, A., 2019. Attention Wave-U-Net for speech enhancement. In: Proceedings of WASPAA. pp. 249–253.
Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R., Cheng, M.-M., Hu, S.-M., 2022. Attention mechanisms in computer vision: A survey. Comput. Vis. Media 8, 1–38.
Habets, E., Gannot, S., Cohen, I., 2009. Late reverberant spectral variance estimation based on a statistical model. IEEE Signal Process. Lett. 16, 770–773.
Han, K., Wang, Y., Wang, D.L., Woods, W.S., Merks, I., Zhang, T., 2015. Learning spectral mapping for speech dereverberation and denoising. IEEE/ACM Trans. Audio Speech Lang. Process. 23, 982–992.
Hendrycks, D., Gimpel, K., 2016. Gaussian error linear units (GELUs). arXiv:1606.08415.
Heymann, J., Drude, L., Haeb-Umbach, R., Kinoshita, K., Nakatani, T., 2019. Joint optimization of neural network-based WPE dereverberation and acoustic model for robust online ASR. In: Proceedings of ICASSP. pp. 6655–6659.
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. In: Proceedings of INTERSPEECH. pp. 2482–2486.
Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of CVPR. pp. 4700–4708.
Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of ICML. pp. 448–456.
Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: Proceedings of ICLR.
Kinoshita, K., Delcroix, M., Nakatani, T., Miyoshi, M., 2007. Multi-step linear prediction based speech dereverberation in noisy reverberant environment. In: Proceedings of INTERSPEECH. pp. 854–857.
Kinoshita, K., Ochiai, T., Delcroix, M., Nakatani, T., 2020. Improving noise robust automatic speech recognition with single-channel time-domain enhancement network. In: Proceedings of ICASSP. IEEE, pp. 7009–7013.
Le Roux, J., Wisdom, S., Erdogan, H., Hershey, J.R., 2019. SDR–half-baked or well done? In: Proceedings of ICASSP. pp. 626–630.
Li, A., Liu, W., Luo, X., Yu, G., Zheng, C., Li, X., 2021. A simultaneous denoising and dereverberation framework with target decoupling. In: Proceedings of INTERSPEECH. pp. 2801–2805.
Li, A., Yu, G., Zheng, C., Liu, W., Li, X., 2023. A general unfolding speech enhancement method motivated by Taylor's theorem. IEEE/ACM Trans. Audio Speech Lang. Process. 32, 3629–3646.
Luo, Y., Mesgarani, N., 2018. TasNet: time-domain audio separation network for real-time, single-channel speech separation. In: Proceedings of ICASSP. pp. 696–700.
Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al., 2018. Mixed precision training. In: Proceedings of ICLR.
Nakatani, T., Yoshioka, T., Kinoshita, K., Miyoshi, M., Juang, B.-H., 2010. Speech dereverberation based on variance-normalized delayed linear prediction. IEEE Trans. Audio Speech Lang. Process. 18, 1717–1731.
Neri, J., Braun, S., 2023. Towards real-time single-channel speech separation in noisy and reverberant environments. In: Processings of ICASSP. p. 5.
Pandey, A., Wang, D.L., 2020. Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain. In: Proceedings of ICASSP. pp. 6629–6633.
Pandey, A., Wang, D.L., 2021. Dense CNN with self-attention for time-domain speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. 29, 1270–1279.
Pandey, A., Wang, D.L., 2022. Self-attending RNN for speech enhancement to improve cross-corpus generalization. IEEE/ACM Trans. Audio Speech Lang. Process. 30, 1374–1385.
Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. PyTorch: An imperative style, high-performance deep learning library. In: Proceedings of NeuralIPS. pp. 8026–8047.
Povey, D., Hadian, H., Ghahremani, P., Li, K., Khudanpur, S., 2018. A time-restricted self-attention layer for ASR. In: Proceedings of ICASSP. pp. 5874–5878.
Purushothaman, A., Dutta, D., Kumar, R., Ganapathy, S., 2023. Speech dereverberation with frequency domain autoregressive modeling. IEEE/ACM Trans. Audio Speech Lang. Process. 32, 29–38.
Ravenscroft, W., Goetze, S., Hain, T., 2023. Deformable temporal convolutional networks for monaural noisy reverberant speech separation. In: Processings of ICASSP. p. 5.

Recommendation, I.-T., 2001. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. Rec. ITU-T P. 862.
Reddy, C.K., Dubey, H., Gopal, V., Cutler, R., Braun, S., Gamper, H., Aichner, R., Srinivasan, S., 2021. ICASSP 2021 deep noise suppression challenge. In: Proceesings of ICASSP. pp. 6623–6627.
Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P., 2001. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In: Proceedings of ICASSP. pp. 749–752.
Salvati, D., Drioli, C., Foresti, G.L., 2023. A late fusion deep neural network for robust speaker identification using raw waveforms and gammatone cepstral coefficients. Expert Syst. Appl. 222, 119750.
Schroter, H., Escalante-B, A.N., Rosenkranz, T., Maier, A., 2022. DeepFilterNet: A low complexity speech enhancement framework for full-band audio based on deep filtering. In: Proceedings of ICASSP. pp. 7407–7411.
Shaw, P., Uszkoreit, J., Vaswani, A., 2018. Self-attention with relative position representations. In: Proceedings of NAACL. pp. 464–468.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.
Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., Zhong, J., 2021. Attention is all you need in speech separation. In: Proceedings of ICASSP. pp. 21–25.
Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J., 2011. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. IEEE/ACM Trans. Audio Speech Lang. Process. 19, 2125–2136.
Talmon, R., Cohen, I., Gannot, S., 2009. Relative transfer function identification using convolutive transfer function approximation. IEEE Trans. Audio Speech Lang. Process. 17, 546–555.
Tan, K., Wang, D.L., 2018. A convolutional recurrent neural network for real-time speech enhancement. In: Proceedings of INTERSPEECH. Vol. 2018, pp. 3229–3233.
Tan, K., Zhang, X., Wang, D.L., 2021. Deep learning based real-time speech enhancement for dual-microphone mobile phones. IEEE/ACM Trans. Audio Speech Lang. Process. 29, 1853–1863.
Valin, J.-M., Giri, R., Venkataramani, S., Isik, U., Krishnaswamy, A., 2022. To dereverb or not to dereverb? Perceptual studies on real-time dereverberation targets. arXiv:2206.07917.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Proceedings of NeuralIPS. pp. 6000–6010.
Wang, D.L., Chen, J., 2018. Supervised speech separation based on deep learning: An overview. IEEE/ACM Trans. Audio Speech Lang. Process. 26, 1702–1726.
Wang, Z.-Q., Wang, D.L., 2020a. Deep learning based target cancellation for speech dereverberation. IEEE/ACM Trans. Audio Speech Lang. Process. 28, 941–950.
Wang, Z.-Q., Wang, D.L., 2020b. Multi-microphone complex spectral mapping for speech dereverberation. In: Proceedings of ICASSP. pp. 486–490.
Wang, Z.-Q., Wang, P., Wang, D.L., 2020. Complex spectral mapping for single-and multi-channel speech enhancement and robust ASR. IEEE/ACM Trans. Audio Speech Lang. Process. 28, 1778–1787.
Wang, Z.-Q., Wichern, G., Le Roux, J., 2021a. Convolutive prediction for monaural speech dereverberation and noisy-reverberant speaker separation. IEEE/ACM Trans. Audio Speech Lang. Process. 29, 3476–3490.
Wang, Z.-Q., Wichern, G., Roux, J.L., 2021b. On the compensation between magnitude and phase in speech separation. IEEE Signal Process. Lett. 28, 2018–2022.
Zhang, J., Plumbley, M.D., Wang, W., 2021. Weighted magnitude-phase loss for speech dereverberation. In: Proceedings of ICASSP. pp. 5794–5798.
Zhao, Y., Wang, D.L., 2020. Noisy-reverberant speech enhancement using DenseUNet with time-frequency attention. In: Proceedings of INTERSPEECH. pp. 3261–3265.
Zhao, X., Wang, Y., Wang, D.L., 2014. Robust speaker identification in noisy and reverberant conditions. IEEE/ACM Trans. Audio Speech Lang. Process. 22, 836–845.
Zhao, Y., Wang, Z.-Q., Wang, D.L., 2018. Two-stage deep learning for noisy-reverberant speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. 27, 53–62.
Zhao, Y., Wang, D., Xu, B., Zhang, T., 2020. Monaural speech dereverberation using temporal convolutional networks with self attention. IEEE/ACM Trans. Audio Speech Lang. Process. 28, 1598–1607.