AFRL-RI-RS-TR-2018-152

DEEP NEURAL NETWORKS FOR SPEECH SEPARATION WITH APPLICATION TO ROBUST SPEECH RECOGNITION

THE OHIO STATE UNIVERSITY

JUNE 2018

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2018-152 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
WAYNE N. BRAY
Work Unit Manager

/ S /
WARREN H. DEBANY, JR
Technical Advisor, Information
Exploitation and Operations Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUN 2018	FINAL TECHNICAL REPORT	SEP 2015 – DEC 2017

4. TITLE AND SUBTITLE

DEEP NEURAL NETWORKS FOR SPEECH SEPARATION WITH APPLICATION TO ROBUST SPEECH RECOGNITION

5a. CONTRACT NUMBER
FA8750-15-1-0279

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

DeLiang Wang

5d. PROJECT NUMBER
G2AU

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Ohio State University, The Office of Sponsored Programs
1960 Kenny Rd
Columbus, OH 43210-1016

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGC
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2018-152

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project will investigate the speech separation problem and apply the results of speech separation to robust automatic speech recognition (ASR). Speech separation has been recently formulated as a time-frequency masking problem, which shifts the research focus to supervised learning. The proposed effort will employ deep neural networks (DNN) as the learning machine for supervised separation The proposed research aims to achieve the following objectives. The first objective is separation of speech from background noise. This will be accomplished by training DNN classifiers on extracted acoustic-phonetic features. The second objective is integration of spectrotemporal context for improved separation performance. Conditional random fields will be used to encode contextual constraints. The third objective is to achieve robust ASR in the DNN framework through integrated acoustic modeling and separation. The performance of the proposed system will be systematically evaluated using the recently constructed CHIME-2corpus.

15. SUBJECT TERMS
Deep Neural Network Speech Separation; Time-frequency masking; Automatic Speech Recognition; Deep Neural Networks for Speech Separation with Application to Robust Speech Recognition

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			WAYNE N. BRAY
U	U	U	UU	58	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

This AFRL contract project was funded in late September 2015, with actual work starting in January 2016. Although the contract was extended to December 2017, the project was completed at the end of September 2017. One doctoral student (Zhong-Qiu Wang) was supported by the project. Another doctoral student (Peidong Wang) was partly funded by this project. This report summarizes the progress made throughout the project period.

Major advances have been made mainly along the following four fronts: monaural speech separation, monaural robust automatic speech recognition (ASR), time-frequency (T-F) masking based beamforming and its application to robust ASR, and spatial features for T-F masking and robust ASR. The foundation of our approach is supervised monaural speech separation in the form of T-F masking and deep neural networks (DNNs). In multi-channel (multi-microphone) cases, monaural T-F masking as well as masking utilizing spatial features serves to guide beamforming and masking-guided beamforming elevates robust ASR performance by large margins. These advances are described in the following sections. The Appendix at the end lists the publications resulting from this project.

2 METHODS, ASSUMPTIONS AND PROCEDURES

For speech separation or enhancement in the short-time Fourier transform (STFT) domain, the ideal solution is to obtain the clean magnitude and clean phase of the target speech, with which clean speech signals can be re-synthesized perfectly. However, phase information is difficult to estimate from noisy utterances. Therefore, many studies focus on recovering the clean magnitude and use the noisy phase for re-synthesis. Recently, deep learning has shown great potential for supervised speech separation since the PI's lab first introduced deep learning to the

domain of speech separation or enhancement [60] [71] [57]. Deep neural networks (DNNs) have been used to estimate an ideal time-frequency (T-F) mask [60], or directly map to clean magnitudes from noisy ones [71] [21]. In [59], Wang *et al.* carefully compare T-F masking and spectral mapping, and suggest that masking should be preferred.

In this project, we investigate how to leverage output patterns or output context information for better mask estimation. We emphasize that improving mask estimation can benefit a lot of tasks, such as speech enhancement [60], speech de-reverberation [21], phase reconstruction [68], and robust automatic speech recognition (ASR) [61] [39] and speaker recognition.

There are clearly strong output patterns in the ideal binary mask (IBM) or the ideal ratio mask (IRM). These output patterns can be potentially utilized to improve mask estimation, as the output patterns represent a kind of regularization that the estimated masks should conform with. In recent years, various neural networks have been employed for mask estimation, such as DNNs [60], convolutional neural networks (CNNs), recurrent neural networks (RNNs) [29] with long-short term memory (LSTM) [65] [9], but none of them explicitly utilize output context for mask estimation.

The key idea of the proposed approach is to use the estimated mask of the previous frames as the additional input to predict the mask at the current frame. This is akin to incorporating an *n*-gram language model defined on the output patterns into traditional frame-level mask estimation. In this way, the contextual information in the output is explicitly utilized, and can be potentially modeled using an RNN. However, formulating it as an RNN would make the optimization process difficult, as the network would be very deep if we unfold the network through time for optimization. In addition, the output activation function in supervised separation is normally sigmoidal, likely leading to vanishing gradient problems during optimization.

We thus propose a recurrent deep stacking network, in which the estimated masks of the previous frames are updated at the end of every training epoch, and the updated estimated masks are then used as additional inputs to train the DNN in the next epoch. At the test stage, we need the estimated masks of several previous frames to predict the mask at the current frame. To obtain them, we formulate the DNN as an RNN to make predictions sequentially. The recurrent connections are from the output units of previous frames to the input of the current frame. In addition, we propose to use the L_1 loss for mask estimation.

After the introduction of deep learning based T-F masking in Section 1.1, the proposed recurrent deep stacking network is presented in Section 1.2, followed by experimental setup and evaluation results detailed in Section 1.3 and 1.4.

3 MASK ESTIMATION

Supervised speech separation uses a supervised learning machine, such as DNN, CNN and RNN, to estimate the IRM [38] [59] from a noisy utterance, among other training targets. With the estimated IRM, the clean magnitude can be reconstructed by point-wise multiplication in the time-frequency domain. Traditionally, the square root of the Wiener filter is used as the IRM for training. In this study, we use a slightly different ideal mask as the training target [58], i.e.

$$M_{t,f} = \min\left(1, \frac{S_{t,f}^2}{Y_{t,f}^2}\right) \tag{1}$$

Where $S_{t,f}^2$ and $Y_{t,f}^2$ represent the speech energy and mixture energy within a specific T-F unit, respectively. The values in this ideal mask are capped between 0 and 1, so that the mask values in different channels are bounded in the same range suitable for training, and sigmoidal units can be utilized as the activation function at the output layer. The motivation for using this target is that after multiplying this ideal mask with the mixture power spectrogram, the resulting

power spectrogram would be closer to the clean power spectrogram than using the standard IRM.

Conventionally, mean square error, i.e. L_2 loss, is used for mask estimation. In this project, we propose to use L_1 as the loss function. Mathematically, the loss and its error gradient are defined in the following equations,

$$Loss = \frac{1}{T}\sum_{t}\sum_{f}\left|M^*_{t,f} - M_{t,f}\right| \tag{2}$$

$$\frac{\partial Loss}{\partial M^*_{t,f}} = \frac{1}{T}\left(1\left[M^*_{t,f} > M_{t,f}\right] - 1\left[M^*_{t,f} \le M_{t,f}\right]\right) \tag{3}$$

where T is the total number of frames in the training data, $M^*_{t,f}$ represents the estimated mask at a specific T-F unit, and $1[\cdot]$ is the indicator function. By using the L_1 loss, we implicitly assume that the error term distribution is Laplacian [7]. We think that this assumption is reasonable considering the sparseness of speech and noise in the time-frequency domain, i.e., for many T-F units, only one source dominates. Because of this property, the histogram of the ideal masks would be largely concentrated around 0 and 1, and exponentially decay from 0 (or 1) to 0.5, at least when room reverberation is not considered. In such cases, it is more reasonable to assume that the error term distribution is Laplacian as well. In our experiments, we will demonstrate that if we use the L_1 loss for training, the error histogram on the validation set would be close to Laplacian, whereas if we use the L_2 loss for training, the error histogram on the validation set would not be similar to Gaussian.

After obtaining the estimated ratio mask from a noisy utterance, we multiply it point-wisely with the noisy power spectrogram using Eq. (4) to get the enhanced power spectrogram.

$$\hat{Y}^2 = M^* \otimes Y^2 \tag{4}$$

where $\otimes$ represents point-wise matrix multiplication in the time-frequency domain. We use the

noisy phase directly for re-synthesis.

3.1 Recurrent Deep Stacking Networks

Our model is essentially a DNN. The input is a combination of noisy features and the estimated masks of several previous frames, i.e.

$$< M^*_{t-w}, \dots, M^*_{t-1}, x_{t-w}, \dots, x_t, \dots, x_{t+w} > \quad (5)$$

where w is the half-window length, and x_t and $M^*_{t,f}$ represent the extracted noisy features and the estimated mask at frame t, respectively. The output is the ideal mask at the central frame, i.e. M_t. By using $M^*_{t-w}, \dots, M^*_{t-1}$ as the additional inputs to predict M_t, the output context is explicitly utilized for mask estimation. It is similar to including a $(w+1)$-gram language model defined on the output patterns into conventional frame-wise mask estimation, which provides useful constraints on M_t after obtaining $M^*_{t-w}, \dots, M^*_{t-1}$. The overall training process is shown in Figure 1. We update all at the end of every training epoch, and use the updated M^*_t as additional inputs for DNN training in the following epoch. The effect is similar to implicitly stacking N DNNs, where N is the number of training epochs.

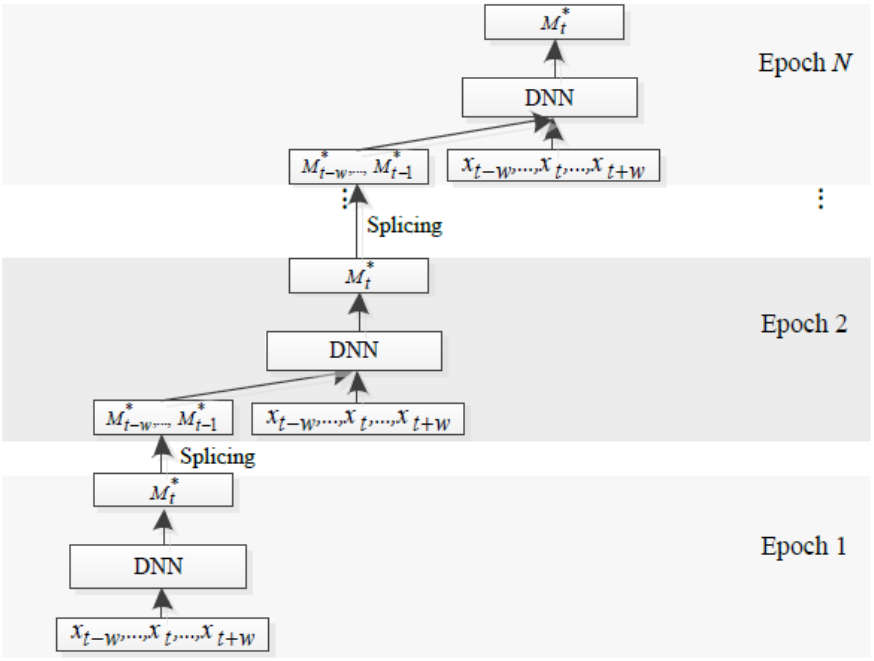

Figure 1 Illustration of the training process of the proposed recurrent deep stacking network.

The DNN model at each epoch is one module in the stack. In this way, a large context window at the input level can also be implicitly used, because the outputs of the DNN in the previous epoch are spliced together as additional inputs in the current epoch, and each output is obtained by the previous DNN using multiple frames. Therefore, the more DNNs we stack, the more input context would be potentially utilized.

Although we stack many DNNs at the training stage, there is no need to save all of them for testing. Interestingly, we only need to save the DNN model after the last training epoch. At the test stage, we formulate it as an RNN, where the recurrent connections are from the output units of the previous frames to the input of the current frame. This way, we can make predictions sequentially. More specifically, since our approach uses only past estimated masks, all the input features will be available when it comes to the current frame. When predicting the mask of the first frame, we just set all the estimated masks of the previous frames to zeros.

We point out that we can actually train the model as an RNN. However, it incurs many optimization difficulties, such as vanishing gradient problems as pointed out in the introduction. In addition, training an RNN from the scratch is much slower than DNN training, because we have to move frame by frame in the forward and backward pass. Furthermore, the data shuffling in RNN training is not as good as that in DNN training due to its sequential nature. More importantly, more advanced DNN training techniques, such as batch normalization [31] and residual connections [22], can be easily incorporated into our method, while including these techniques into RNN or LSTM training may be quite difficult [41].

Several previous studies have applied deep stacking or ensemble networks [30] to supervised speech separation [39] [73] [76], but only a limited number (two or three) of DNNs or several shallow networks are stacked. In these studies, each module in the stack is trained from the scratch using the outputs from lower modules together with original noisy features, and therefore each module has to go through a number of epochs for training. In our approach, we train our DNN for a fixed number of epochs, and the DNN model at each epoch is considered as an implicit module in the stack. Thus, a large number of modules can be stacked, and more context information in the input level can be utilized due to stacking. Most differently, all the stacked models in the previous studies have to be saved for testing, while only one model needs to be saved for our method. By formulating the trained DNN model as an RNN at the test stage, we explicitly incorporate output context information into mask estimation.

3.2 Results and Discussion

We conduct our experiments on the noisy and reverberant CHiME-2 dataset (task-2) [54]. The reverberant and noisy signals are created by first convolving the clean signals in the WSJ0-5k corpus with binaural room impulse responses (BRIRs), and then adding reverberant noises

recorded at six different SNR levels linearly spaced from -6 dB to 9 dB. The noises are recorded in a domestic living room and kitchen, which include a rich collection of sounds, such as electronic devices, background speakers, distant noises, footsteps, background music, and so on. The BRIRs are recorded in the same environments. There are 7,138 utterances in the training data (~14.5h in total), 409 utterances for each SNR level in the validation data (~4.5h in total), and 330 utterances for each SNR level in the test data (~4h in total).

Our system is monaural in nature. We merge the two-channel signals by a simple average. The effect is the same as applying delay-and-sum beamforming to the binaural signals, because the speaker is designed to be approximately in front of the two microphones. In our study, we use the averaged reverberant signals as the reference signals, so that we can construct ideal masks for DNN training, and calculate various evaluation metrics, such as the Short-Time Objective Intelligibility (STOI), Perceptual Estimation of Speech Quality (PESQ) and Signal-to-Distortion Ratio (SDR). STOI and PESQ values are the objective measures of speech intelligibility and quality, respectively. Note that our model only tries to remove or attenuate additive noises.

The DNN in our study has four hidden layers, each with 2048 exponential linear units (ELUs) [12]. In our experiments, ELUs lead to faster convergence and better performance over the commonly used rectified linear units (ReLUs). The dropout rates of the input layer and all the hidden layers are set to 0.05. Besides the estimated masks of the previous frames, we use log power spectrogram features with a symmetric 19-frame context window as the inputs, meaning ww is set to 9. The window length is 25 ms and the window shift is 10 ms. We perform 512-point fast Fourier transform (FFT) when extracting log power spectrograms. The dimension of the log

Table 1 Comparison of SDR scores on test set (boldface indicates best result)

Method	Loss function	-6dB	-3dB	0dB	3dB	6dB	9dB	Average
Unprocessed	-	-2.55	-1.12	1.11	2.78	4.48	5.78	1.75
DNN	L_2	8.94	10.42	12.28	13.90	15.60	17.51	13.11
DNN	L_1	9.76	11.12	12.88	14.43	16.05	17.89	13.69
Recurrent Deep Stacking Networks	L_1	10.35	11.70	13.43	14.91	16.46	18.25	14.18
Recurrent Deep Stacking Networks	+Signal Approximation	**10.76**	**12.06**	**13.69**	**15.08**	**16.57**	**18.33**	**14.41**
LSTM [65]	Signal Approximation	10.46	11.85	13.40	14.86	16.34	18.07	14.17

Table 2 Comparison of PESQ scores on test set

Method	Loss function	-6dB	-3dB	0dB	3dB	6dB	9dB
Unprocessed	-	2.138	2.327	2.492	2.662	2.854	3.049
DNN	L_2	2.791	2.940	3.076	3.217	3.356	3.506
DNN	L_1	2.888	3.049	3.186	3.321	3.449	3.586
Recurrent Deep Stacking Networks	L_1	2.996	3.162	3.295	3.432	3.533	3.663
Recurrent Deep Stacking Networks	+Signal Approximation	**3.014**	**3.181**	**3.315**	**3.448**	**3.559**	**3.685**
Phoneme-specific Speech Separation [62]	Signal Approximation	2.731	2.884	3.011	3.146	3.284	3.430

Table 3 Comparison of STOI scores on test set

Method	Loss function	-6dB	-3dB	0dB	3dB	6dB	9dB
Unprocessed	-	0.737	0.778	0.813	0.852	0.881	0.909
DNN	L_2	0.871	0.895	0.914	0.932	0.944	0.957
DNN	L_1	0.878	0.901	0.919	0.936	0.946	0.959
Recurrent Deep Stacking Networks	L_1	**0.886**	**0.909**	**0.925**	**0.940**	**0.950**	**0.961**
Recurrent Deep Stacking Networks	+Signal Approximation	0.884	0.907	0.924	0.939	0.948	0.959
Phoneme-specific Speech Separation [62]	Signal Approximation	0.861	0.886	0.905	0.922	0.935	0.949

power spectrogram is therefore 257, and so is the output dimension in our DNN. No pre-emphasis is performed before FFT. All the features are globally mean-variance normalized before DNN training. We re-compute the mean and variance of the estimated masks after every update. Note that we need to feed all the training data to update the estimated masks after every training epoch. The network is trained using AdaGrad with a momentum term for 30 epochs. The learning rate is fixed at 0.005 in the first 10 epochs and linearly decreased to 10^{-4} in subsequent epochs. The momentum is linearly increased from 0.1 to 0.9 in the first 5 epochs and fixed at 0.9 afterwards.

3.2.1 L1 Loss for Mask Estimation

The comparison between the L_1 and L_2 loss is presented in the second and third entries in Table 1, Table 2 and Table 3. We can clearly see that using the L_1 loss for DNN training leads to consistently better SDR, PESQ and STOI scores at all the six SNR levels. Note that in our experiments, we just change the loss functions for DNN training and fix all the other hyper-parameters in order to make a fair comparison. In Figure 2, we plot the histogram of the ideal masks. Clearly, the distribution has two modes around 0 and 1, and exponentially decays towards the middle. In Figure 3, we plot the histograms of the errors at all the T-F units on the -6 dB subset of the validation set, one for each loss function. We can see that if we use the L_1 loss for training, the histogram is pretty similar to Laplacian distribution, which justifies the assumptions. In contrast, if we use the L_2 loss for training, the histogram clearly does not resemble a Gaussian. We think that this explains why the L_1 loss leads to better performance in our experiments.

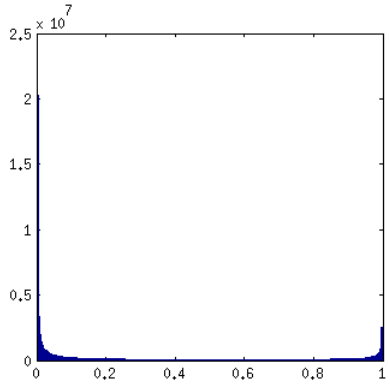

Figure 2 The histogram of all the values in the ideal masks on the -6 dB subset of the validation set.

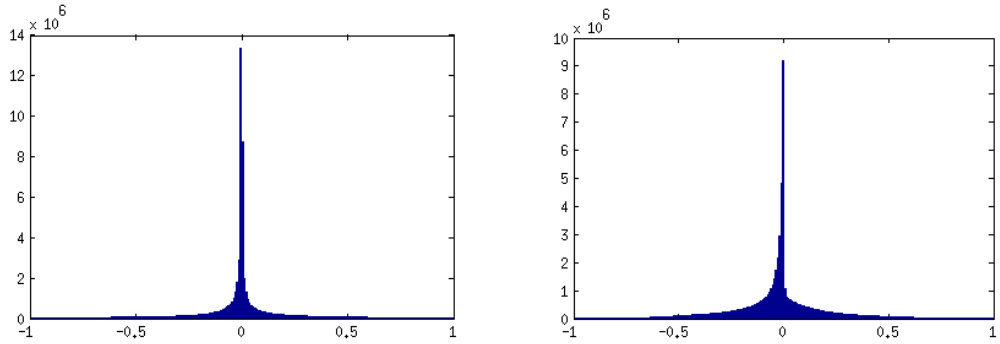

Figure 3 Error histograms on the -6 dB subset of the validation set. The left histogram is obtained using the DNN trained with the L1 loss, and the right histogram is obtained using the DNN trained with the L2 loss.

3.2.2 Performance of Recurrent Deep Stacking Networks

We first train our recurrent deep stacking networks using the L_1 loss until convergence. Then we switch to the signal approximation loss used in [62] and further train the model until convergence. Note that in our experiments, training the model using the signal approximation loss from the scratch gives much worse performance than using the L_1 or the L_2 loss, as is suggested in [66]. By comparing the third and fourth entries in Table 1, Table 2, and Table 3, we can see that modeling output context leads to clear improvements especially in terms of SDR and PESQ scores. Further training the model using the signal approximation loss leads to better SDR and PESQ results while slightly worse STOI numbers.

We compare our methods with several other studies with experiments on the same dataset in the literature. All of them use log power spectrogram features. In [62], a phoneme-specific speech separation approach that utilizes the information from robust ASR systems is proposed. Their model for speech separation uses a number of DNNs, one for each phoneme, trained with the signal approximation loss. Only STOI and PESQ scores are reported in their study. From the last two entries in Table 2 and Table 3, we can see that our results are clearly better. The results reported in [65] represent a series of efforts [67] [66] [15] [10] by several groups on the CHiME-

2 dataset. Only SDR scores are reported to measure the performance of speech separation in their studies. As reported in the last two entries of Table 1, our model obtains slightly better results than the strong LSTM model trained with the signal approximation loss reported in [65]. It should be noted that in [65] better SDR results are reported by using phase information and information from a robust ASR system.

3.3 Conclusion

We have proposed recurrent deep stacking networks to explicitly incorporate contextual information in output patterns for mask estimation. In addition, we have proposed to use the L_1 loss for mask estimation, which gives us consistently better results than the widely used L_2 loss. Experimental results on the CHiME-2 dataset (task-2) are encouraging. The proposed recurrent deep stacking algorithm can be applied to improve many other tasks, in which the output context provides useful constraints, such as acoustic modeling in automatic speech recognition and sequence labeling in natural language processing. One potential drawback of the proposed approach is that the input dimension is dependent on the output dimension. Nonetheless, the findings in this study suggest that, at a minimum, explicitly modeling output patterns likely yields consistent improvements for time-frequency masking.

4 MONAURAL ROBUST ASR

Modern ASR technology has been successfully used in many real-world scenarios. While microphone arrays are widely employed. Monaural ASR is easier to deploy and more desirable in many situations. This section investigates monaural ASR in adverse real-world scenarios.

Recently, one of the most popular monaural acoustic model types is the convolutional, long short-term memory, fully connected deep neural networks (CLDNNs) [45]. Applying the wide

residual (convolutional) network and bidirectional long short-term memory (BLSTM) layers in a CLDNN framework, wide residual BLSTM network (WRBN) yields the best performance on the monaural speech recognition task using the baseline language model in the 4th speech separation and recognition challenge (CHiME-4) [25]. WRBN may, however, be improved using better LSTM dropout methods and speaker adaptation techniques.

Dropout for LSTM has shown to be effective to alleviate the overfitting problem in the RNN training process [27]. For speech recognition tasks, Moon et al. propose a *rnnDrop* method [37]. It samples a dropout mask once per utterance and applies the mask on a cell vector. The method of Gal and Ghahramani samples the dropout masks similarly but applies to the input and hidden vectors (Gal dropout) [18]. Semeniuta et al. compare two dropout mask sampling approaches, per-step (frame-wise) and per-sequence (utterance-wise) [48]. They propose to apply dropout on a cell update vector (Semeniuta dropout). Cheng et al. conduct extensive experiments on the dropout methods for LSTMs and conclude that applying utterance-wise sampled dropout masks on the output, forget, and input gates yields the best result (Cheng dropout) [11].

Speaker adaptation aims at attenuating the distribution mismatch between the training and test data caused by speaker differences. The techniques can be classified into three categories, feature-space, model-space, and feature augmentation based [36]. One of the dominant techniques in the feature space is the feature-space maximum likelihood linear regression (fMLLR) [19]. To apply fMLLR to DNN based acoustic models, a well-trained Gaussian mixture model is used to obtain fMLLR features, upon which the DNN based system is built. An MLLR based iterative adaptation technique is also proposed to update Gaussian parameters using the decoding result in the previous iteration [69]. Another popular feature-space technique is linear input network (LIN) [47] [39]. It learns a speaker-specific linear transformation of the

acoustic model input. For commonly used model-space techniques, a subset of DNN parameters are adapted. These include linear hidden network (LHN) [35], learning hidden unit contributions (LHUC) [50], and recently proposed speaker adaptation for batch normalized acoustic models [64]. For feature augmentation based methods, auxiliary features, such as i-vectors and speaker-specific bottleneck features, are used as additional information for the acoustic model [46] [51].

The rest of this section is organized as follows. In Section 2.1 we explain utterance-wise recurrent dropout and iterative speaker adaptation. In Sections 2.2 and 2.3, we show the experiment setup and results. Finally, we provide concluding remarks in Section 2.4.

4.1 System Description

A DNN-HMM based monaural speech recognition system consists of two parts, an acoustic model and a decoder. Modifications to the system can be conducted in roughly three ways: acoustic model related, interaction between the acoustic model and the decoder, and decoder related. We improve WRBN in all three categories. For acoustic model training, we use a new utterance-wise recurrent dropout method. To adapt the acoustic model using the decoder, we propose an iterative speaker adaptation technique. For the parameters related to the decoder, we enlarge beamwidth in the decoding graph.

4.1.1 Utterance-Wise Recurrent Dropout

A typical LSTM layer can be expressed by the three equations below.

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \\ \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \\ \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \\ f(\mathbf{W}_g\mathbf{x}_t + \mathbf{U}_g\mathbf{h}_{t-1} + \mathbf{b}_g) \end{pmatrix} \tag{6}$$

$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \mathbf{g}_t \tag{7}$$

$$\mathbf{h}_t = \mathbf{o}_t \otimes f(\mathbf{c}_t) \tag{8}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ denote the input, forget, and output gates at step t, and $\mathbf{g}_t$ the vector of cell updates. $\mathbf{c}_t$ is the updated cell vector, and $\mathbf{c}_t$ is used to update the hidden state $\mathbf{h}_t$. $\boldsymbol{\sigma}$ is the sigmoid function, and f is typically chosen to be a tangent hyperbolic function.

In WRBN, Eq. (6) is simplified to Eq. (9) below,

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1}) \\ \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1}) \\ \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1}) \\ f(\mathbf{W}_g \mathbf{x}_t + \mathbf{U}_g \mathbf{h}_{t-1}) \end{pmatrix} \tag{9}$$

One major difference between WRBN and conventional DNN based acoustic models is WRBN's emphasis on utterance-wise training [61] [25]. In order to train the LSTM in an utterance-wise fashion, the dropout method should be both recurrent and with little temporal information loss. We list the dropout methods satisfying both requirements in (10)-(13), corresponding to *rnnDrop* by Moon et al., Gal dropout, Semeniuta dropout, and Cheng dropout, respectively. Dropout is denoted as a $d()$ function.

$$\mathbf{c}_t = d(\mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \mathbf{g}_t) \tag{10}$$

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i d_x(\mathbf{x}_t) + \mathbf{U}_i d_h(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_f d_x(\mathbf{x}_t) + \mathbf{U}_f d_h(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_o d_x(\mathbf{x}_t) + \mathbf{U}_o d_h(\mathbf{h}_{t-1})) \\ f(\mathbf{W}_g d_x(\mathbf{x}_t) + \mathbf{U}_g d_h(\mathbf{h}_{t-1})) \end{pmatrix} \tag{11}$$

$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes d(\mathbf{g}_t) \tag{12}$$

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} d_i(\sigma(\mathbf{W}_i \boldsymbol{x}_t + \mathbf{U}_i \mathbf{h}_{t-1})) \\ d_f(\sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1})) \\ d_o(\sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1})) \\ f(\mathbf{W}_g \mathbf{x}_t + \mathbf{U}_g \mathbf{h}_{t-1}) \end{pmatrix} \tag{13}$$

A potential problem of (10) is that the cells that are dropped out may be completely excluded

from the whole training process of the utterance. Eq. (11) may suffer from the same problem since different gates share the same masks in this method. Eqs. (12) and (13) apply dropout only on a part of the vectors, which may make the remaining part vulnerable to overfitting. Our utterance-wise recurrent dropout, shown below, tries to avoid the problems in the above dropout methods:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i d_{xit}(\mathbf{x}_t) + \mathbf{U}_i d_{hi}(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_f d_{xft}(\mathbf{x}_t) + \mathbf{U}_f d_{hf}(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_o d_{xot}(\mathbf{x}_t) + \mathbf{U}_o d_{ho}(\mathbf{h}_{t-1})) \\ f(\mathbf{W}_g d_{xgt}(\mathbf{x}_t) + \mathbf{U}_g d_{hg}(\mathbf{h}_{t-1})) \end{pmatrix} \quad (14)$$

Four independently sampled utterance-wise masks are applied to all of the four hidden vectors. For the dropout on the input vectors, we opt for the conventional frame-wise method since applying utterance-wise dropout may completely lose the information in some feature dimensions.

4.1.2 Iterative Speaker Adaptation

Speaker adaptation is commonly used in the best-performing systems of the CHiME-4 challenge. Using the decoded path as the label, the acoustic model can be adapted to specific test speakers, reducing the mismatch between the training and test data. In our work, we apply the unsupervised LIN speaker adaptation [47]. For each speaker in the test set, we train an 80x80 linear input layer. This layer is shared among the three input channels in WRBN, corresponding to static, delta, and delta-delta features. Observing a significant improvement brought by speaker adaptation, we propose to iterate the adaptation process by using the newly generated decoding result as the label for another adaptation iteration. Note that the decoding result here is the final result after the RNN language model rescoring. This iterative adaptation method is similar to a prior work using MLLR [69], but our work is in the context of the LIN adaptation for a DNN

based acoustic model.

There are two ways to conduct iterative speaker adaptation, by simply changing the label and keeping all other settings the same, or by stacking an additional linear input layer in each iteration. Note that, although mathematically multiple linear layers amount to a single layer, the second method ensures that the "acoustic model" (the stacked linear layer(s) and the original acoustic model) being adapted is the same one that generated the adaptation label. We conduct experiments on both methods and compare them in this study.

4.1.3 Large Decoding Beamwidths

Due to the differences in training platforms (the one in [25] is Chainer and ours TensorFlow) and decoding systems, our result using the original WRBN is slightly worse than the one reported in [25]. To compensate this system bias, we keep the WRBN acoustic model fixed and adjust the decoding parameters in the Kaldi scripts [44]. Specifically, we make the beamwidth and lattice beamwidth ten times larger than those used in the original WRBN. We also enlarge the lower and upper boundaries of the number of active tokens.

Unlike segment-wise trained conventional DNN acoustic models, WRBN takes as input complete utterances. The decoders in WRBN based systems, in our opinion, may also need to be adjusted such that relatively long-term dependencies are kept. Note that although decoding beamwidths are larger, an empirical observation is that the decoding speed is not influenced greatly.

4.2 Experimental Setup

4.2.1 Dataset

Our experiments are conducted on the CHiME-4 corpus [55]. It is a read speech corpus with the objective of distant-talking ASR. There are two types of data, recorded and simulated. The

real data is recorded in noisy environments, including bus, cafe, pedestrian area, and street junction. The simulated data, on the other hand, is generated by artificially mixing clean speech with noisy backgrounds. The ultimate goal of the CHiME-4 challenge is to recognize the real recorded utterances.

The training set of the CHiME-4 corpus contains 1600 real utterances and 7318 simulated utterances for each of the six microphone channels. The real utterances are uttered by 4 speakers and the simulated utterances are from the 83 speakers of the WSJ0 training set (SI-84). For the monaural task, the development set consists of 410 real utterances and 410 simulated utterances for each of the four acoustic environments, i.e. bus, cafe, pedestrian area, and street junction. Similarly, the monaural test set has 330 real recordings and 330 simulated utterances for each environment. The speakers in the training, development, and test set do not overlap. The utterances in the development and test set are randomly chosen from the six utterances recorded by the corresponding six microphones. Note that channels with hardware issues or masked by the user's hands or clothes, i.e. failed channels, are not selected.

4.2.2 Implementation Details

Using the same decoding parameters as in the original WRBN based system, our result is 0.3% (absolute) worse than reported in [25]. We think this may be caused by the differences in training platforms and decoding systems. So we keep the WRBN model fixed and adjust the decoding parameters. Setting the decoding beamwidth to 180, lattice beamwidth 120, the minimal number of active tokens 20000 and the maximal number of tokens 80000, we are able to get a word error rate (WER) of 10.43%. Since the reported WER without speaker adaptation is 10.4%, we think that the system bias may be compensated for by the new decoding parameters.

We fine-tune the WRBN using our utterance-wise recurrent dropout for five epochs. In

addition to the dropout on LSTM, we also apply conventional dropout in the residual blocks [25]. All dropout rates are set to 0.2. We make use of the Adam optimizer and set the initial learning rate to 10^{-5}.

After language model rescoring, we apply LIN based iterative speaker adaptation. For each of the real and simulated set, we train a linear layer for each speaker for ten epochs. The optimizer is Adam and the initial learning rate is 10^{-4}. The linear layers, i.e. the 80x80 weight matrices, are initialized to be identity matrices. After the first adaptation process, we get the language model rescored result and use it as the label for the next iteration. For the straightforward method, i.e. simply replacing the adaptation label with a new one, we reuse the network structure and reinitialize the linear layers to identity matrices. For the method of stacking an additional layer in each iteration, we take the combination of the stacked layer(s) in the previous iteration(s) and the original acoustic model as the new acoustic model, keep them fixed, and train a new linear layer for them. In this work, we apply iterative speaker adaptation for three iterations, including the first adaptation process. During the WER calculation, language model rescoring, and speaker adaptation, all of the language model weights are chosen based on the development set results.

4.3 Results and Discussion

4.3.1 Results and Comparisons

Our model obtains a WER of 8.28% on the real recorded data of the CHiME-4 evaluation set. The results and the comparisons with the best monaural speech recognition systems are shown in Table 4. Baseline and Unconstrained denote the baseline RNN language model and unconstrained language models, respectively.

Table 4 WER (%) comparisons of the proposed model and two best monaural ASR systems

System	Baseline		Unconstrained	
	simu	real	simu	real
Du *et al.*	13.62	11.15	11.81	**9.15**
WRBN	11.68	**9.88**	11.11	9.34
Proposed	11.14	**8.28**	-	-

Our model outperforms the previous best model using the baseline RNN language model by 16.19% relatively. It is even better than the best model using an unconstrained language model by 9.51% relatively. We expect the WER of our system to be further reduced using a better language model, especially when combined with our iterative speaker adaptation technique.

4.3.2 Results in Different Environments

We test the generalization ability of our model by comparing it with the original WRBN in all four environments. The comparisons are shown in Table 5. Note that the WRBN results are those using the unconstrained language model [25]. Four acoustic environments are denoted as *bus*, *caf*, *ped*, and *str*.

Table 5 WER (%) comparisons in different acoustic environments

Environment	WRBN		Proposed	
	simu	real	simu	real
bus	8.07	13.22	**8.03**	**11.87**
caf	13.17	9.45	**12.94**	**8.65**
ped	**10.22**	7.75	10.44	**6.65**
str	**12.98**	6.93	13.15	**5.96**
average	**11.11**	9.34	11.14	**8.28**

The results show that our model is more robust than the original WRBN in all real scenarios by substantial margins. For the simulated data, in addition to language model differences, we

think the limitations of current simulation techniques may also be part of the reason why the results of the two methods are close [33].

4.3.3 Step-by-Step Results

The results on the test set after each step are shown in Table 6. Note that we add the results after one iteration of the speaker adaptation in the *speaker adaptation* row.

Table 6 Step-by-step WERs (%)

steps	simu	real
original WRBN	13.03	10.74
+ large beamwidth	12.72	10.43
+ modified Gal dropout	**12.40**	**9.72**
+ speaker adaptation	11.52	8.81
+ iterative speaker adaptation	**11.14**	**8.28**

The system bias is compensated by enlarging the decoding beamwidths. After applying the utterance-wise recurrent dropout, the WER is reduced to 9.72%, which is already better than the final result using the baseline language model in [25]. One iteration of the speaker adaptation yields a WER of 8.81%, outperforming the corresponding result 9.88% by 10.83% relatively. Using the speaker adaptation for two more iterations, we observe a further improvement and get our best result of 8.28%.

4.3.4 Results of Two Iterative Speaker Adaptation Methods

The results of the two iterative speaker adaptation methods are shown in Table 7. The first adaptation method, denoted as *Iter*, simply changes the label and reuses the structure of the previous iteration. The second method stacks an additional linear layer in each iteration and is thus denoted as *Stack*.

Table 7 WER (%) comparisons of two iterative speaker adaptation methods

Iterations	Iter		Stack	
	tri-gram	RNN	tri-gram	RNN
1	11.01	8.81	11.01	8.81
2	10.59	**8.51**	**10.32**	8.52
3	10.42	**8.28**	**10.08**	8.45

While *Iter* yields better results after RNN language model rescoring, *Stack* performs better when using the simple trigram language model. The better tri-gram results and smaller improvements brought by the language model rescoring process indicate that *Stack* is better at incorporating language-level information into the acoustic model.

4.4 Conclusion

We have proposed an utterance-wise recurrent dropout method and an iterative speaker adaptation technique for robust monaural speech recognition. Each of the proposed methods yields a substantial improvement on the monaural track of the CHiME-4 corpus. The WER of our best model is 8.28%, outperforming the previous best system by 16.19% relatively. Future directions on robust monaural speech recognition include adding speech separation frontends, upgrading the components of CLDNN acoustic models, designing better decoders for utterance-wise trained acoustic models, and boosting the performance of end-to-end systems on small corpora.

5 MASKING BASED BEAMFORMING AND MULTI-CHANNEL ASR

Modern electronic devices normally contain multiple microphones for speech applications. Acoustic beamforming techniques based on microphone arrays have shown to be quite beneficial

for robust ASR [34]. With multiple microphones, spatial information can be exploited and corrupted signals can be reconstructed with high noise reduction and at the same time with low speech distortion, only if underlying acoustic transfer functions can be accurately estimated [6] [49]. Conventionally, acoustic transfer functions are estimated from direction of arrival (DOA) estimation and the knowledge of microphone geometry. Recently, acoustic beamforming algorithms based on T-F masking have demonstrated great potential in the CHiME-3 and CHiME-4 challenge [55] [72] [13] [75]. The key idea is to estimate a T-F mask from multichannel signals so that the spatial covariance matrices of speech and noise can be derived for beamforming. A main advantage of the T-F masking based approaches is versatility, as the learning machine only needs to learn how to estimate a T-F mask during training, which is a well-studied task in monaural speech enhancement, and the same learned model and algorithmic pipeline can be directly applied to microphone arrays with any number of microphones, without any knowledge of the underlying microphone geometry.

In this section, we present our beamforming approaches based on T-F masking and deep learning. The overall algorithmic pipeline is to first obtain a speech mask from multichannel signals, from which the speech covariance matrix and acoustic transfer function can be estimated and used for minimum variance distortionless response (MVDR) beamforming. The beamformed signals are then fed into backend acoustic models for decoding.

We first introduce the well-known MVDR beamformer in Section 3.1 and then present an Eigen decomposition based method for acoustic transfer function estimation in Section 3.2. Next, we present another new method for acoustic transfer function estimation, which is based on the STFT ratios of speech-dominant T-F units. Experimental setup and evaluation results are presented in 3.4. Conclusions are made in Section 3.5.

5.1 MVDR Beamforming

Assuming no or very low reverberation and only one fixed target source, the acoustic model in the STFT domain is formulated as

$$\boldsymbol{y}(t,f) = \boldsymbol{c}(f)s(t,f) + \boldsymbol{n}(t,f) \tag{15}$$

where (t, f) is the STFT vector of the received speech, $L(t, f)$ is the STFT value of the target speaker, and $\boldsymbol{n}(t, f)$ is the STFT vector of the received noise at a specific T-F unit. (f) is the so-called steering vector or acoustic transfer function between the target source and the microphones at every frequency channel.

The classical MVDR beamformer [17] finds a weight vector for every frequency band, (f), such that the target speech along the look direction is maintained, while the interference from other directions is suppressed. Mathematically,

$$\begin{aligned} \boldsymbol{w}^*(f) &= \underset{\boldsymbol{w}(f)}{argmin}\ \ \boldsymbol{w}(f)^H \Phi_n(f) \boldsymbol{w}(f) \\ s.t.\ \ \boldsymbol{w}(f)^H &\boldsymbol{c}(f) = 1 \end{aligned} \tag{16}$$

where $\Phi(f)$ is the noise covariance matrix and $(\cdot)^H$ stands for conjugate transpose. The close-form solution is given as

$$\boldsymbol{w}^*(f) = \frac{\Phi_n(f)^{-1}\boldsymbol{c}(f)}{\boldsymbol{c}(f)^H \Phi_n(f)^{-1}\boldsymbol{c}(f)} \tag{17}$$

The beamformed signal is then obtained as

$$\hat{y}(t,f) = \boldsymbol{w}^*(f)^H \boldsymbol{y}(t,f) \tag{18}$$

As we can see, the key for MVDR beamforming is the accurate estimation of $\boldsymbol{c}\ (f)$ and $\Phi_n\ (f)$. In this project, we use DNN based supervised T-F masking to estimate them.

5.2 DNN-Based Eigen-Beamforming

The masking based approach [72] calculates the spatial covariance matrix of target speech and uses its principal eigenvector as the estimated steering vector. In this study, we follow this approach but with extensions. The major novelty here is that we estimate the masks using DNNs in a supervised way, rather than the spatial clustering approaches as [72]. The motivation is that many studies have shown the effectiveness of DNN based T-F masking for monaural enhancement [60] [59] [63] and robust ASR tasks [3] [61]. It is likely that the masks estimated in this way lead to better steering vector estimation and beamforming.

The first step of the proposed algorithm is to estimate one speech mask from each of DD microphone signals. Then these DD masks are condensed into one single mask by performing max pooling at each T-F unit as in Eq. (19). The condensed mask essentially represents the portion of speech energy at each T-F unit, and can be utilized to compute the noise covariance matrix, $\widehat{\Phi}_n(f)$, as in Eq. (20). We use the complement of the speech mask to obtain the noise mask. The speech covariance matrix, $\widehat{\Phi}_s(f)$, is computed as the difference between the noisy speech covariance matrix, $\widehat{\Phi}_y(f)$, and the noise covariance matrix, $\widehat{\Phi}_n(f)$. Note that we compute these parameters for each frequency band.

$$\widehat{M} = max\left(\widehat{M}_1, \ldots, \widehat{M}_D\right) \tag{19}$$

$$\widehat{\Phi}_n(f) = \frac{\sum_{t=1}^{T}(1-\widehat{M}(t,f))\boldsymbol{y}(t,f)\boldsymbol{y}(t,f)^H}{\sum_{t=1}^{T}(1-\widehat{M}(t,f))} \tag{20}$$

$$\widehat{\Phi}_y(f) = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}(t,f)\boldsymbol{y}(t,f)^H \tag{21}$$

$$\widehat{\Phi}_s(f) = \widehat{\Phi}_y(f) - \widehat{\Phi}_n(f) \tag{22}$$

$$\hat{\boldsymbol{c}}(f) = \mathcal{P}\{\widehat{\Phi}_s(f)\} \tag{23}$$

With the covariance matrices computed, the acoustic transfer function, $\hat{c}(f)$, is estimated as the principal eigenvector of $\widehat{\Phi}_s(f)$ as in Eq. (23). The rationale is that if the noise can be

removed, the oracle speech covariance matrix itself would be a symmetric rank-one matrix. In such a case, its principal eigenvector would be (f) [20]. Here we emphasize that we use the subtraction in Eq. (22) to obtain the estimate $\widehat{\Phi}_s(f)$. We could actually directly compute $\widehat{\Phi}_s(f)$ using the estimated mask as is done in the calculation of $\widehat{\Phi}_n(f)$, however, the noise covariance obtained via weighted pooling would be more accurate, as there are normally many frames containing only noises, i.e. speech silence frames, which would be easily detected by the DNN. Therefore, the speech covariance matrix can be better estimated by using the subtraction, assuming the un- correlatedness between speech and noise signals.

With Eqs. (20) and (23), an MVDR beamformer can be derived for enhancement using (17). After enhancement results are obtained using Eq. (18), log Mel filterbank features are extracted and fed into acoustic models for ASR decoding.

5.3 Relative Transfer Function Estimation via STFT Ratios

In the previous subsection, we introduced an acoustic transfer function estimate that is based on Eigen decomposition of speech covariance matrices. This section proposes a conceptually much simpler and mathematically less involved algorithm for relative transfer function (RTF) estimation. The proposed algorithm is based on the direct utilization of speech-dominated T-F unit pairs, without computing covariance matrices, Eigen decomposition, or estimating gains or time delays, thus making fewer assumptions. Intuitively, for a T-F unit pair where both T-F units are dominated by speech, the RTF can be reasonably estimated as the ratio of the two STFT coefficients. Note that by a T-F unit pair, we mean the two corresponding T-F units between the signals of the two recording channels (microphones).

Due to auditory masking and signal sparsity in the time-frequency domain, only one source would dominate for most of the T-F units [56]. In such cases, for a speech-dominated unit pair,

the noise level would be low that the physical model becomes

$$\boldsymbol{y}(t,f) = \boldsymbol{c}(f)s(t,f) + \boldsymbol{\varepsilon} \tag{24}$$

where $\boldsymbol{\varepsilon}$ represents a negligibly small term. In such a case, the RTF with respect to a reference microphone at a specific T-F unit, $\bar{c}(t,f)$, can be estimated as

$$\bar{\boldsymbol{c}}(t,f) = \frac{\boldsymbol{c}(f)}{c^{ref}(f)} = \frac{\boldsymbol{c}(f)s(t,f)}{c^{ref}(f)s(t,f)} \approx \frac{\boldsymbol{y}(t,f)}{y^{ref}(t,f)} \tag{25}$$

To improve robustness, we first normalize $\bar{c}(t,f)$ to unit length and then perform weighted pooling within each frequency channel to obtain $\bar{c}(f)$, i.e.

$$\bar{\boldsymbol{c}}(t,f) = \frac{\bar{\boldsymbol{c}}(t,f)}{\|\bar{\boldsymbol{c}}(t,f)\|} \tag{26}$$

$$\bar{\boldsymbol{c}}(f) = \frac{\sum_t \eta(t,f)\bar{\boldsymbol{c}}(t,f)}{\sum_t \eta(t,f)} \tag{27}$$

where (t, f) is the weights denoting the importance of the unit pair. It is defined as

$$\eta(t,f) = \prod_{i=1}^{D} 1[\hat{M}_i(t,f) > \theta](\hat{M}_i(t,f) - \theta) \tag{28}$$

Where $\overline{M_\iota}$ is the estimated mask representing the speech portion for the signal at microphone i, θ is a manually set threshold to filter out unreliable unit pairs.

We emphasize that the normalization in Eq. (26) leads to a more robust estimate of $\bar{c}(f)$, as it removes the influence of diverse energy levels at different T-F units, and additionally reduces the effects due to microphone failures. Eq. (28) means that only the unit pairs with both T-F units strongly dominated by speech should be considered for RTF estimation and the higher the values are in the estimated masks, the more weights are placed. Obviously, we need to estimate a mask for the target signal at every microphone.

Finally, we normalize $\bar{c}(f)$ to have unit length to get the estimated RTF for MVDR beamforming:

$$\hat{c}(f) = \frac{\bar{c}(f)}{\sqrt{\bar{c}(f)^H \bar{c}(f)}} \tag{29}$$

It should be noted that, for the frequency channels with no speech-dominated unit pairs, the noisy speech at the reference microphone is used as the output directly. We summate over all the values in each estimated mask and choose the microphone with the largest summation as the reference microphone. The rationale is that for the signal with the highest input SNR, the largest percentage of energy would normally be retained by our DNN based mask estimator.

Following [72] [23] and Eq. (20), the noise covariance matrix is estimated as

$$\hat{\Phi}_n(f) = \frac{\sum_t \xi(t,f)\boldsymbol{y}(t,f)\,\boldsymbol{y}(t,f)^H}{\sum_t \xi(t,f)} \tag{30}$$

where $\xi(t,f)$ is the weight representing the importance of the T-F unit pair for noise covariance matrix estimation. In our study, it is defined as

$$\xi(t,f) = \prod_{i=1}^{D} 1[1 - \hat{M}_i(t,f) > \gamma](1 - \hat{M}_i(t,f) - \gamma) \tag{31}$$

where γ is a tunable threshold to select noise-dominated unit pairs for noise covariance matrix computation. We obtain the noise portion as the complement of the speech portion.

With Eq. (29) and (30), an MVDR beamformer can be derived for enhancement using Eq. (17). After enhancement results are obtained using (18), log Mel filterbank features are extracted and fed into acoustic models for decoding.

Our approach makes fewer assumptions than other beamformers. Compared with traditional TDOA (time delay of arrival) estimation approaches or the weighted delay-and-sum (WDAS) beamformer [28] [5], our approach estimates directly a complex and continuous gain rather than

separately estimates a time delay and a gain for steering vector derivation. In addition, our approach does not require any knowledge of microphone geometry, or compute speech covariance matrices. There is no Eigen decomposition or post-filtering involved. Therefore the amount of computation in our approach is considerably less compared with the GEV beamformer [23] [24] or the MVDR beamformer [75].

5.4 Results and Discussion

Our experiments are conducted on the two-channel and six-channel task of the CHiME-4 challenge [55]. The six-channel CHiME-4 dataset re-uses the data in WSJ0-5k and CHiME-3, and features one-, two-, and six-channel tasks. Note that CHiME-3 and CHiME-4 use exactly the same data. The difference is that CHiME-3 allows participants to use all the microphone channels for testing, while CHiME-4 restricts the number of microphone channels to be used for testing. The six microphones are mounted on a tablet, with the second one in the rear and the other five in the front. It incorporates simulated utterances and real recordings from four daily environments, i.e. bus, pedestrian area, cafe, and street, exhibiting significant training and testing mismatches in terms of speaker, noise and spatial characteristics (see Sect. 2.2.1). The training data contains 7,138 simulated and 1,600 real utterances, the development set consists of 1,640 simulated and 1,640 real utterances, and the test set includes 1,320 simulated and 1,320 real utterances. Each of the three real subsets is recorded from four different speakers. For the two-channel task, only the signals from randomly selected two of the front five channels are provided in the development and test set. As a result, there are multiple microphone geometries underlying the test data for the two-channel task.

Our acoustic model is trained on all the noisy signals from all the six microphones, i.e. 7,138*6+1,600*5 utterances (~104h). We did not use the second microphone in the real training set as it is much more corrupted than the other five. We follow the common pipelines in the Kaldi toolkit [44] to build our ASR systems, i.e. GMM-HMM training, DNN training, sMBR training, language model (LM) rescoring, and speaker adaptation. Our DNN based acoustic model has seven hidden layers, each with 2,048 exponential linear units. There are 3,161 senone states in our system. The input feature is 40-dimensional log Mel filterbank feature with its deltas and double deltas, and an 11-frame symmetric context window. We perform sentence level mean-variance normalization before global mean-variance normalization. The dropout rates are set to 0.3. Batch normalization [31] and AdaGrad are utilized to speed up training. To compare

Table 8 Comparison of the ASR performance (%WER) with other systems on the CHiME-3 dataset

Approaches	Dev. set		Test set	
	SIMU	REAL	SIMU	REAL
Proposed deep eigen-beamformer + sMBR and trigram LM	6.38	5.64	7.03	7.60
+Five-gram LM and RNNLM	4.60	3.75	5.25	5.53
+Unsupervised speaker adaptation	**2.98**	**2.81**	**3.26**	**3.70**
Higuchi *et al*. [26]	3.63	3.45	4.46	5.83
Heymann *et al*. (https://github.com/fgnt/nn-gev)	5.01	4.53	5.60	7.45

our overall ASR system with other systems, we apply the challenge-standard five-gram language model and the RNN language model for lattice rescoring. In addition, we apply the unsupervised speaker adaptation algorithm proposed in our recent study [64] for run-time adaptation. We use WER on the real utterances of the development set for parameter tuning, as the CHiME-4 challenge ranks all the submitted systems according to the performance on the real test set only.

The DNN for mask estimation is trained using all the 7,138x6 simulated utterances (~90h) in the training set. It has four hidden layers, each with 2,048 exponential linear units. Sigmoidal units are used in the output layer as the IRM is naturally bounded between zero and one. The log

power spectrogram features are mean normalized at the sentence level before global mean-variance normalization. We symmetrically splice 19 frames as the input to the DNN. The dropout rates are set to 0.1. The window length is 25 ms and the hop size is 10 ms. Pre-emphasis and Hamming windowing are applied before performing 512-point FFT. The input dimension is therefore 257x19, and the output dimension is 257. For the two-channel task, θ in Eq. (28) is set to 0.5, which means that the speech energy should be at least the same as the noise energy for a speech-dominant T-F unit, and γ in Eq. (31) is set to 0.5 as well, meaning that the noise energy should be larger than the speech energy for a noise-dominant T-F unit. For the six-channel task, θ and γ are both set to zero.

In the following sections, we first report DNN based Eigen-beamforming results on the six-channel CHiME-3 dataset, and then present the performance of the STFT ratio based RTF estimation approach on the two-channel and six-channel track of the CHiME-4 dataset.

5.4.1 Results of Deep Eigen-Beamforming

The results on the six-channel CHiME-3 dataset are presented in Table 8. We first feed the beamformed speech into the acoustic model obtained after sequence training and a trigram language model for decoding, and then use the task-standard five-gram and RNN language model for lattice rescoring. The WER we obtained on the real test set is 5.53%, which is already better than the winning solution [72] of CHiME-3. By further performing speaker adaptation in our recent study [64], the WER is further pushed to 3.70% WER. Note that the system in [72] uses clustering for masking based beamforming. Their acoustic model is an advanced CNN with the "network in network" structure. Complicated cross-adaptation techniques are included in their system to deal with speaker variations. The system by Heymann *et al*. uses a BLSTM for IBM estimation and a generalized eigenvector beamformer for robust ASR. Their result is 7.45%

WER on the real test set. The results obtained by our system clearly demonstrate the effectiveness of the proposed beamformer and the overall ASR system.

5.4.2 Results of RTF Estimation based on STFT Ratios

We compare the performance of our system with several other beamformers on the six- and two-channel task of CHiME-4. The setup of each beamformer is detailed in Table 9. These beamformers have been previously applied to the CHiME-4 corpus and shown strong robustness. We use the acoustic model after sMBR training and the trigram language model for decoding. We emphasize that for all the masking based beamformers listed in Table 9, we use the same estimated masks from our DNN for a fair comparison.

Table 9 WER (%) comparison of different beamformers (sMBR training and tri-gram LM for decoding) on the six-channel track

Approaches	Covariance matrices	Beamforming weights	Post-filters	Dev. set		Test set	
				SIMU	REAL	SIMU	REAL
BeamformIt [1] [28]	None	See [42]	None	8.62	7.28	12.81	11.72
MVDR via SRP-PHAT [5]	$\hat{\Phi}_n(f)$ from 400-800ms context	$\hat{c}(f)$ via SRP-PHAT, see [43] $\hat{w}(f) = \frac{\hat{\Phi}_n(f)^{-1}\hat{c}(f)}{\hat{c}(f)^H\hat{\Phi}_n(f)^{-1}\hat{c}(f)}$	None	6.32	9.38	7.05	14.60
GEV beamformer [23] [24] [25]	$\hat{M} = median(\hat{M}_1, \ldots, \hat{M}_D)$ $\hat{\Phi}_n(f) = \frac{\sum_t(1-\hat{M}(t,f))\mathbf{y}(t,f)\mathbf{y}(t,f)^H}{\sum_t(1-\hat{M}(t,f))}$ $\hat{\Phi}_s(f) = \frac{\sum_t \hat{M}(t,f)\mathbf{y}(t,f)\mathbf{y}(t,f)^H}{\sum_t \hat{M}(t,f)}$	$\hat{w}(f) = \mathcal{P}\{\hat{\Phi}_n(f)^{-1}\hat{\Phi}_s(f)\}$ $\mathcal{P}\{\cdot\}$ – principal eigenvector	$g_{BAN}(f) = \frac{\sqrt{\hat{w}(f)^H\hat{\Phi}_n(f)\hat{\Phi}_n(f)\hat{w}(f)/D}}{\hat{w}(f)\hat{\Phi}_n(f)\hat{w}(f)}$	5.79	5.84	6.70	7.97
PMWF-0 [16] [14] [70]	Same as above	$\hat{w}(f) = \frac{\hat{\Phi}_n(f)^{-1}\hat{\Phi}_y(f)}{trace(\hat{\Phi}_n(f)^{-1}\hat{\Phi}_y(f))}\mathbf{u}_r$	None	6.05	5.86	8.04	8.43
MVDR via eigendecomposition I	Same as above	$\hat{c}(f) = \mathcal{P}\{\hat{\Phi}_s(f)\}$ $\hat{w}(f) = \frac{\hat{\Phi}_n(f)^{-1}\hat{c}(f)}{\hat{c}(f)^H\hat{\Phi}_n(f)^{-1}\hat{c}(f)}$	None	5.91	5.62	7.20	8.30
MVDR via eigendecomposition II [75]	$\hat{M}, \hat{\Phi}_n(f)$ as above $\hat{\Phi}_{sn}(f) = \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}(t,f)\mathbf{y}(t,f)^H$ $\hat{\Phi}_s(f) = \hat{\Phi}_{sn}(f) - \hat{\Phi}_n(f)$	Same as above	None	6.13	5.65	6.98	8.07
Proposed	$\hat{\Phi}_n(f)$ as above	See Section 3.2 (not using Eq. (26))	None	5.65	5.49	6.44	7.89
	$\hat{\Phi}_n(f)$ as in Eq. (30) and (31)	See Section 3.2 (not using Eq. (26))	None	5.65	5.45	6.40	7.68
	Same as above	See Section 3.2 (using Eq. (26))	None	5.64	**5.40**	6.23	**7.30**

Table 10 WER (%) Comparison with other systems (using the constrained RNNLM for decoding) on the six-channel track

Approaches	Dev. set		Test set	
	SIMU	REAL	SIMU	REAL
Proposed beamformer + sMBR and tri-gram LM	5.64	5.40	6.23	7.30
+Five-gram LM and RNNLM	3.77	3.43	4.46	5.24
+Unsupervised speaker adaptation	2.69	2.70	3.09	3.65
Du *et al.* [13] (with model ensemble)	2.61	2.55	3.06	3.24
Best single model of [13]	-	2.88	-	3.87

Heymann *et al*. [48]	2.75	2.84	3.11	3.85

The BeamformIt represents the official WDAS beamformer implemented using the BeamformIt toolkit [1] [28]. It uses the GCC-PHAT algorithm for time delay estimation and the cross-correlation function for gain estimation in a segment-by-segment fashion. Then, the time-domain signals are delayed, scaled and summed together. It is a strong representative baseline of conventional TDOA approaches for beamforming. The MVDR via SRP-PHAT algorithm [5] is another official baseline provided in the challenge. It uses the conventional SRP-PHAT algorithm for DOA estimation. The initial estimated time delays at each time frame are further regularized by the Viterbi algorithm to enforce the assumption that the sound source position is approximately in the front. The gains are assumed to be equal across different microphone channels. With these two, a steering vector is derived for MVDR beamforming. The noise covariance matrix is estimated from 400-800ms context immediately before each utterance. Note that the simulated data in the CHiME-4 challenge is created by the same sound localizer, therefore this approach performs quite well on the simulated data, while much worse on the real data. For the GEV beamformer, following the original algorithms [23] [24] [25], we combine the two estimated masks using median pooling before computing the speech and noise covariance matrices. After that, generalized Eigen decomposition is performed to obtain beamforming weights. A post-filter based on blind analytic normalization is further appended to reduce speech distortions. The PMWF-0 approach [49] uses matrix operations on speech and noise covariance matrices to compute the weights. It is later combined with T-F masking based approaches in [16] [14] [70], where u_f is a one-hot vector denoting the index of the reference microphone. For the MVDR via Eigen decomposition I, we use the principal eigenvector of the speech covariance matrix as the estimate of the steering vector, assuming that the speech covariance matrix is a

rank-one matrix, although this assumption may not hold when there is room reverberation, e.g. in the bus or cafeteria environment. In MVDR via Eigen decomposition II, we follow the algorithm for covariance matrix calculation proposed in [72] [75], where the speech covariance matrix is obtained by subtracting the noise covariance matrix from the covariance matrix of noisy speech. As we can see from the last entry of Table 9, our approach consistently outperforms the alternative approaches in all the simulated and real subsets, especially on the real test set. Another comparison is provided in the first two entries of the proposed beamformer in Table 9, where we use the same noise covariance matrix as in the other beamformers together with the proposed RTF estimation for MVDR beamforming. We can see that using Eqs. (30) and (31) to estimate the noise covariance matrix leads to a slight improvement (from 7.89% to 7.68% WER). In the last entry of the proposed beamformer, we use Eq. (26) to normalize $y(t,f) \,/\, y^{\mathrm{ref}} f\ (t, f)$ before weighted pooling. Consistent improvement has been observed (from 7.68% to 7.30% WER). This is likely because of the normalization of diverse energy levels, and better handling of extremely large or small ratios caused by microphone failures.

We then use the task-standard language models to re-score the lattices, and perform run-time unsupervised speaker adaptation [64]. The results are reported in Table 10. The best result we have obtained on the real test set is 3.65% WER. We compare our results with the results from other systems, which are obtained using the same constrained RNNLM for decoding[1]. The winning system by Du *et al.* [13] obtains 3.24% WER on the real test set, and their overall system is an ensemble of multiple DNN and deep CNN based acoustic models trained from augmented training data. Their best single model trained on the augmented training data obtains

[1] *See http://spandh.dcs.shef.ac.uk/chime_challenge/results.html for the ranking of all the results obtained when using the baseline RNNLM for decoding. Note that all the teams in the challenge were requested to report the decoding results using the official RNNLM.*

3.87% WER on the real test set (according to the Table 3 of [13]). In their solution, they combine a clustering method, a DNN based method, and the feedbacks from backend ASR systems for mask estimation. The RTF is obtained via Eigen decomposition. Then, a general side lobe canceller with post-filtering is constructed for beamforming. The runner-up system by Heymann *et al.* [25] utilizes WRBN for acoustic modeling (see Sect. 2) and a BLSTM model for GEV beamforming. Their best result is 3.85% WER on the real test set. We emphasize that our system uses simple DNNs for both mask estimation and acoustic modeling, and we do not use any data augmentation, such as adding the beamformed signals for acoustic modeling, or any model or system ensemble, as we aim for a simple algorithm for RTF estimation. The results presented here clearly demonstrate the effectiveness of the proposed algorithm and our overall ASR system.

For the two-microphone task, the RTF at each T-F unit is estimated as the ratio between a signal and the corresponding reference signal as in Eq. (25). The ASR results on the two-channel task are reported in Table 11 and Table 12. In Table 11, we use the same mask estimator for beamforming, and the same acoustic model after sequence training and the tri-gram language model for decoding. The results in each entry of Table 11 are obtained using the same algorithm detailed in the corresponding entry of Table 9. The only difference is D is set to two now. For all the matrix inversions, we use the close-form solution of two-by-two matrices to avoid numeric issues. Similar trends as in Table 9 and 10 are observed, indicating that our approach also performs well in the two-microphone case. Nonetheless, the relative improvement over other beamformers is slightly smaller than in the six-channel task. Finally, we apply language model re-scoring and speaker adaptation to our system. The results are presented in Table 12. Similar trends to Table 10 are observed.

Table 11 WER (%) comparison of different beamformers (using sMBR training and tri-gram LM for decoding) on the two-channel track

Approach	Dev. set		Test set	
	SIMU	REAL	SIMU	REAL
BeamformIt	10.56	8.68	15.83	15.30
MVDR via SRP-PHAT	9.22	9.52	11.37	16.29
GEV beamformer	9.09	7.64	10.97	12.55
PMWF-0	9.00	7.66	12.33	12.85
MVDR via eigendecomposition I	9.19	7.66	11.69	12.47
MVDR via eigendecomposition II	9.05	7.50	10.79	12.42
Proposed beamformer	8.90	7.32	10.58	12.00
	8.74	**7.29**	10.50	11.84
	8.75	7.32	10.36	**11.81**

Table 12 WER (%) comparison with other systems (using the constrained RNNLM for decoding) on the two-channel track

Approach	Dev. set		Test set	
	SIMU	REAL	SIMU	REAL
Proposed beamformer + sMBR and tri-gram LM	8.74	7.29	10.50	11.84
+Five-gram LM and RNNLM	6.60	4.98	7.77	8.81
+Unsupervised speaker adaptation	4.95	3.84	5.60	6.10
Du *et al*. [13] (with model ensemble)	4.89	3.56	7.30	5.41
Best single model of	-	4.05	-	6.87
Heymann *et al*. [25]	4.45	3.8	5.38	6.44

5.5 Conclusion

We have proposed two novel methods for RTF estimation, which are based on Eigen decomposition and STFT ratios weighted by a T-F mask. Deep learning based time- frequency masking plays an essential role in the accurate estimation of the statistics for MVDR beamforming. Large improvements have been observed on the CHiME-3 and CHiME-4 datasets in terms of ASR performance. Although mathematically and conceptually much simpler, the proposed approach using mask-weighted STFT ratios has shown consistent improvement over competitive methods on both the six- and two-channel tasks of the CHiME-4 challenge.

Masking based beamforming approaches rely heavily on the availability of speech-dominant T-F units, where phase information is not much contaminated. In daily recorded utterances, the

number of such T-F units is commonly sufficient for RTF estimation, and DNN performs well at identifying them, even with just energy features. Future research will analyze and improve the performance in very noisy and highly reverberant environments.

6 SPATIAL FEATURES FOR T-F MASKING AND ROBUST ASR

In the previous section, DNNs rely only on single-channel spectral information to estimate the IRM from every microphone signal. The independently estimated masks are then combined into a single mask, which is used to weight spatial covariance matrices for beamforming, as detailed in Section 3.2. An advantage of using single-channel information for T-F masking is that the DNN model trained this way is applicable regardless of the number of microphones and microphone geometry.

Different from these studies, we incorporate spatial features as additional inputs for model training in order to complement the spectral information for more accurate mask estimation. Through pooling spatial features over microphone pairs, the applicability of the proposed approach is also not impacted by the number of microphones and microphone geometry.

A key observation motivating the study in this section is that a real-world auditory scene is usually comprised of one directional target speaker, a number of directional interfering sources, and diffuse noise or room reverberation coming from various directions. To distinguish the directional target source from the other directional sources, robust speaker localization is needed to determine the direction that contains the target speech. If the target direction is known, directional features indicating whether the signal at each T-F unit is from that direction can be utilized to extract the target speech from that direction, and filter out the interference and reverberation from other directions. In addition, diffuse noises and reflections caused by room reverberation reach microphones from various directions. This property can be exploited to

derive inter-channel coherence based features to indicate whether a T-F unit is dominated by a directional source. We emphasize that spectral information is crucial for suppressing noise or reverberation coming from directions around the target direction. To take all these considerations into account, we simply encode them as discriminative input features for mask estimation. This way, complementary spectral and spatial information are utilized to boost speech separation.

Previous efforts employ directional features for DNN based mask estimation. Most of the earlier studies assume that the target speech comes from a fixed direction, typically the front direction in a binaural setup. In [32], interaural time differences (ITD), interaural level differences (ILD) and entire cross-correlation coefficients are used as primary features for sub-band IBM estimation in the cochleagram domain. Subsequently, Zhang and Wang [74] propose to combine ITD, ILD, and spectral features derived from a fixed beamformer for mask estimation. Although these approaches show good performance when the target is in the front, they unlikely perform well when target speech is from other directions. Other studies perform single-channel post-filtering or spatial filtering on beamforming outputs for further noise reduction [52] [42] [8]. For coherence-based features, previous attempts [40] [4] in robust ASR are mainly focused on using them as post-filters for beamforming. Different from the previous studies, we incorporate spatial and spectral features as extra input for DNN based mask estimation. This way, DNN can exploit the complementary nature of spectral and spatial information, leading to better mask estimation and subsequent covariance matrices estimation. This in turn results in better beamforming and robust ASR performance.

The following subsections present two spatial features for better mask estimation. The diffuse feature is designed to suppress diffuse noises and the directional feature is designed to suppress interference sources from nontarget directions. An example of the diffuse and

directional feature is shown in Figure 4. As can be seen from Figs. 4(c) and 4(d), they are both well correlated with the IRM depicted in Fig. 4(b).

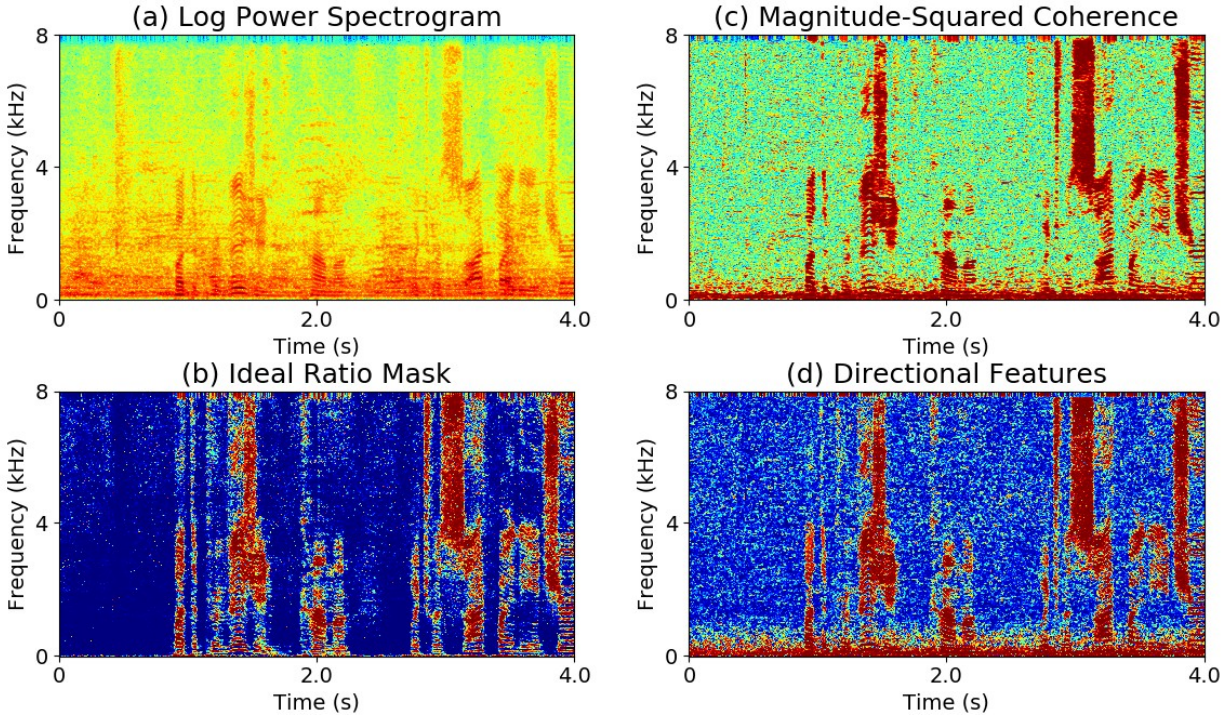

Figure 4 Illustration of the spectral and spatial features using a simulated utterance in the CHiME-4 dataset. (a) and (b) are obtained using the first channel, and (c) and (d) are computed using all the six microphone signals. In (d), the ideal ratio mask is us

6.1 Magnitude Squared Coherence

If a T-F unit pair is dominated by directional target speech, the unit responses would be coherent. Similarly, for a unit pair dominated by diffuse noises or room reverberations, their responses would be incoherent. Hence the coherence can be utilized as spatial features to differentiate directional and non-directional sources. Our study employs the magnitude squared coherence (MSC) as additional features for DNN based T-F masking.

To compute the MSC features, we first calculate the spatial covariance matrix of the noisy speech $\widehat{\Phi}_y(t,f)$ as

$$\widehat{\Phi}_y(t,f) = \frac{1}{2w+1}\sum_{t'=t-w}^{t+w} \boldsymbol{y}(t',f)\boldsymbol{y}(t',f)^H \tag{32}$$

where w represents the half-window length. Then we calculate the inter-channel coherence (ICC) between microphone a and j using

$$\mathrm{ICC}(i,j,t,f) = \frac{\widehat{\Phi}_y(t,f,i,j)}{\sqrt{\widehat{\Phi}_y(t,f,i,i)}\sqrt{\widehat{\Phi}_y(t,f,j,j)}} \tag{33}$$

Finally, we pool over the ICCs of all the microphone pairs to obtain the MSC features:

$$\mathrm{MSC}(t,f) = \frac{1}{P}\sum_{i=1}^{D}\sum_{j=i+1}^{D} |\mathrm{ICC}(i,j,t,f)| \tag{34}$$

where $P = (D-1)/2$ is the total number of microphone pairs and $|\cdot|$ extracts the magnitude. Note that the pooling operation here is a straightforward way to combine multiple microphone signals, and would significantly improve the quality of MSC features.

Intuitively, if a T-F unit is dominated by a directional source across all the microphone channels, $\mathrm{ICC}(i, j, t, f)$ would be approximately equal to $e^{-\imath 2\pi \frac{f}{N} f_s \tau_{i,j}}$ where $\tau_{i,j}$ is the underlying time delay between microphone signal i and j, f_s is the sampling rate, $\imath$ is the imaginary unit, and N is the number of DFTs. The resulting $\mathrm{MSC}(t, f)$ would therefore be close to one after the absolute operation. In contrast, if the T-F unit is dominated by diffuse noises or room reverberations, $\mathrm{ICC}(i, j, t, f)$ would be close to a *sinc* function [20] defined as $\sin\left(2\pi \frac{f}{n} f_s \frac{d_{i,j}}{c_s}\right)/(2\pi \frac{f}{n} f_s \frac{d_{i,j}}{c_s})$, where $d_{i,j}$ is the spacing between microphone i and j, and c_s is the sound speed in air. It would be close to zero in high-frequency bands or when the microphone distance is large. The ww in Eq. (32) is simply set to one in our study, as increasing it would make the MSC feature smoother and become less discriminative for mask estimation. An example is illustrated in Fig. 4(c). At low frequencies, the MSC feature is not very useful, while it is very discriminative to the IRM at high frequencies.

We use MSC features as extra inputs to our neural networks for mask estimation. It should be

emphasized that interference could also be a directional source. Therefore it is beneficial to combine MSC features and spectral features as well as phase features introduced later for mask estimation. One favorable property of the MSC feature is that it is derived from noisy signals directly. Our study utilizes the MSC feature for time-frequency masking. This approach leverages the learning power of DNN to improve mask estimation, and therefore benefits later beamforming.

6.2 Direction-Invariant Directional Features

Suppose that the true time delay between two microphone signals is known in advance, the observed phase difference at each T-F unit pair should be aligned with the time delay if the unit pair is speech dominant. Based on this observation, the difference between the observed phase difference and the hypothesized phase difference is indicative of whether the unit pair is dominated by the speech from the hypothesized direction, or noises and inferences from the other directions [42] [40]. More specifically, we use the following equation to derive the directional features for model training.

$$DF(t,f) = \frac{1}{P}\sum_{i=1}^{D}\sum_{j=i+1}^{D}\cos\left(\angle y_i(t,f) - \angle y_j(t,f) - \frac{2\pi f}{N} f_s \hat{\tau}_{i,j}\right) \tag{35}$$

where $\angle y_i(t,f) - \angle y_j(t,f)$ stands for the observed phase difference between microphone signal i and j at a T-F unit pair, and $\frac{2\pi f}{N} f_s \hat{\tau}_{i,j}$ is the hypothesized difference given the estimated time delay $\hat{\tau}_{i,j}$ in seconds. The 2π-periodic cosine operation properly deals with potential phase-wrapping effects. If the time delay $\hat{\boldsymbol{\tau}}_{i,j}$ is accurately estimated, the resulting feature would be close to one for speech-dominant unit pairs, and much smaller than one for noise-dominant pairs. Note that when there are more than two microphones ($D > 2$), we simply pool all the microphone pairs to get the final feature. This strategy is found to improve the quality of spatial feature

extraction.

Although recent studies suggested that TODA can be robustly estimated using time-frequency masking [43], our study does not explicitly estimate TDOAs. Instead, we use the estimated steering vector from the MVDR beamformer to derive spatial features, as the steering vector itself contains all the information about time delays and gain differences. This strategy removes the need for a separate sound localization module and thus simplifies the system. In addition, it avoids the linear phase and planar wave assumption, which may not hold in practice. Mathematically, the spatial feature is computed as follows:

$$DF(t,f) = \frac{1}{P}\sum_{i=1}^{D}\sum_{j=i+1}^{D}\cos\left\{\angle y_i(t,f) - \angle y_j(t,f) - \left(\angle \hat{c}_i(f) - \angle \hat{c}_j(f)\right)\right\} \tag{36}$$

where $\angle \hat{c}_i(f)$ is the phase term extracted from the estimated steering vector, and therefore $\angle \hat{c}_i(f) - \angle \hat{c}_j(f)$ represents the estimated phase difference at the frequency f of microphone i and j. Essentially, Eq. (36) measures whether the signal is from the estimated location. By using spatial features for DNN training, we can extract the signal from the estimated target direction.

Previous efforts have applied directional features for DNN training. Their directional features however are mainly designed for fixed target directions, and therefore are not invariant to target directions. In [2], the target speaker is assumed to be in the front, so the phase difference for T-F unit pairs dominated by the target speech should be close to zero $\cos(\angle \hat{y}_i(t,f) - \angle \hat{y}_j(t,f)$ and is directly used as the features to build an auto-encoder based speech enhancement system. Different from these studies, the features derived in this study are location-invariant. The invariance is achieved by subtracting the estimated phase difference from the observed phase difference so that a high value in the derived directional feature of a unit pair always indicates that the pair is dominated by target speech.

As can be seen, the directional features in Eq. (36) need an accurate estimation of the steering

vector, $\hat{c}(f)$, to yield high-quality and discriminative features. We use the principal eigenvector of the estimated speech covariance matrix as the steering vector estimate. This is a proven strategy for accurate steering vector estimation [72] [75].

6.3 Results and Discussion

We evaluate our algorithms on the six-channel task of the CHiME-4 dataset. The details of this dataset and our backend ASR system have been described in Section 3.4.

Table 13 WER (%) comparison with other approaches on the six-channel track

Approach	Dev. set		Test set	
	SIMU	REAL	SIMU	REAL
BeamformIt	8.62	7.28	12.81	11.72
MVDR via SRP-PHAT	6.32	9.38	7.05	14.60
MSC as the Estimated Mask (no training)	6.49	6.16	9.77	9.91
Log Power Spectrogram	5.67	5.16	6.09	7.28
Log Power Spectrogram + MSC	5.63	5.08	6.31	6.92
Log Power Spectrogram + DF	5.82	5.06	6.49	6.70
+Five-gram LM and RNNLM	3.90	3.11	4.33	4.54
+Unsupervised speaker adaptation [64]	2.83	**2.54**	3.11	**3.08**
Du *et al*. [13] (with model ensemble)	2.61	2.55	3.06	3.24
Best single model of [13]	-	2.88	-	3.87
Heymann *et al*. [25]	2.75	2.84	3.11	3.85

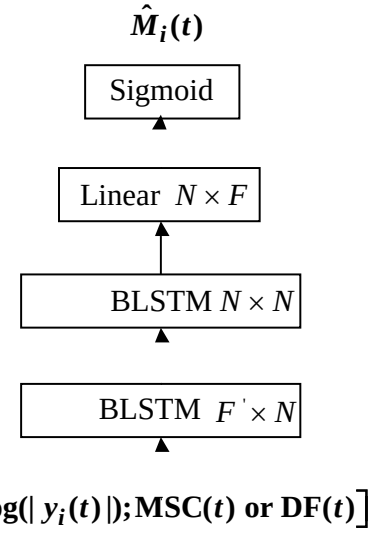

Figure 5 Network architecture for mask estimation.

Multiple BLSTMs taking in different features are trained for mask estimation using the 7,138*6 utterances (~90h) in the simulated training data of CHiME-4. The BLSTMs contain three hidden layers, each with 600 hidden units in each direction. Sigmoidal units are used in the output layer, as the IRM is naturally bounded between zero and one. The network architecture is depicted in Figure 5. The frame length is 32 ms and the shift is 8 ms. After Hamming windowing, 512-point FFT is performed to extract 257-dimensional log power spectrogram features for BLSTM training. No pre-emphasis is applied. We apply 0.1 dropout rate to the output of each BLSTM layer. Sentence-level mean normalization is performed on the spectral features to deal with channel mismatches and reverberations, while no sentence-level normalization is performed on spatial features. All of the features are globally normalized to zero mean and unit variance before being fed into the network. During training, we use the ideal speech covariance matrix computed directly from clean speech to derive $\hat{\mathrm{c}}(f)$ in Eq. (36). At runtime, we use the model trained using the log power spectrogram feature together with the MSC feature to get an estimate of $\hat{\mathrm{c}}(f)$. When using spatial features for training, we found it helpful to initialize the corresponding parts of the network using a well-trained model built by only using the log power spectrogram features, likely because spectral information itself is very important for mask estimation.

The ASR results are presented in Table 13, where we use our DNN based acoustic model after sMBR training for decoding, and the task-standard tri-gram language model for decoding unless specified otherwise. For baseline methods, see Section 3.4.

As a comparison, we use the MSC feature, $\mathrm{MSC}(t,f)$, as the speech mask and $1-\mathrm{MSC}(t,f)$ as the noise mask, to construct an MVDR beamformer for enhancement and robust ASR. Note that the range of $\mathrm{MSC}(t,f)$ has been linearly mapped to $[0,1]$ within each utterance. Surprisingly, this

simple approach, which does not even require any training or spatial clustering, achieves 9.91% WER on the real test set. This is probably because the real noises recorded in the CHiME- 3 and 4 dataset are mostly diffuse noises. This makes sense as in practice the acoustic scene in a bus, cafeteria, pedestrian area, and on the street would contain noises or interferences from many directions, such as engine noises, background speakers, wind noises or room reverberations. Even if directional sources are present, they are typically much weaker than the target speaker when the SNR is not very low[2] or when the speaker-microphone distance is not very large. In such case, the speech covariance matrix computed via weighted pooling would still be dominated by the target speech.

Using the log power spectrogram feature to train a BLSTM to predict the IRM, we get to 7.28% WER. Adding MSC features for BLSTM training further pushes the performance to 6.92% WER. For the model trained with the log power spectrogram and directional features, we first use the model trained with the log power spectrogram and MSC features to get $\hat{\mathbf{c}}(f)$ and then use it to compute the directional features using Eq. (36). The result is further improved to 6.70% WER. The directional features yield better performance over the MSC features. This is expected as noises or inferences could also be directional. Note that after adding spatial features, the performance on the simulated data however becomes worse, although consistent improvement is observed on the real data. This is likely because of the specific data simulation procedure adopted in the CHiME-3 and 4 corpus, which uses the least mean square algorithm to estimate the speech and noise images from a far-field recording and its corresponding close-talk recording. This procedure would likely introduce artifacts in the simulated data, especially sensitive phase information that is important for spatial feature derivation.

[2] ASR systems tend not be used in very noisy environments

Using the task-standard five-gram and RNNLM language model for lattice re-scoring, the result is improved to 4.54% WER. Note that the system so far is fully speaker independent. Further applying our unsupervised speaker adaptation [64] improves the performance to 3.08% WER. This result is better than the 3.24% WER obtained in the winning solution of the CHiME-4 challenge by Du *et al*. [13]. As commented before, their acoustic model is a combination of one DNN-based acoustic model and four CNN-based acoustic models trained from augmented training data. The input feature is a combination of log Mel filterbank features, fMLLR features and i-vectors. Their T-F masking based MVDR beamformer is constructed using a complex GMM based spatial clustering algorithm [72], a DNN based IRM estimator, the silence frames determined by the backend ASR systems, and an iterative mask refinement strategy [53]. The runner-up system by Heymann *et al*. [25] uses a BLSTM to drive a T-F masking based generalized eigenvector beamformer [23], and WRBN for acoustic modeling. Input-level linear transform is performed on each test speaker for unsupervised speaker adaptation. Their best performance when using the task-standard RNNLM is 3.87% WER. Different from these state-of-the-art systems, our approach focuses on frontend beamforming. Even with a simple feed-forward DNN as the backend acoustic model, our system has shown better performance. This clearly demonstrates the advantage of the proposed beamforming algorithm.

6.4 Conclusion

We have proposed a novel approach to integrate spectral and spatial features to improve T-F masking based beamforming. A consistent improvement has been observed on the six-channel task of the CHiME-4 challenge. Although the computation of the directional features requires a separate localization-like procedure, our results indicate that directional and diffuse features

contain discriminative information for supervised mask estimation. Hence combining them with spectral features for DNN training should lead to better mask estimates. To further improve recognition performance, future research would use deep learning based post-filtering to achieve further noise reduction, as beamformed signals currently are directly fed into backend acoustic models for decoding.

7 REFERENCES

[1] X. Anguera and C. Wooters, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 15, pp. 2011–2022, 2007.

[2] S. Araki, *et al.*, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proceedings of ICASSP*, pp. 116-120, 2015.

[3] D. Bagchi, *et al.*, "Combining spectral feature mapping and multi-channel model-based source separation for noise-robust automatic speech recognition," in *Proceedings of ASRU*, pp. 71-75, 2015.

[4] H. Barfuss, C. Huemmer, A. Schwarz, and W. Kellermann, "Robust coherence-based spectral enhancement for speech recognition in adverse real-world environments," *Comp. Speech Lang.*, vol. 46, pp. 388–400, 2017.

[5] J. Barker, R. Marxer, E. Vincent, and A. Watanabe, "The third CHiME speech separation and recognition challenge: dataset, task and baselines," in *Proceedings of IEEE ASRU*, pp. 5210-5214, 2015.

[6] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing.* Berlin: Springer, 2008.

[7] C.M. Bishop, *Pattern recognition and machine learning.* New York: Springer, 2006.

[8] A. Brutti, A. Tsiami, A. Katsamanis, and P. Maragos, "A phase-based time-frequency masking for multi-channel speech enhancement in domestic environments," in *Proceedings of Interspeech*, pp. 2875–2879, 2014.

[9] J. Chen and D.L. Wang, "Long short-term memory for speaker generalization in supervised speech separation," *J. Acoust. Soc. Am.*, vol. 141, pp. 4705-4714, 2017.

[10] Z. Chen, S. Watanabe, H. Erdogan, and J. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Proceedings of Interspeech*, 2015.

[11] G. Cheng, *et al.*, "An exploration of dropout with LSTMs," in *Proceedings of Interspeech*, 2017.

[12] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep neural network learning by exponential linear units (ELUs)," in *Proceedings of International Conference on Learning Representations*, 2016.

[13] J. Du, Y.-H. Tu, J. Sun, and e. al., "The USTC-iFlyteck system for the CHiME4 challenge," in *Proceedings of the CHiME-4 Workshop*, 2016.

[14] H. Erdogan, T. Hayashi, J.R. Hershey, T. Hori, and C. Hori, "Multi-channel speech recognition: LSTMs all the way through," in *Proceedings of CHiME-4 Workshop*, 2016.

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

[16] H. Erdogan, J.R. Hershey, S. Watanabe, M. Mandel, and J.L. Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proceedings of Interspeech*, pp. 1981-1985, 2016.

[17] O.L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, pp. 926-935, 1972.

[18] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proceedings of NIPS*, pp. 1019–1027, 2016.

[19] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comp. Speech Lang.*, vol. 12, pp. 75-98, 1998.
[20] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multi-microphone speech enhancement and source separation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 25, pp. 692–730, 2017.
[21] K. Han, *et al.*, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 23, pp. 982-992, 2015.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, pp. 770–778, 2016.
[23] J. Heymann, L. Drude, and A. Chinaev, "BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge," in *Proceedings of ASRU*, pp. 444–451, 2015.
[24] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proceedings of ICASSP*, pp. 196-200, 2016.
[25] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual BLSTM network with discriminative speaker adaptation for robust speech recognition," in *Proceedings of CHiME-4 Workshop*, pp. 196-200, 2016.
[26] T. Higuchi, N. Ito, T. Yoshioka, and e. al., "Robust MVDR beamforming using time- frequency masks for online/offline ASR in noise," in *Proceedings of ICASSP*, pp. 5210- 5214, 2016.
[27] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhotdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580*, 2012.
[28] T. Hori, *et al.*, "The MERL/SRI system for the 3rd CHiME challenge using beamforming, robust feature extraction, and advanced speech recognition," in *Proceedings of ASRU*, pp. 475–481, 2015.
[29] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 23, pp. 2136-2147, 2015.
[30] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, pp. 1944–1957, 2013.
[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.
[32] Y. Jiang, D.L. Wang, R.S. Liu, and Z.M. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 22, pp. 2112-2121, 2014.
[33] K. Kinoshita, *et al.*, "A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP J. Adv. Sig. Proc.*, vol. 20162016.
[34] K. Kumatani, *et al.*, "Microphone array processing for distant speech recognition: towards real-world deployment," in *Proceedings of Annual Summit and Conference on Signal and Information Processing*, pp. 1-10, 2012.
[35] B. Li and K.C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *Proceedings of Interspeech*. 2010.
[36] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 22, pp. 745–777, 2014.

[37] T. Moon, H. Choi, H. Lee, and I. Song, "Rnndrop: A novel dropout for RNNs in ASR," in *Proceedings of ASRU*, pp. 65-70, 2015.
[38] A. Narayanan and D.L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proceedings of ICASSP*, pp. 7092-7096, 2013.
[39] A. Narayanan and D.L. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 22, pp. 826– 835, 2014.
[40] Z. Pang and F. Zhu, "Noise-robust ASR for the third CHiME challenge exploiting time-frequency masking based multi-channel speech enhancement and recurrent neural network," *arXiv:1509.07211*, 2015.
[41] G. Pereyra, Y. Zhang, and Y. Bengio, "Batch normalized recurrent neural networks," in *Proceedings of ICASSP*, pp. 2657–2661, 2016.
[42] P. Pertil and J. Nikunen, "Microphone array post-filtering using supervised machine learning for speech enhancement," in *Proceedings of Interspeech*, pp. 2675–2679, 2014.
[43] P. Pertila and E. Cakir, "Robust direction estimation with convolutional neural networks based steered response power," in *Proceedings of ICASSP*, pp. 6125–6129, 2017.
[44] D. Povey, *et al.*, "The KALDI speech recognition toolkit," in *Proceedings of ASRU*, 2011.
[45] T. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proceedings of ICASSP*, pp. 4580–4584, 2015.
[46] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, pp. 55-59, 2013.
[47] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proceedings of ASRU*, pp. 24– 29, 2011.
[48] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent dropout without memory loss," *arXiv:1603.05118*, 2016.
[49] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 18, pp. 260–276, 2010.
[50] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 24, pp. 1450–1463, 2016.
[51] T. Tan, *et al.*, "Speaker-aware training of LSTM-RNNs for acoustic modelling," in *Proceedings of ICASSP*, pp. 5280–5284, 2016.
[52] I. Tashev and A. Acero, "Microphone array post-processor using instantaneous direction of arrival," in *Proceedings of IWAENC*, 2006.
[53] Y. Tu, J. Du, L. Sun, F. Ma, and C. Lee, "On design of robust deep models for CHiME-4 multi-channel speech recognition with multiple configurations of array microphones," in *Proceedings of Interspeech*, pp. 394–398, 2017.
[54] E. Vincent, *et al.*, "The second 'CHiME' speech separation and recognition challenge: an overview of challenge systems and outcomes," in *Proceedings of ASRU*, pp. 162–167, 2013.
[55] E. Vincent, A. Watanabe, A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Comp. Speech Lang.*, vol. 46, pp. 535-557, 2017.
[56] D.L. Wang and G.J. Brown, Ed., *Computational auditory scene analysis: Principles, algorithms, and applications*. Hoboken NJ: Wiley & IEEE Press, 2006.

[57] D.L. Wang and J. Chen, "Supervised speech separation based on deep learning: an overview." arXiv:1708.07524, 2017.
[58] Y. Wang, A. Misra, and K. Chin, "Time-frequency masking for large scale robust speech recognition," in *Proceedings of Interspeech*, pp. 2469–2473, 2015.
[59] Y. Wang, A. Narayanan, and D.L. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 22, pp. 1849-1858, 2014.
[60] Y. Wang and D.L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 21, pp. 1381-1390, 2013.
[61] Z.-Q. Wang and D.L. Wang, "A joint training framework for robust automatic speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 24, pp. 796-806, 2016.
[62] Z.-Q. Wang and D.L. Wang, "Phoneme-specific speech separation," in *Proceedings of ICASSP*, pp. 146-150, 2016.
[63] Z.-Q. Wang and D.L. Wang, "Recurrent deep stacking networks for supervised speech separation," in *Proceedings of ICASSP*, pp. 71-75, 2017.
[64] Z.-Q. Wang and D.L. Wang, "Unsupervised speaker adaptation of batch normalized acoustic models for robust ASR," in *Proceedings of ICASSP*, pp. 4890-4894, 2017.
[65] F. Weninger, *et al.*, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *Proceedings of LVA/ICA*, 2015.
[66] F. Weninger, J. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proceedings of GlobalSIP*, pp. 740-744, 2014.
[67] F. Weninger, J. Le Roux, J. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," in *Proceedings of Interspeech*, pp. 865– 869, 2014.
[68] D.S. Williamson, Y. Wang, and D.L. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 24, pp. 483–492, 2016.
[69] P.C. Woodland, D. Pye, and M.J.F. Gales, "Iterative unsupervised adaptation using maximum likelihood linear regression," in *Proceedings of ICSLP*, pp. 1133–1136, 1996.
[70] X. Xiao, *et al.*, "A study of learning based beamforming methods for speech recognition," in *Proceedings of CHiME-4 Workshop*, 2016.
[71] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 23, pp. 7-19, 2015.
[72] T. Yoshioka, M. Ito, M. Delcroix, and e. al., "The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proceedings of IEEE ASRU*, 2015.
[73] X.-L. Zhang and D.L. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 24, pp. 967-977, 2016.
[74] X. Zhang and D.L. Wang, "Deep learning based binaural speech separation in reverberant environments," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 25, pp. 1075-1084, 2017.
[75] X. Zhang, Z.-Q. Wang, and D.L. Wang, "A speech enhancement algorithm by iterating single- and multi-microphone processing and its application to robust ASR," in *Proceedings of ICASSP*, pp. 276-280, 2017.
[76] X. Zhang, H. Zhang, S. Nie, G. Gao, and W. Liu, "A pairwise algorithm using the deep stacking network for speech separation and pitch estimation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 24, pp. 1066-1078, 2016.

APPENDIX. PUBLICATIONS RESULTING FROM THIS PROJECT

[1] Zhong-Qiu Wang and DeLiang Wang, "Recurrent deep stacking networks for supervised speech separation", in *Proceedings of ICASSP*, pp. 71-75, 2017.

[2] Xueliang Zhang, Zhong-Qiu Wang, and DeLiang Wang, "A speech enhancement algorithm by iterating single- and multi-microphone processing and its application to robust ASR", in *Proceedings of ICASSP*, pp. 276-280, 2017.

[3] Zhong-Qiu Wang and DeLiang Wang, "Unsupervised speaker adaptation of batch normalized acoustic models for robust ASR", in *Proceedings of ICASSP*, pp. 4890-4894, 2017.

[4] Zhong-Qiu Wang and DeLiang Wang, "On spatial features for supervised speech separation and its application to beamforming and robust ASR", in *Proceedings of ICASSP*, to appear, 2018.

[5] Zhong-Qiu Wang and DeLiang Wang, "Mask weighted STFT ratios for relative transfer function estimation and its application to robust ASR", in *Proceedings of ICASSP*, to appear, 2018.

[6] Peidong Wang and DeLiang Wang, "Utterance-wise recurrent dropout and iterative speaker adaptation for robust monaural speech recognition", in *Proceedings of ICASSP*, to appear, 2018.

[7] Peidong Wang and DeLiang Wang, "Filter-and-convolve: a CNN based multichannel complex concatenation acoustic model", in *Proceedings of ICASSP*, to appear, 2018.

LIST OF ACRONYMS

ASR - automatic speech recognition
BLSTM - bidirectional long short-term memory
BRIR - binaural room impulse responses
CLDNN - fully connected deep neural networks
D0A - direction of arrival
DNNs deep neural networks
ELU - exponential linear units
FFT - fast Fourier transform
fMLLR - feature-space maximum likelihood linear regression
IBM - ideal binary mask
ILD - interaural level differences
IRM - ideal ratio mask
ITD - interaural time differences
LHN - linear hidden network
LHUC - learning hidden unit contributions
LIN - linear input network
LM - language model
LSTM - long short term memory
MSC - magnitude squared coherence
MVDR - Minimum Variance Distortionless Response
PESQ - Perceptual Estimation of Speech Quality
RLU - rectified linear units
RNN - recurrent neural networks
RTF - relative transfer function
SDR - Signal-to- Distortion Ratio
SNR - signal-to-noise ratio (SNR)
STFT - short-time Fourier transform
STOI - Short-Time Objective Intelligibility
T-F - time-frequency
WDAS - weighted delay-and-sum
WER - word error rate
WRBN - wide residual BLSTM network