INCORPORATING INTRA-SPECTRAL DEPENDENCIES WITH A RECURRENT OUTPUT LAYER FOR IMPROVED SPEECH ENHANCEMENT

Khandokar Md. Nayem and Donald S. Williamson

Department of Computer Science, Indiana University, USA
knayem@iu.edu, *williads@indiana.edu*

ABSTRACT

Deep-learning based speech enhancement systems have offered tremendous gains, where the best performing approaches use long short-term memory (LSTM) recurrent neural networks (RNNs) to model temporal speech correlations. These models, however, do not consider the frequency-level correlations within a single time frame, as spectral dependencies along the frequency axis are often ignored. This results in inaccurate frequency responses that negatively affect perceptual quality and intelligibility. We propose a deep-learning approach that considers temporal and frequency-level dependencies. More specifically, we enforce spectral-level dependencies within each spectral time frame through the introduction of a recurrent output layer that models the Markovian assumption along the frequency axis. We evaluate our approach in a variety of speech and noise environments, and objectively show that this recurrent spectral layer offers performance gains over traditional approaches. We also show that our approach outperforms recent approaches that consider frequency-level dependencies.

Index Terms— speech enhancement, intra-spectral correlations, recurrent neural networks, long short-term memory

1. INTRODUCTION

Speech enhancement, which strives to effectively remove unwanted background noise, is an important problem for several applications, including voice-based home assistants (e.g. Google Home and Amazon Echo), hearing aids, and many military applications. The performance of these devices and applications severely degrades when noise is present, as noise makes it difficult to understand speech, largely due to spectral and temporal masking effects that render the speech inaudible. Humans can recognize speech in extreme noisy environments, but this is not the case for the above systems. Thus, it is important that speech enhancement continue to improve to increase the usability of these devices.

Deep learning is advancing the field of speech enhancement, where a wide range of architectures have been used to address this problem. Approaches have been developed using various networks, including, deep neural networks (DNNs) [1, 2, 3, 4], autoencoders [5, 6], long short-term memory (LSTM) networks [3, 7, 8], and convolutional neural networks (CNNs) [9]. Speech enhancement generally takes the form of either time-frequency (T-F) masking or signal approximation. Masking-based approaches estimate T-F filters that generate clean-speech estimates by applying a filter to the noisy speech. In [1], it is shown that a ratio masking approach outperforms other T-F masking targets, where a DNN is used to estimate the ideal ratio mask (IRM). A ratio mask that uses phase information (e.g. phase-sensitive mask (PSM)) is proposed in [8]. This approach uses a phase-sensitive cost function and a LSTM recurrent neural network (RNN). More recent approaches use deep clustering (DC), which groups learned activations into classes (e.g. speech dominant or noise dominant), to form a binary mask (BM) [10]. The BM retains T-F units that are speech dominant. DC approaches use LSTM networks to capture inter-temporal speech correlations. Signal-based approaches, which directly estimate the speech magnitude response, have also been proposed recently, where variants of LSTM, DNN, and RNN networks are used [2, 5, 3, 4].

All the above approaches produce T-F outputs that are based on prior network layers and prior (in time) outputs of that T-F unit. In other words, the spectral output at a particular time-frequency point is not based on the spectral output at adjacent or nearby frequency points. This is problematic as it is known that speech has spectral dependencies along the frequency axis [11, 12, 13]. The above approaches only address spectral correlations across time. Two recent approaches have been developed to address frequency-level dependencies, but they have only been evaluated for automatic speech recognition [14] or audio restoration after coding [15]. Both approaches use dedicated LSTM modules to learn spectral dependencies, but this is either done at the sub-band frequency-level or over all time. Additionally, these approaches do not consider local spectral dependencies over short-time instances. Nevertheless, these approaches have shown that incorporating spectral dependencies offers noticeable improvements, but it is not clear if this will have the same impact for speech enhancement.

978-1-7281-0824-7/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: Indiana University. Downloaded on June 10,2020 at 13:51:16 UTC from IEEE Xplore. Restrictions apply.

In this paper, we propose an intra-spectral (e.g. across-frequency) recurrent layer that captures frequency dependencies within each time frame of a speech signal. Given a noisy speech input, multiple LSTM layers first capture the temporal dependencies of speech. We then append the proposed intra-spectral recurrent layer to enforce spectral-level dependencies. The entire network is trained to estimate the log-magnitude spectrum of clean speech. Other spectral-dependent approaches [14, 15] have been proposed previously, but our approach effectively captures the intra-spectral dependencies. To the best of our knowledge, intra-spectral dependencies have not been investigated for monaural speech separation.

The rest of the paper is organized as follows. Traditional deep learning-based speech enhancement is discussed in section 2. In section 3, we describe our proposed intra-spectral recurrent (ISR) and intra-spectral bi-direectional recurrent (ISBR) layers. A discussion of the experiments and results is provided in section 4. We conclude in section 5.

2. NEURAL NETWORK-BASED SPEECH ENHANCEMENT

Let's define s_t as a clean speech signal and n_t as unwanted background noise, both in the time domain. $\hat{s}_t$ is an estimate of the clean speech signal, which is an enhanced version of the noisy speech mixture, m_t (e.g. $m_t = s_t + n_t$). $S_{t,k}$ is the T-F domain signal at time t and frequency k, which is computed from s_t using the short-time Fourier transform (STFT). Correspondingly, $S_{t,k}$ has a magnitude response, $|S_{t,k}|$, and a phase response, $\theta_{S_{t,k}}$, where $S_{t,k} = |S_{t,k}|e^{i\theta_{S_{t,k}}}$. Most speech enhancement systems enhance the magnitude response of noisy speech, $|M_{t,k}|$, in order to produce an estimated version, $|\hat{S}_{t,k}|$. An estimate of the time-domain signal, $\hat{s}_t$ is produced by combining the enhanced magnitude response, with the phase response of the noisy speech. Note that we are not addressing phase enhancement, and will leave that for future work.

Signal-based speech enhancement uses a function, $F_\phi(\cdot)$, to learn a mapping between noisy and clean speech (e.g. $|\hat{S}_{t,k}| = F_\phi(|M_{t,k}|)$, where ϕ defines the system parameters. The mapping is determined by minimizing an objective function over all training examples. The objective function, which is typically the mean-square error (MSE), compares each samples estimated speech signal to the true clean speech (e.g. $\sum_{t,k}\left(|\hat{S}_{t,k}| - |S_{t,k}|\right)^2$).

2.1. Deep neural network (DNN) approach

DNNs map noisy speech to estimates of clean speech by processing the input through multiple layers of neurons. In this case, ϕ represents specific weight values and network configurations. Lets define $\boldsymbol{p}_t$ as the input vector to the DNN model (e.g. $\boldsymbol{p}_t = |M_{t,:}|$) and $\hat{y}_{t,k}$ as the predicted output (e.g. $\hat{y}_{t,k} = |\hat{S}_{t,k}|$) at a specific T-F unit. Here, : indicates that values across all frequency points are retained. Additionally, we define n^l as the number of neurons in the l^{th} layer, $\boldsymbol{V}^l \in \mathbb{R}^{n^l \times n^{l-1}}$ as the weight matrix in the l^{th} layer, and $\boldsymbol{z}^l \in \mathbb{R}^{n^l}$ as the bias vector in the l^{th} DNN layer. Note that $l \in [1, L]$. The neuron output at each layer is computed as defined in Eq. (1), where σ is the activation function and $\boldsymbol{a}_t^l$ is the n^l-dimensional vector of neuron outputs at the t^{th} time frame.

$$\boldsymbol{a}_t^l = \sigma(\boldsymbol{V}^l \boldsymbol{a}_t^{l-1} + \boldsymbol{z}^l) \tag{1}$$

The output is generated when $l = L$, so $\hat{\boldsymbol{y}}_t = \boldsymbol{a}_t^L$. Similarly, the input is defined when $l - 1 = 0$ (or $\boldsymbol{a}_t^0 = \boldsymbol{p}_t$), where $\boldsymbol{p}_t$ represents the short-time log-magnitude spectrum at time t, *i.e.* $\boldsymbol{p}_t = |M_{t,:}|$. Standard backpropagation is used to determine optimal values for the weight matrices, $\boldsymbol{V}^l$, and bias vectors, $\boldsymbol{z}^l$, at each layer.

It is clear from (1), that the output at a certain layer, only depends on the outputs from the prior layer. Additionally for the output layer, the spectral output at each neuron does not depend on spectral outputs from other output-layer neurons. The network can feasibly give uncorrelated (across time and frequency) outputs, which is undesired since speech is both spectrally and temporally correlated.

2.2. Long short-term memory (LSTM) approach

LSTM networks are a type of recurrent architecture that utilize short- and long-term temporal information to make temporally-correlated predictions. This is accomplished by considering current and previously observed data. More specifically, a LSTM generates outputs using the following calculations:

$$\boldsymbol{f}_t^l = \sigma_g(\boldsymbol{W}_f^l \boldsymbol{a}_t^{l-1} + \boldsymbol{U}_f^l \boldsymbol{h}_{t-1}^l + \boldsymbol{b}_f^l) \tag{2}$$

$$\boldsymbol{i}_t^l = \sigma_g(\boldsymbol{W}_i^l \boldsymbol{a}_t^{l-1} + \boldsymbol{U}_i^l \boldsymbol{h}_{t-1}^l + \boldsymbol{b}_i^l) \tag{3}$$

$$\boldsymbol{o}_t^l = \sigma_g(\boldsymbol{W}_o^l \boldsymbol{a}_t^{l-1} + \boldsymbol{U}_o^l \boldsymbol{h}_{t-1}^l + \boldsymbol{b}_o^l) \tag{4}$$

$$\boldsymbol{c}_t^l = \boldsymbol{f}_t^l \circ \boldsymbol{c}_{t-1}^l + \boldsymbol{i}_t^l \circ \sigma_c(\boldsymbol{W}_c^l \boldsymbol{a}_t^{l-1} + \boldsymbol{U}_c^l \boldsymbol{h}_{t-1}^l + \boldsymbol{b}_c^l) \tag{5}$$

$$\boldsymbol{h}_t^l = \boldsymbol{o}_t^l \circ \sigma_h(\boldsymbol{c}_t^l) \tag{6}$$

$$\boldsymbol{a}_t^l = \sigma_a(\boldsymbol{W}_a^l \boldsymbol{h}_t^l + \boldsymbol{b}_a^l), \quad l \in [1, L] \tag{7}$$

In the above equations, $\boldsymbol{f}_t^l$, $\boldsymbol{i}_t^l$, $\boldsymbol{o}_t^l$ represent the l^{th}-layer's activation vectors for the forget, input and output gates of a LSTM cell, at time t. $\boldsymbol{h}_t^l$ is the hidden state vector and $\boldsymbol{c}_t^l$ is the cell state vector of the l^{th}-layer. As before, $\boldsymbol{a}_t^l$ is the vector of neuron outputs for $l \in [1, L]$, where $\boldsymbol{a}_t^0 = \boldsymbol{p}_t$ is the input vector. Each LSTM layer has n^l LSTM units. Additionally, $\boldsymbol{W} \in \mathbb{R}^{n^l \times n^{l-1}}$, $\boldsymbol{U} \in \mathbb{R}^{n^l \times n^l}$, and $\boldsymbol{b} \in \mathbb{R}^{n^l}$ are the weight and bias matrices that are optimized during training. The subscripts on the weight matrices and bias vectors indicate the type of gate, cell or output. σ_g denotes the gate activation function, whereas σ_c and σ_h denote the activation

Authorized licensed use limited to: Indiana University. Downloaded on June 10,2020 at 13:51:16 UTC from IEEE Xplore. Restrictions apply.

functions for the cell and hidden states, respectively. The final network output is defined as $\hat{\boldsymbol{y}}_t = \boldsymbol{a}_t^L$.

The LSTM network unrolls along the time axis, which means relationships across time frames are learned by the model. This structure, however, does not learn spectral relationships along the frequency axis. An ideal model would perform inference along both time and frequency axes, because both contain correlations. Thus, we propose a model that captures both temporal and frequency-level dependencies.

3. PROPOSED APPROACH

We propose to capture intra-spectral correlations with a recurrent layer that uses a first-order Markov assumption. In other words, knowing that adjacent spectral components are dependent, we design a recurrent layer that functions as a Markov chain, where the spectral output at a certain frequency is provided as input to adjacent neurons. This is done along the entire frequency axis, and a depiction is shown in Fig. 1 (a). This recurrent layer is denoted as an intra-spectral recurrent (ISR) layer.

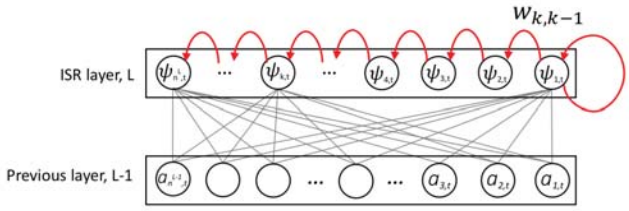
A traditional LSTM network (see section 2.2) is used as our base network structure, as this captures temporal correlations. The LSTM network is first pre-trained, then an ISR output layer replaces the original output layer of the LSTM to incorporate across frequency-level dependencies. Our proposed ISR output layer is a recurrent layer, where each neuron represents a frequency bin of the signal. A Markov chain-like recurrent structure learns the spectral dependencies from low to high frequencies. More specifically, the ISR output layer uses the output of the base LSTM network, $\boldsymbol{a}_t^{L-1}$, as input. The spectral output vector of the ISR layer is denoted as $\boldsymbol{\psi}_t$. The individual spectral response at a corresponding frequency bin is denoted as $\psi_{k,t}$, where k indexes the frequency axis. Outputs from the ISR layer are computed as follows,

$$\boldsymbol{\Delta} = \sigma(\boldsymbol{R}^L \boldsymbol{a}_t^{L-1} + \boldsymbol{\beta}^L) \tag{8}$$

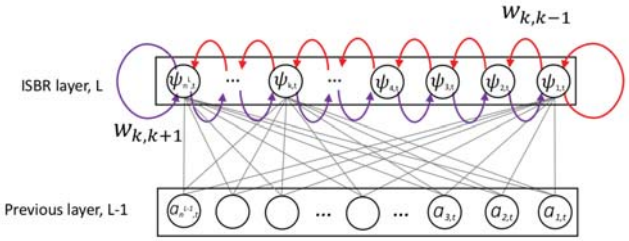
$$\psi_{1,t} = \Delta_1 + \sigma_\psi(w_{1,1} \times \psi_{1,t-1}) \tag{9}$$

$$\psi_{k,t} = \Delta_k + \sigma_\psi(w_{k,k-1} \times \psi_{k-1,t}), \quad k \in [2, n^L] \tag{10}$$

where $\boldsymbol{\Delta}$ is the vector of activations, $\{\Delta_1, \ldots, \Delta_{n^L}\}$, based on inputs from the prior LSTM layer, $\boldsymbol{R}^L \in \mathbb{R}^{n^L \times n^{L-1}}$ is the weight matrix, and $\boldsymbol{\beta}^L \in \mathbb{R}^{n^L}$ is the bias vector. $\boldsymbol{w}_{k,k-1}$ represents the weight from the $(k-1)^{\text{st}}$ to k^{th} frequency component, within the recurrent output layer. σ and σ_ψ are the activation functions for the feed-forward and recurrent paths. Activation functions are applied separately to the feed-forward and recurrent paths, since this is similar to a logistic regression-based network, which has performed well for other tasks. In equations (8–10), we see that a lower to higher frequency first-order Markovian dependency is maintained from $\boldsymbol{\psi}_{1,t}$ to $\boldsymbol{\psi}_{n^L,t}$. Note that outputs are computed sequentially from the lowest to the highest frequency neuron.



(a) Intra-spectral recurrent (ISR) layer



(b) Intra-spectral bi-directional recurrent (ISBR) layer

Fig. 1: Depiction of the proposed spectrally-dependent recurrent layers.

Additionally, we propose an intra-spectral bi-directional recurrent (ISBR) layer, which has Markov chain-style recurrent neurons from low to high frequencies and from high to low frequencies. This is done to account for spectral dependencies across both (increasing and decreasing) directions along the frequency axis. That means a certain frequency component $\psi_{k,t}$ is dependent on its immediate lower frequency component $\psi_{k-1,t}$, on its immediate higher frequency component $\psi_{k+1,t}$ and on the activations Δ_k. In addition to Eq. (8), the ISBR layer incorporates high to low spectral-frequency dependencies using the following rules:

$$\begin{aligned}\psi_{1,t} &= \Delta_1 + \sigma_\psi(w_{1,2} \times \psi_{2,t}) \\ &\quad + \sigma_\psi(w_{1,1} \times \psi_{1,t-1})\end{aligned} \tag{11}$$

$$\begin{aligned}\psi_{n^L,t} &= \Delta_{n^L} + \sigma_\psi(w_{n^L,n^L} \times \psi_{n^L,t-1}) \\ &\quad + \sigma_\psi(w_{n^L,n^L-1} \times \psi_{n^L-1,t})\end{aligned} \tag{12}$$

$$\begin{aligned}\psi_{k,t} &= \Delta_k + \sigma_\psi(w_{k,k+1} \times \psi_{k+1,t}) \\ &\quad + \sigma_\psi(w_{k,k-1} \times \psi_{k-1,t}), \quad k \in [2, n^L - 1]\end{aligned} \tag{13}$$

where $\boldsymbol{w}_{k,k+1}$ represents the weight from the $(k+1)^{\text{st}}$ to k^{th} frequency component, to account for high to low frequency correlations. The final output is therefore $\hat{\boldsymbol{y}}_t = \boldsymbol{\psi}_t^L$, which is the enhanced spectrum of the t^{th} time frame. In our paper, we use a first-order Markovian assumption along both frequency directions, but this can be extended to higher-order assumptions to capture longer spectral dependencies. We leave this as future work.

Authorized licensed use limited to: Indiana University. Downloaded on June 10,2020 at 13:51:16 UTC from IEEE Xplore. Restrictions apply.

4. EXPERIMENTS AND RESULTS

4.1. Dataset

Our proposed approach is evaluated on the IEEE speech corpus [16], which consists of 720 utterances from a single male speaker. Three non-overlapping sets are used for training, cross-validation, and testing. The training set contains 500 clean utterances, whereas 110 utterances are each used for cross-validation and testing purposes. Each utterance is combined with four noise signals (speech-shaped noise, cafeteria, factory, and babble). Clean training and validation utterances are mixed with noise signals at -3, 0, 3 dB signal-to-noise ratios (SNRs). For testing, two additional SNRs (-6 and 6 dB) are used. Each noise signal is about 4 minutes long. Ten random cuts from the first two minutes of the noises are mixed with each training and validation utterance, resulting in 60000 training signals (500 utterances $\times$ 4 noises $\times$ 3 SNRs $\times$ 10 random cuts) and 13200 validation signals (110 utterances $\times$ 4 noises $\times$ 3 SNRs $\times$ 10 random cuts). Testing utterances are mixed with ten random cuts from the last two minutes of the noises, and the total number of test utterances becomes 22000 (110 utterances $\times$ 4 noises $\times$ 5 SNRs $\times$ 10 random cuts). Note that all the signals are down-sampled to 16 kHz sampling rates and the -6 and 6 dB signals are unseen by the model during training. The random noise cuts from two halves ensures that our model does not see the noise segments in the training phase. The validation set determines the model parameters with early stopping.

The noisy-speech log-magnitude spectrograms are the inputs to our model, where the models are trained to estimate the clean-speech log-magnitude spectrograms. The spectrograms are generated using 40 ms time frames with 50% overlapping Hann windows and 640 FFT sampling points.

4.2. Experimental Setup

We experiment with two baseline deep architectures, namely, a DNN and a LSTM recurrent neural network. The structure of the DNN model is similar to [5], except pre-training and fine-tuning steps are not performed. The DNN consists of three fully connected hidden layers and each hidden layer has 321 units. The output layer has 321 units. The rectified linear (ReLU) function [17] is used as the activation function for all layers including the output layer, because all the values of magnitude spectrogram are in the range of $[0, +\infty)$. Batch normalization layers are used in between each layer so that the empirical statistics of the entire dataset remains the same. Adam optimization [18] is used with momentum and the loss function is the mean-square error. The learning rate is 0.001 and the maximum epoch number is 80. Early stopping with 5 iterations is applied for best model selection using the validation set. We use Xavier initialization [19] to initialize the model. Our proposed approach replaces the original-DNN output layer with the ISR layer, where a fine-tuning step is used to compute new output weights and biases. The proposed DNN+ISR network is denoted as D-ISR.

Our LSTM-based network consists of a single LSTM layer with 256 cells, a time distributed dense layer (321 units) and our proposed recurrent output layer (321 units). ReLU is used as the activation function for the dense and output layers. A sigmoid logistic function is used for the gate activation function, while hyperbolic tangent functions are used for the cell and hidden states. Batch normalization layers are also used in between layers. Adam optimization with a MSE loss function is used. Model initialization and parameter selection are the same as the DNN. The output layer of the LSTM network is replaced by either the ISR or ISBR layer, and the model is retrained. The two proposed LSTM approaches are denoted as L-ISR and L-ISBR, respectively, for the single- and bi-directional models.

The approaches are evaluated with three commonly-used objective metrics, namely, the Perceptual Evaluation of Speech Quality (PESQ) [20], the short-time objective intelligibly (STOI) [21] and the scale-invariant speech distortion ratio (SI-SDR) [22, 23]. These metrics are often used to assess speech enhancement, and they have been used in several studies. SI-SDR is used instead of SDR, since it has been shown that SDR can give misleading results [23].

4.3. Comparisons and Results

We compare our approach against a traditional signal-based DNN approach [5] and a traditional temporally-correlated LSTM approach [7]. Both approaches directly estimate clean speech spectrograms. The DNN approach does not consider any correlations, while the LSTM architecture considers temporal correlations. We also compare against a recently proposed model that uses frequency and temporal correlations with a LSTM [14]. This approach, denoted as L-FT, has two recurrent stages, one addresses spectral dependencies, while the second addresses temporal recurrence. A frequency LSTM (F-LSTM) layer first extracts summarized frequency information by scanning frequency sub-bands of a time frame in a sliding window manner. These summarized vectors from each F-LSTM cell are merged into a super vector that is considered as a trajectory of frequency patterns for the current time frame. Then multiple time LSTM (T-LSTM) layers use this super vector to learn temporal dependencies. This is significantly different from ours because the F-LSTM cannot determine local-spectral dependencies, since it operates at the sub-band and not frequency-bin level. We test with [14] to compare local versus subband frequency dependencies.

Table 1 shows the average performance of the different models at each noise, using the seen SNRs. The proposed L-ISBR model outperforms the other models according to PESQ scores. In terms of STOI, L-ISBR performs best in SSN and factory noise; whereas D-ISR performs best in cafe and babble noises. According to SI-SDR, L-ISBR performs best in

Authorized licensed use limited to: Indiana University. Downloaded on June 10,2020 at 13:51:16 UTC from IEEE Xplore. Restrictions apply.

Table 1: Average scores of the different models for seen SNRs (e.g. -3, 0 and 3 dB). Best results are shown in **bold**.

	PESQ				STOI				SI-SDR			
	SSN	Cafe	Factory	Babble	SSN	Cafe	Factory	Babble	SSN	Cafe	Factory	Babble
Mixture	1.95	1.86	1.83	1.77	0.71	0.62	0.65	0.59	-0.51	-2.06	-0.96	-1.97
DNN [5]	2.04	1.89	2.02	1.89	0.75	0.63	0.72	0.56	-1.75	-1.1	-1.4	-1.39
LSTM	2.12	1.97	2.05	1.95	0.77	0.64	0.76	0.62	-0.96	-1.35	-0.15	-0.44
D-ISR	2.24	2.08	2.26	2.08	0.85	**0.76**	0.86	**0.76**	-1.49	-2.91	-2.75	-3.48
L-ISR	2.27	2.21	2.29	2.11	0.82	0.68	0.84	0.72	0.06	-1.34	**0.17**	-1.3
L-ISBR	**2.3**	**2.24**	**2.31**	**2.13**	**0.88**	0.74	**0.87**	0.73	**2.35**	**-0.12**	-0.94	**-0.01**
L-FT [14]	2.12	2.01	2.07	2.04	0.82	0.74	0.82	0.66	1.04	-1.16	-0.88	-0.1

SSN, babble and cafe noises. Its worth noting that the proposed approaches each outperform the traditional DNN and LSTM approaches that do not enforce frequency-level dependencies. This is important because it means that frequency-level correlations are important for improved quality and intelligibility. Its also worth noting that the L-ISBR approach

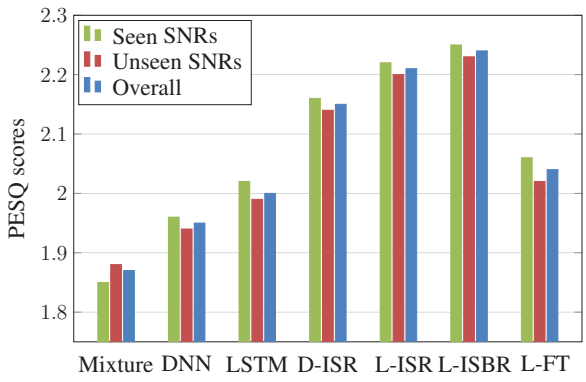
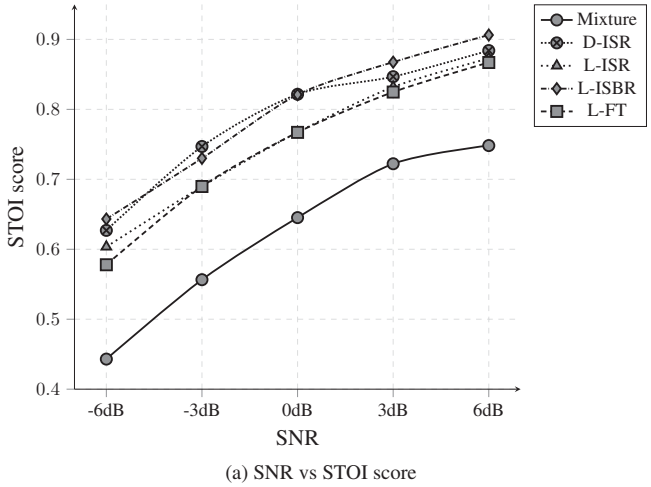


Fig. 2: PESQ scores for seen and unseen SNR conditions.

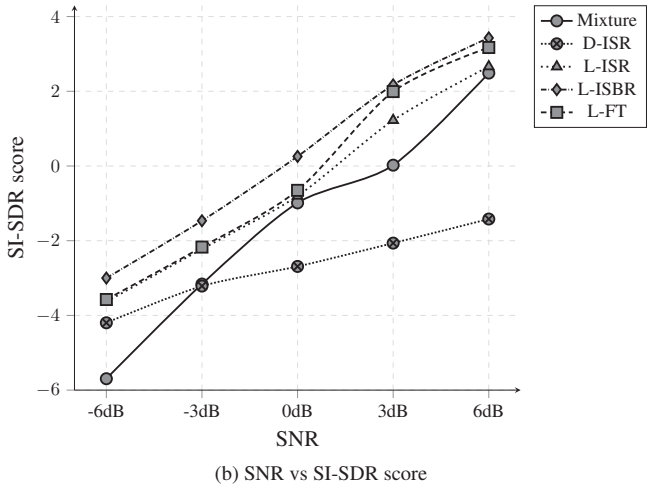
nearly always outperforms the L-FT model, indicating that local short-term spectral-dependencies outperform subband and long-term ones.

An aggregated comparison over all noise types is shown in Fig. 2, where PESQ scores are shown for the seen SNR, unseen SNR (e.g. -6 and 6 dB) and overall cases. The results show that all approaches offer improvements over the noisy speech mixture for the seen and unseen cases. Likewise, our proposed D-ISR and L-ISR approaches, respectively outperform the traditional DNN and LSTM approaches for the seen and unseen cases. The proposed L-ISBR model performs best over all other models. This occurs because the L-ISBR accurately models temporal and spectral correlations, along both frequency directions (e.g. increasing and decreasing).

Figure 3 shows STOI and SI-SDR performance at each SNR level. In terms of SI-SDR, our proposed L-ISBR approach performs best at each SNR, where it performs noticeably better at the more challenging lower SNR cases. According to STOI, the D-ISR and L-ISBR approaches perform



(a) SNR vs STOI score



(b) SNR vs SI-SDR score

Fig. 3: Average performance at each SNR for (a) STOI and (b) SI-SDR.

similarly at most SNRs, and best overall. All approaches improve objective intelligibility as compared to the noisy speech mixture. Example audio samples are available at github.iu.edu/knayem/IntraSpectral.

Authorized licensed use limited to: Indiana University. Downloaded on June 10,2020 at 13:51:16 UTC from IEEE Xplore. Restrictions apply.

5. CONCLUSION

Our proposed intra-spectral layers, along with a base LSTM network, successfully capture both temporal and spectrally correlations. The results show that these layers improve performance over traditional speech enhancement approaches, and a comparison approach that considers T-F correlations. This is exhibited over a variety of noise and SNR values. This model, however, only considers first-order spectral dependencies and does not consider phase-level dependencies. Future work will incorporate high-order spectral and phase dependencies.

6. REFERENCES

[1] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM TASLP*, vol. 22, pp. 1849–1858, 2014.

[2] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Sig. Process. Letters*, vol. 21, pp. 65–68, 2013.

[3] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. GlobalSIP*, pp. 577–581, 2014.

[4] B. O. Odelowo and D. V. Anderson, "A study of training targets for deep neural network-based speech enhancement using noise prediction," in *Proc. ICASSP*, pp. 5409–5413, 2018.

[5] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM TASLP*, vol. 23, pp. 7–19, 2014.

[6] D. S. Williamson, "Monaural speech separation using a phase-aware deep denoising auto encoder," in *Proc. MLSP*, pp. 1–6, 2018.

[7] F. Weninger, F. Eyben, and B. Schuller, "Single-channel speech separation with memory-enhanced recurrent neural networks," in *Proc. ICASSP*, pp. 3709–3713, 2014.

[8] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*, pp. 708–712, 2015.

[9] H. Zhao, S. Zarar, I. Tashev, , and C.-H. Lee, "Convolutional-recurrent neural networks for speech enhancement," in *Proc. ICASSP*, pp. 2401–2405, 2018.

[10] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, pp. 31–35, 2016.

[11] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice*. Upper Saddle River, NJ: Prentice Hall, 1st ed., 2002.

[12] P. C. Loizou, *Speech Enhancement: Theory and Practice*. Boca Raton, FL, USA: CRC, 2007.

[13] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 2009.

[14] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "Lstm time and frequency recurrence for automatic speech recognition," in *Proc. ASRU*, pp. 187–191, 2015.

[15] J. Deng, B. Schuller, F. Eyben, D. Schuller, Z. Zhang, H. Francois, and E. Oh, "Exploiting time-frequency patterns with lstm-rnns for low-bitrate audio restoration," *Neural Computing and Applications*, pp. 1–13, 2019.

[16] E. Rothauser, "IEEE recommended practice for speech quality measurements," *IEEE Trans. on Audio and Electroacoustics*, vol. 17, pp. 225–246, 1969.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, pp. 315–323, 2011.

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

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, pp. 249–256, 2010.

[20] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. ICASSP*, vol. 2, pp. 749–752, 2001.

[21] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE/ACM TASLP*, vol. 19, pp. 2125–2136, 2011.

[22] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE/ACM TASLP*, vol. 14, pp. 1462–1469, 2006.

[23] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?," in *Proc. ICASSP*, pp. 626–630, 2019.

Authorized licensed use limited to: Indiana University. Downloaded on June 10,2020 at 13:51:16 UTC from IEEE Xplore. Restrictions apply.