A Joint Training Framework for Robust Automatic Speech Recognition

Zhong-Qiu Wang and DeLiang Wang, *Fellow, IEEE*

***Abstract*—Robustness against noise and reverberation is critical for ASR systems deployed in real-world environments. In robust ASR, corrupted speech is normally enhanced using speech separation or enhancement algorithms before recognition. This paper presents a novel joint training framework for speech separation and recognition. The key idea is to concatenate a deep neural network (DNN) based speech separation frontend and a DNN-based acoustic model to build a larger neural network, and jointly adjust the weights in each module. This way, the separation frontend is able to provide enhanced speech desired by the acoustic model and the acoustic model can guide the separation frontend to produce more discriminative enhancement. In addition, we apply sequence training to the jointly trained DNN so that the linguistic information contained in the acoustic and language models can be back-propagated to influence the separation frontend at the training stage. To further improve the robustness, we add more noise- and reverberation-robust features for acoustic modeling. At the test stage, utterance-level unsupervised adaptation is performed to adapt the jointly trained network by learning a linear transformation of the input of the separation frontend. The resulting sequence-discriminative jointly-trained multistream system with run-time adaptation achieves 10.63% average word error rate (WER) on the test set of the reverberant and noisy CHiME-2 dataset (task-2), which represents the best performance on this dataset and a 22.75% error reduction over the best existing method.**

***Index Terms*—CHiME-2, deep neural networks (DNN), joint training, robust automatic speech recognition, speech separation, time-frequency masking.**

I. Introduction

DNN-HMM hybrid methods become the dominant approach in automatic speech recognition. Different from traditional GMM-HMM methods, which use GMMs for acoustic modeling, the DNN-HMM approach uses DNN to predict senone states based on acoustic inputs with a large context window [19]. Compared with traditional GMM-HMM approaches, large improvement has been observed. Since speech recognition is itself a sequence classification problem, recently, different neural network architectures that can capture sequential dependencies, such as the convolutional neural networks (CNNs) [37], recurrent neural networks (RNNs) [12], long-short term memory (LSTM) [15] and time-delayed neural networks [33], are introduced to improve the performance of ASR systems in addition to commonly used feed-forward DNNs. Although a lot of progress has been made in ASR on clean speech, the performance still drops sharply in the presence of reverberation, mismatched noises and low SNR conditions. Improving the robustness of ASR systems in such environments remains a challenge.

Although DNN-based acoustic models are robust to noisy input with small variations [57], speech separation algorithms are able to significantly improve recognition performance even when deep neural networks are used for acoustic modeling [4]. Recently, different DNN-based speech separation methods, such as the time-frequency (T-F) masking [55], [46], [48], spectral mapping [56], [14], [1], and signal approximation [54], [8], [53], are developed and shown to perform surprisingly well even in highly adverse environments.

When incorporating speech separation into ASR, there are three commonly used strategies. The first strategy is to conduct acoustic modeling on clean speech. At the test stage, a separation frontend is used to enhance noisy speech before recognition [29], [6]. A disadvantage would occur when the separation frontend introduces distortions unseen by the acoustic model trained on clean speech [29]. Nonetheless, this strategy is still useful from a practical perspective as it allows modular research on noise-robust ASR. The second strategy avoids the distortion problem to some extent by using a separation frontend to enhance both training and test set first, and conducts acoustic modeling on the enhanced training set. It may be able to improve the recognition performance since the features may become cleaner after enhancement. However, the performance of this approach is highly dependent on the performance of the separation frontend [4], [41]. As suggested in [40], it might be better to let the acoustic model see enough input variations at the training stage. The third strategy performs acoustic modeling on noisy speech and at the test stage, noisy features are fed to the acoustic model for decoding directly or feature enhancement first. The resulting multi-condition training strategy is shown to be very effective [43] but gives unimpressive performance in matched conditions [25]. In addition, when the performance of speech separation is good, using the acoustic model trained on noisy data for decoding may not sound like a reasonable idea. Different strategies have their own advantages and disadvantages, and which strategy should be adopted is highly dependent on the situation.

Manuscript received August 13, 2015; revised November 30, 2015 and January 31, 2016; accepted February 03, 2016. Date of publication February 11, 2016; date of current version March 08, 2016. This work was supported in part by an NSF grant (IIS-1409431), in part by an AFOSR grant (FA9550-12-1-0130), and in part by the Ohio Supercomputer Center. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Murat Saraclar.

Z.-Q. Wang is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210-1277 USA (e-mail: wangzhon@cse.ohio-state.edu).

D. Wang is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210-1277 USA, and also with the Center for Cognitive and Brain Sciences, The Ohio State University, Columbus, OH 43210-1277 USA (e-mail: dwang@cse.ohio-state.edu).

Digital Object Identifier 10.1109/TASLP.2016.2528171

Speech separation and recognition are not two independent tasks and they can clearly benefit from each other. Many studies in robust ASR focus on improving the performance of speech separation [26]. In other studies, first-pass recognition results [27], [53], [3] or language models [18] are utilized to help speech separation. In our previous studies [30], [31], [50], we proposed to integrate speech separation and acoustic modeling via joint adaptive training. In this study, we further develop this approach and propose various techniques to elevate the performance. The present work mainly makes the following four contributions.

First, we concatenate a DNN-based speech separation frontend, a trainable mel-filterbank and a DNN-based acoustic model together to build a larger and deeper DNN, and jointly adjust the weights in each module via the back-propagation algorithm. Note that mel-filtering can be represented as one layer in a neural network [36] since it is just a linear transformation of power spectrogram. The separation frontend is trained to reconstruct noise-free power spectrogram via time-frequency masking. Acoustic modeling is done in the mel-spectrogram domain. With joint training, the separation frontend and filterbank are able to provide enhanced features expected by the acoustic model. In addition, the linguistic information contained in the acoustic model is allowed to flow back to influence the separation frontend and filterbank. Furthermore, the filterbank can be trained according to the separation frontend and acoustic model [36]. Second, concatenating the separation frontend and acoustic model to form a bigger DNN naturally leads us to sequence-discriminative training applied to the jointly trained DNN for further improvement. This way, at the training stage, the information from language models can be flowed back to influence not only the acoustic model but also the separation frontend by optimizing sequence-discriminative criterion. Third, utterance-level unsupervised adaptation is performed at the test stage to adapt the jointly trained DNN to potentially mismatched conditions or new speakers to some extent. Fourth, we find that adding more robust features for acoustic modeling can significantly improve the robustness of ASR systems. Traditionally, log mel-spectrogram is widely used as the only feature for DNN-based acoustic models [29], [41], [52], [24], [10], [35], partly because DNN is believed to be capable of extracting highly nonlinear discriminative information from relatively raw input. However, DNN robustness against reverberation and noise is limited. In this study, we incorporate additional robust features, such as AMS [23], RASTA-PLP [17], MRCG [2], and PNCC [21], for acoustic modeling. This multi-stream strategy improves recognition rate substantially.

The proposed sequence-discriminative jointly-trained multi-stream approach achieves 10.63 percent average WER on the test set of the noisy and reverberant CHiME-2 dataset (task-2) [43]. To our knowledge, this represents the best result on this dataset to date.

The rest of this paper is organized as follows. We describe our joint training approach in Section II, followed by experiments and evaluations in Section III. We conclude this paper in Section IV. A preliminary version of this work is presented in [50]. There are major differences from this preliminary work, including the use of sequence training and unsupervised adaptation in the present study, as well as the Kaldi toolkit to give better baseline ASR systems and clean alignments. These methodological differences lead to a large performance improvement over [50]. Also, more systematic comparisons are provided in this paper.

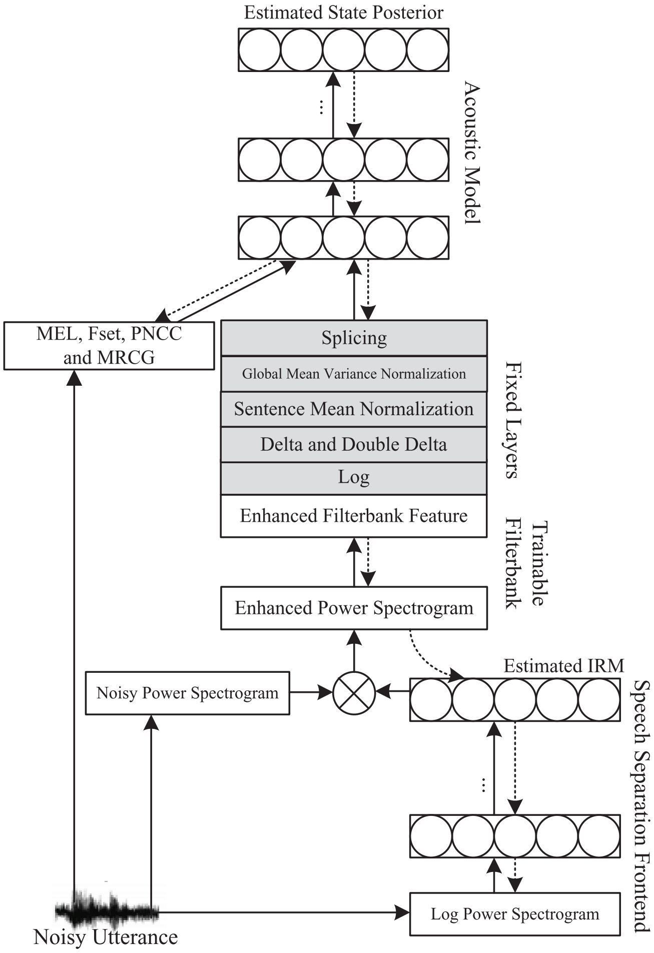

Fig. 1. Schematic diagram of the proposed joint training framework. The layer shown in gray means that the weights or operations of that layer are fixed. Solid and dotted arrows indicate the directions of forward pass and backward pass, respectively. See text for more details.

II. System Description

Our system is built in a step-by-step way. We first train a separation frontend and an acoustic model separately, both using DNNs. Then we concatenate the separation frontend, mel-filterbank and acoustic model together to construct a deeper and larger DNN, and jointly adjust the weights in all modules. After that, we replace the cross-entropy criterion used at the joint training stage with sequence-discriminative criterion for sequence training. Finally, we perform utterance-level unsupervised adaptation at the test stage. The overall framework of our system is shown in Fig. 1. More details are provided in the following sections.

A. Speech Separation

Originated in computational auditory scene analysis [45], T-F masking has shown considerable potential for removing additive noise in noisy speech. The key idea of this method is to estimate the ideal binary mask (IBM) [44] that identifies speech dominant and noise dominant T-F units, or the ideal ratio mask (IRM) [32], which represents the ratio of speech energy to the sum of speech energy and noise energy within each T-F unit. In this framework, speech separation is formulated as a supervised mask estimation problem. Different learning machines, such as GMMs, support vector machines (SVMs) and multi-layer perceptrons (MLPs), have been used for mask estimation. Recently, DNN is employed for mask estimation, and achieves very promising separation performance in both matched and un-matched test conditions [55]. Recent listening tests show that DNN based IBM estimation produces substantial speech intelligibility improvements of noisy utterances for both hearing-impaired and normal-hearing listeners [16]. In addition, different training targets are carefully analyzed recently [49], and it is suggested that the IRM is likely to be a better training target for supervised speech separation. Therefore, we utilize DNNs to estimate the IRM in this study.

The ideal mask can be defined in different T-F representation domains, such as cochleagram domain, mel-spectrogram domain or power spectrogram domain. In line with later joint training, the IRM in this study is defined in the power spectrogram domain [49]:

$$M\left(t,f\right)=\frac{S\left(t,f\right)}{S\left(t,f\right)+N\left(t,f\right)} \tag{1}$$

where M is the IRM of a noisy signal created by mixing a noise-free utterance with a noise signal at a specific SNR level, S represents the power spectrogram of the noise-free utterance, N stands for the power spectrogram of the noise signal, and t and f index time and frequency respectively.

At the test stage, the IRM must be estimated from noisy utterances. We employ a DNN as the discriminative learning machine for mask estimation. The DNN has four hidden layers each with 1024 rectified linear units (ReLUs) [11]. There are 161 sigmoidal units in the output layer, corresponding to the dimension of each frame in the power spectrogram. No pretraining is performed. Starting from random initialization, the network is trained for a maximum of 50 epochs to minimize the cross-entropy loss function within each T-F unit using stochastic gradient descent with momentum and Adagrad [7]. The loss function is defined as:

$$\begin{aligned} L\left(M^{*}\right)=&-\frac{1}{T}\sum_{t,f}[M\left(t,f\right)\log M^{*}\left(t,f\right)\\ &+\left(1-M\left(t,f\right)\right)\log\left(1-M^{*}\left(t,f\right)\right) \end{aligned} \tag{2}$$

where $M^{*}\left(t,f\right)$ is the estimated mask at time t and frequency f, and T is the total number of frames in the dataset. The momentum is linearly increased from 0.1 to 0.5 in the first 5 epochs and fixed at 0.9 afterward. The learning rate is fixed at 0.005 in the first 20 epochs and linearly decreased to 0.0005 in the following 30 epochs. The dropout rates of the input layer and all hidden layers are set to 0.3. The maximum L_2 norm of the incoming weights of each neuron in the hidden layers is set to 1. The mini-batch size is set to 256. A development dataset is used for parameter tuning and early stopping.

The feature used for mask estimation is log-compressed power spectrogram. We splice a large context window of 19 frames centered at the current frame as the input to DNN. In this study, the frame length is 20 ms and frame shift is 10 ms. Therefore, for a signal with 16 kHz sampling rate, the input dimension corresponding to one frame is 3059 ($161 * 19$). Note that the log power spectrogram feature is globally mean-variance normalized before splicing.

At the test stage, after obtaining the estimated IRM from a noisy utterance using the trained DNN, we multiply it pointwisely with the power spectrogram of the noisy utterance to get the enhanced power spectrogram, i.e.

$$X^{*}=M^{*}\otimes X \tag{3}$$

where X^{*} is the resulting enhanced power spectrogram, M^{*} is the estimated IRM, X is the noisy power spectrogram, and $\otimes$ represents point-wise matrix multiplication.

B. Acoustic Modeling

The DNN-HMM hybrid approach is dominant in ASR today. We utilize a DNN with 7 hidden layers each with 2048 ReLUs for acoustic modeling. The DNN is trained to estimate the posterior probability of each senone state by minimizing cross-entropy at the training stage. All the other training recipes follow the DNN training for mask estimation presented in the previous section.

Log mel-spectrogram is widely used as the only feature for acoustic modeling. However, mel-spectrogram itself is not robust to noise and reverberation. To improve the robustness of ASR systems, we add more robust features for acoustic modeling as different features contain different and perhaps complementary information for senone state discrimination. In this study, we use a subset of the following features for acoustic modeling.

- 40-dimensional log mel-spectrogram together with its delta and double delta components (MEL). We perform sentence level mean normalization before splicing an 11-frame context window;
- 256-dimensional multi-resolution cochleagram (MRCG) [2] with its delta and double deltas. The recently proposed MRCG is shown to be relatively robust to additive noise for mask estimation;
- 31-dimensional power-normalized cepstral coefficients (PNCC) [21] together with their deltas and double deltas. Sentence level mean normalization is performed before splicing an 11-frame context window. The PNCC feature is shown to be robust to reverberation and additive noise;
- 13-dimensional RASTA-PLP [17]. The context window is set to 7;
- 15-dimensional amplitude modulation spectrogram (AMS) [23] extracted from each of 26 channels;
- 31-dimensional narrowband mel-frequency cepstral coefficients (MFCC) with the analysis window of 20 ms. The context window is set to 7;
- 31-dimensional wideband MFCC with the analysis window of 200 ms. The context window size is set to 7.

The last four features, denoted as Fset, are shown to have complementary power for mask estimation [31], [47][1]. In this study, we directly use Fset features for acoustic modeling. The frequency ranges are all set 64 to 8000 Hz. Following common practice, we use 9 frames centered at the current frame to calculate delta and double delta components. With the features mentioned above for acoustic modeling, the input dimension is 4026 $(40 * 3 * 11 + 256 * 3 + 31 * 3 * 11 + 13 * 7 + 15 * 26 + 31 * 7 + 31 * 7)$. They are globally mean-variance normalized before DNN training. To facilitate comparison, we always include the MEL features for acoustic modeling.

C. Joint Training

As illustrated in Fig. 1, the key idea for joint training is to concatenate an acoustic model DNN and a speech separation DNN to form a larger and deeper neural network, and jointly adjust the weights in all modules. The link for concatenating the separation frontend and the acoustic model is a trainable filterbank layer and a set of layers with fixed operations, which basically represent the extraction of the enhanced MEL features (with delta and double deltas and an 11-frame context window) (see also [30], [31], [50]). More specifically, after obtaining the estimated IRM from the separation frontend based on the log power spectrogram of a noisy utterance, we multiply it point-wisely with the noisy power spectrogram to get the enhanced power spectrogram. The enhanced power spectrogram is then fed into the trainable filterbank layer to get the enhanced filterbank feature. Afterwards, we compress it logarithmically, add delta and double deltas, perform sentence-level mean normalization, conduct global mean-variance normalization, and splice 11 frames to yield the enhanced MEL features. The enhanced MEL features, together with other robust features, are finally passed to the acoustic model to estimate state posterior probabilities. Interestingly, the joint training framework can be performed in a single neural network because the point-wise multiplication, filtering, sentence- and global-level normalization, adding delta and double deltas are all linear transformations. In addition, the derivatives of the logarithmic function can be easily computed. Therefore, we are able to flow the error signal from the acoustic model back to the filterbank layer and the separation frontend, and jointly train all modules using the back-propagation algorithm.

A similar frontend and backend joint training approach was presented by Gao *et al.* [9], where feature mapping is employed as the frontend. It has been suggested that masking is likely a better approach than mapping for speech separation [49]. In addition, the output dimension of their frontend is equal to the input dimension, which consists of many consecutive frames and is large. In contrast, we obtain enhancement results per single frame. Furthermore, their frontend obtains enhanced MEL features by direct mapping instead of using a trainable filterbank layer and fixed layers to transform the enhanced power spectrogram.

In our approach, parameter initialization is critical before joint training. Randomly initializing all the parameters is unlikely to be effective considering the size of the network. Here we use the weights in a separately trained acoustic model and a separately trained separation frontend to initialize the corresponding parts of the DNN for joint training. Following [36], we initialize the parameters in the trainable filterbank (FB) layer using

$$W^{FB} = \exp\left(W^{*}\right) \tag{4}$$

where W^{*} is initialized to

$$W^{*} = \log\left(\max\left(Mel\_FB, eps\right)\right) \tag{5}$$

Here Mel_FB denotes the standard 40-dimensional mel-filterbank and eps is a small constant (10^{-3} in this study). With (4), every time W^{*} is updated, all the parameters in the filterbank are ensured to be non-negative. With (5), all the parameters in the filterbank can be updated. Using an eps term instead of 0 in the mel-filterbank for initialization is found to consistently improve the performance of our system.

The whole network is trained for 15 epochs to minimize the cross-entropy criterion from the acoustic model alone. In preliminary studies, we tried to put a weight between the loss of the acoustic model and the loss of the separation frontend, expecting the performance of both tasks to improve. However, no clear improvement on the ASR performance was observed. The learning rate is fixed at 0.0005 for the first 8 epochs and linearly decreased to 10^{-5} for the next 7 epochs. Note that the learning rate of 0.0005 is the smallest learning rate used in the previous sections[2]. The momentum is fixed at 0.9 for all the epochs. The mini-batch size is set to 512. No dropout and weight normalization is performed at the filterbank layer and fixed layers. The sentence-level mean of each utterance and global mean and variance are updated at the beginning of each epoch in the forward pass. All the other network setup and training strategies follow the DNN training in the previous sections.

D. Sequence-Discriminative Training

The previous sections describe how the DNN-based acoustic models are trained to minimize the cross-entropy criterion at the frame level. As automatic speech recognition is itself a sequence classification problem, it is sensible to optimize the sequence-discriminative criterion to better capture the essence of this problem. It is widely known that sequence training is helpful for GMM-HMM systems. In recent studies, sequence training is also found to be useful for DNN-HMM hybrid systems [42], [31]. Here, we investigate the effectiveness of sequence training criterion on the joint training system. As suggested in [42], different sequence training criterion gives similar performance on recognition rates. In this study, we replace the frame-wise cross-entropy criterion with the state-level minimum Bayes risk (sMBR) [22] and back-propagate the error

[1] We tried to use this feature set to estimate the IRM defined in the power spectrogram domain and in the mel-spectrogram domain as well, but the ASR performance is not as good as using the log power spectrogram directly.

[2] The optimization processes of the separately trained acoustic model and separation frontend have already converged long before reaching the 0.0005 learning rate. Therefore, the improvement from joint training is not simply because of using a very small learning rate on un-converged models.

signal from this criterion to influence the weights in the acoustic model, filterbank and separation frontend. This method is expected to improve the performance of speech recognition. We believe that this method may also benefit mask estimation since the error signal from the sequence training criterion contains information from language models, which is rarely exploited in speech separation research.

To speed up the sMBR training, we re-generate the lattices after the first epoch, and further train the network for six epochs. The learning rate is linearly decreased from 10^{-5}, which is the smallest learning rate used at the joint training stage, to 10^{-6}. The acoustic scaling factor is fixed at 0.1. The mini-batch size is variable corresponding to the length of each utterance. The sentence-level mean of each utterance is updated dynamically in the forward pass for each mini-batch. All the other network setup and training recipes follow the DNN training at the joint training stage. Most of the times, the performance on the development set converges in 3-5 epochs after re-generating the lattices.

E. Unsupervised Adaptation

Adaptation is commonly performed on well-trained acoustic models to compensate the differences between training and test conditions. It can be done in a supervised or unsupervised way, depending on whether the labels of adaptation data are available. Many adaptation methods have been proposed for DNN based acoustic models, such as linear transformation [39], [29], conservative training [58], and subspace based methods [38]. In [28], it is suggested that the linear input network (LIN) and linear hidden network based approaches are better than linear output network, factorization and KL-divergence based adaptation.

We perform unsupervised adaptation to our jointly trained acoustic models following the LIN approach. At the test stage, given a single test utterance, we first use the un-adapted jointly-trained sequence-discriminative model to generate initial decoding results. The first-pass decoded state sequence is then used as the labels for learning a linear transformation of the input features of the separation frontend by minimizing the cross-entropy criterion calculated from the acoustic model, with all the other parameters fixed. The linear transformation is defined as follows:

$$\hat{x}_{t,f} = w_f x_{t,f} + b_f \tag{6}$$

where $x_{t,f}$ denotes the globally mean-variance normalized log power spectrogram, corresponding to the un-adapted input of the separation frontend, $\hat{x}_{t,f}$ denotes the adapted features, and w_f and b_f are the parameters to be learned. For a test utterance, the number of parameters to be learned is 322 $(161 + 161)$, which is approximately in the same range of the number of frames in the test utterance.

For each utterance, the adaptation process is run for 20 epochs with a mini-batch size equal to the length of the utterance. The learning rate is linearly decreased from 0.005 to 10^{-5}. We simply adopt the learned parameters at the last epoch due to the lack of a development set. All the other training recipes and network setup follow the DNN training described in the previous sections. Note that we also perform dropout on the adapted features, which consistently improves the performance due to alleviated overfitting. After we obtain all the linear transformation for each test utterance, we re-generate the likelihood and run a second-pass decoding to obtain the final results.

It may be argued that the cross-entropy loss function used at the adaptation stage would counteract the effect of the sequence-discriminative criterion used at the joint training stage. We note that this would not be a problem since the cross-entropy criterion will make posterior estimates closer to the initial decoding results generated from the sequence-discriminative model.

A similar adaptation method was proposed in [29]. One key difference is that we perform adaptation on the input of the separation frontend rather than on the output of the separation frontend. We think that our strategy is better since, if we perform adaptation on the input of the separation frontend, the enhancement results would be changed in a highly non-linear way rather than in a simple linear fashion.

Finally, we believe that this unsupervised adaptation technique with the learned linear transformation can also adapt a well-trained separation frontend to new test environments to some extent.

III. Evaluations and Comparisons

We evaluate our method on the recently proposed reverberant and noisy CHiME-2 dataset (task-2) [43][3]. The CHiME-2 dataset is created by first convolving clean utterances in the WSJ0-5 k dataset with time-varying binaural room impulse responses (BRIRs) and then mixing with reverberant noises at six SNR levels equally spaced from -6 dB to 9 dB. The BRIRs and reverberant noises are recorded with the same microphone and living room setup. The recorded noises contain major noise sources in a typical kitchen or living room, such as competing speakers, electronic devices, footsteps, laughter, and distant noises. The multi-conditional training set (si_tr_s) contains 7138 utterances ($\sim$14.5h) in total. The development set (si_dt_05) contains 409 utterances at each SNR level ($\sim$4.5h). The test set (si_et_05) contains 330 utterances at each SNR level ($\sim$4h). The CHiME-2 dataset also provides reverberant noises, and reverberant noise-free utterances corresponding to the multi-conditional training set. With the noises, clean speech, reverberant noise-free utterances, and noisy-reverberant utterances available, we can readily evaluate the recognition performance together with speech separation performance of our system.

Our system is monaural in nature, and we simply average the signals from the left and right channel before extracting features. In our experiments, this technique is much better than only using one of these two channels. A GMM-HMM system is built using the Kaldi toolkit [34] on the clean utterances in the WSJ0-5 k to get the senone state for each frame of the corresponding noisy-reverberant utterances. Following the common pipeline in the Kaldi toolkit, the GMM-HMM system

[3] Available at http://spandh.dcs.shef.ac.uk/chime_challenge/chime2013/WSJ0/.

TABLE I
PERFORMANCE (% WER) USING MULTICONDITION TRAINING WITH MORE ROBUST FEATURES FOR ACOUSTIC MODELING

Features for Acoustic Modeling	dev. set	test set						
	Average	-6dB	-3dB	0dB	3dB	6dB	9dB	Average
MEL	19.40	26.77	20.49	16.14	12.80	10.67	10.11	16.16
+sMBR	17.24	23.87	17.35	13.64	11.30	9.10	8.28	13.92
+adaptation	**16.81**	**22.64**	**16.85**	**12.78**	**10.44**	**8.69**	**7.79**	**13.20**
MEL+PNCC	18.54	25.13	18.57	14.94	11.73	9.51	8.57	14.74
+sMBR	16.52	23.22	16.59	12.46	10.52	8.24	7.49	13.09
+adaptation	16.10	22.03	16.33	12.22	10.29	7.66	7.36	12.65
MEL+PNCC+MRCG	17.99	23.33	17.92	14.20	11.36	8.95	8.05	13.97
+sMBR	15.97	22.01	15.62	12.18	10.59	8.18	7.12	12.62
+adaptation	15.57	21.17	15.21	11.83	10.55	7.77	6.80	12.22
MEL+PNCC+MRCG+Fset	17.93	23.09	17.17	13.32	10.41	8.71	8.07	13.46
+sMBR	15.63	21.17	14.96	12.24	9.83	7.68	7.14	12.17
+adaptation	**15.48**	**20.51**	**14.68**	**11.77**	**9.70**	**7.49**	**7.02**	**11.86**

is first built using the MFCC feature. Then we concatenate 13-dimensional MFCC feature within a 7-frame context window, and utilize linear discriminant analysis (LDA) to compress the concatenated feature to 40 dimensions. After that, we decorrelate it via maximum likelihood linear transform (MLLT) and use feature-space maximum likelihood linear regression (fMLLR) to reduce speaker variance, which is estimated by speaker adaptive training. The resulting cross-word tied-state tri-phone GMM-HMM system contains 1965 senone states in total. The initial clean alignments are obtained by performing forced alignment on the clean utterances. To refine the initial clean alignments, we further train a DNN-based acoustic model using the MEL features of the clean utterances, and re-generate clean alignments. Such clean alignments are used as the labels for training all the acoustic models in this study. Note that the DNN-HMM hybrid system built on the clean utterances is a powerful recognizer. It achieves 2.15% WER on the clean test set of the WSJ0-5 k dataset. Therefore, we believe that these high-quality labels can guide the DNN-based acoustic model to perform well on discriminating different senone states even when the input features are very noisy and the input SNR is very low. We use the CMU pronunciation dictionary and the official 5 k close-vocabulary tri-gram language model in our experiments. Note that this language model is used for decoding at the test stage and generating the lattices of the training utterances at the sequence training stage.

The training data for mask estimation is obtained from parallel noisy-reverberant and reverberant noise-free data. The mixed noise signals can be obtained by direct subtraction. With these datasets available, we train a separation frontend using the method detailed in Section II.A. The frontend is trained to remove additive noise in noisy-reverberant utterances. Note that the noisy-reverberant dataset, i.e. the multi-conditional training data, is used for both mask estimation and acoustic modeling.

Our experiments are done in an incremental manner. We first build our acoustic models using feature subsets selected according to the performance on the development set. Then we jointly train the acoustic models with the separation frontend. Afterwards, we perform sequence training on the jointly trained DNN. Finally, we perform unsupervised adaptation to the sequence-discriminative jointly-trained DNN at the test stage.

A. *Expanded Feature Set for Acoustic Modeling*

We first report the results of incorporating more robust features for acoustic modeling. In this experiment, no speech enhancement or separation is performed. We simply train acoustic models multi-conditionally by adding more robust features and do not tune the network structure or training recipes for each feature set. To push up the baselines, we perform sequence training on the multi-conditionally trained acoustic models, which is followed by unsupervised adaptation at the test stage. The WER results are presented in Table I.

If we only train our acoustic models using the cross-entropy criterion, with the commonly used MEL features alone, we are able to obtain 16.16% average WER on the test set. Note that if we just use the default DNN code for the CHiME-2 dataset in the Kaldi toolkit, we can only obtain 17.49% average WER on the test set. This is consistent with the results obtained in [13]. The major differences are that we use ReLUs, dropout and Adagrad for training, while the default DNN code uses sigmoidal units, pre-training and stochastic gradient descent. By adding the PNCC feature, the average WER can be reduced to 14.74%. After appending the MRCG feature, the WER is brought down to 13.97%. The performance is further pushed to 13.46% average WER after we add the Fset features. Note that this result is already better than our previous best result [50] using the same set of features on this dataset, mainly because better clean alignments are generated using the Kaldi toolkit.

We then apply sequence training to the multi-conditionally trained acoustic models. The training recipes follow the sequence training described in Section II.D. We observe that sequence training leads to large improvement for all the input features, and the relative improvement becomes smaller if more features are used for acoustic modeling.

Finally, we apply utterance-level unsupervised adaptation to the sequence-discriminative acoustic models. Similar to Section II.E, given a test utterance, we first decode it to obtain a hypothesized state sequence, from which we learn a linear transformation of the input features. To reduce the number of parameters to be learned and make a fair comparison with later experiments, we only learn a linear transformation for the MEL features. Learning linear transformations for other features may decrease the performance, simply because too many parameters are learned. Thus, the total number of parameters to be

TABLE II
PERFORMANCE (% WER) COMPARISON OF THE PROPOSED APPROACH WITHOUT EXTRA ROBUST FEATURES

Approaches	Acoustic Model	dev. set	test set						
		Average	-6dB	-3dB	0dB	3dB	6dB	9dB	Average
Plug-and-Play	MEL	18.22	23.58	18.53	14.85	12.42	9.68	9.56	14.77
	+sMBR	16.63	22.72	16.12	13.81	10.84	8.61	8.39	13.42
	+adaptation	16.05	21.18	15.82	12.16	10.54	8.14	7.88	12.62
Re-training	Enhanced MEL	18.67	25.85	19.20	15.93	12.52	9.96	9.21	15.45
	+sMBR	17.08	24.38	17.19	13.66	11.10	8.69	8.20	13.87
	+adaptation	16.59	23.54	16.40	12.76	10.55	8.37	7.66	13.21
Re-training	Enhanced MEL + MEL	18.31	25.31	18.83	15.69	11.94	9.23	8.89	14.98
	+sMBR	16.50	24.10	16.68	14.18	10.42	8.63	7.88	13.65
	+adaptation	16.07	22.70	16.14	13.32	9.96	7.88	7.40	12.9
Jointly training frontend, AM and filterbank	Jointly enhanced MEL	17.63	22.55	17.65	14.42	11.36	9.23	8.74	13.99
	+sMBR	15.28	20.44	14.66	12.39	9.81	7.73	7.38	12.07
	+adaptation	**14.56**	**18.72**	**13.77**	**11.36**	**9.32**	**7.32**	**6.86**	**11.23**
Jointly training frontend and AM	Jointly enhanced MEL	17.62	23.15	17.69	14.72	11.38	9.30	9.15	14.23
	+sMBR	15.30	20.61	14.89	12.48	9.81	7.85	7.49	12.19
	+adaptation	14.60	19.13	13.67	11.40	9.19	7.51	7.08	11.33
Directly training a large DNN	Log power spectrogram + MEL	19.06	24.88	18.91	15.15	12.57	10.44	9.25	15.2

learned is 240 $(40 * 3 + 40 * 3)$ for each test utterance. From Table I, we can see that unsupervised adaptation leads to consistent improvement, while the relative improvement for acoustic models with more features becomes smaller as well.

Compared with only using the MEL features, adding all the extra robust features for acoustic modeling reduces the average WER by 2.7 (16.16% to 13.46%), 1.75 (13.92% to 12.17%), and 1.34 percentage points (13.20% to 11.86%) without sequence training or adaptation, with sequence training but no adaptation, and with sequence training and adaptation, respectively. These considerable improvements occur probably because features are extracted from different domains using different filterbanks, compression operations and environmental compensations, and therefore they likely complement each other for acoustic modeling on multi-conditional data. This suggests that relying on the DNN to learn optimal non-linear features from relatively raw input, e.g. the MEL features, may not be the optimal strategy for robust ASR. Combining the feature learning ability of DNNs and domain knowledge may be a better way for improving the robustness of ASR systems.

As shown in Table I, the average WER on the development set keeps decreasing as we add more and more features. Therefore, in the following experiments, we add the PNCC, MRCG and Fset features for acoustic modeling. Note that we do not perform any kind of enhancement on these extra features since they are considered to be inherently robust in our study. To facilitate comparisons, we also report the results based on the MEL features alone.

B. Plug-and-Play and Re-Training Approaches

Before presenting the results of the joint training approach, we explore two alternative strategies when incorporating speech separation into ASR systems.

The first strategy, denoted as plug-and-play, is to train our acoustic models using the MEL features alone or the MEL + PNCC + MRCG + Fset features. In the test stage, we use the trained separation frontend to get the enhanced power spectrogram which is then passed to the mel-filterbank to get the enhanced MEL features. Finally, together with other robust features, the enhanced MEL features are passed to the acoustic model for decoding. As shown in the first entry of Table II, if we only use the MEL features for acoustic modeling, the frontend can lead to 1.39% (16.16% to 14.77%), 0.5% (13.92% to 13.42%), and 0.58% (13.20% to 12.62%) absolute improvement without sequence training or adaptation, with sequence training but no adaptation, and with sequence training and adaptation, respectively. We can see that the relative improvement of using our frontend becomes much smaller if the acoustic model has been sequence-trained. Note that for unsupervised adaptation, we learn a linear transformation of the enhanced MEL features. The first-pass decoding results for adaptation are obtained by applying the plug-and-play approach to the sequence-discriminative acoustic model. Again, the number of parameters to be learned is 240 $(40 * 3 + 40 * 3)$. Performing unsupervised adaptation on the enhanced MEL features can lead to 0.8% (13.42% to 12.62%) average WER reduction. Similar observations can be found in the first entry of Table III, in which we use the MEL + PNCC + MRCG + Fset features for acoustic modeling.

The second alternative, denoted as re-training, is to train our acoustic models using the enhanced MEL features alone or the enhanced MEL + PNCC + MRCG + Fset features. At the test stage, after we get the enhanced MEL features, together with other robust features, we feed all of them to the acoustic model for decoding. Note that, again, the Fset, MRCG and PNCC features are directly extracted from the original noisy-reverberant utterances. The results are shown in the second entry in Tables II and III, respectively. Note that, adaptation is performed only on the enhanced MEL features. Motivated by deep stacking [5], [53], the unenhanced MEL features are additionally incorporated for acoustic modeling. The results are reported in the third entry of Tables II and III, without and with extra robust features, respectively. We can see that adding the unenhanced MEL features for acoustic modeling brings some gains for the re-training approach.

Comparing the results from plug-and-play and re-training approach, we find that the former strategy typically scores

TABLE III
PERFORMANCE (% WER) COMPARISON OF THE PROPOSED APPROACH WITH EXTRA ROBUST FEATURES

Approaches	Acoustic Model	dev. set	test set						
		Average	-6dB	-3dB	0dB	3dB	6dB	9dB	Average
Plug-and-Play	MEL+PNCC+MRCG+Fset	16.90	21.32	15.26	12.52	10.11	7.83	7.44	12.41
	+sMBR	15.34	20.04	13.64	11.56	9.56	7.64	7.08	11.59
	+adaptation	14.98	19.65	13.49	11.32	9.30	7.34	6.91	11.34
Re-training	Enhanced MEL+PNCC+MRCG+Fset	16.98	23.20	16.72	12.89	10.37	8.24	7.57	13.17
	+sMBR	15.80	22.96	16.16	12.55	9.55	7.86	7.34	12.74
	+adaptation	15.28	22.04	15.49	12.16	9.21	7,66	7.12	12.28
Re-training	Enhanced Mel+MEL+PNCC+MRCG+Fset	17.08	22.60	16.53	12.74	10.14	8.24	7.38	12.94
	+sMBR	15.52	22.87	15.58	12.61	9.40	7.70	6.76	12.49
	+adaptation	14.97	20.85	14.68	12.07	9.06	7.42	6.61	11.78
Jointly training frontend, AM and filterbank	Jointly enhanced MEL+PNCC+MRCG+Fset	15.58	20.23	14.40	11.73	9.73	7.38	7.45	11.82
	+sMBR	14.33	19.20	13.30	10.74	8.76	6.89	6.84	10.96
	+adaptation	**13.81**	**18.23**	**13.02**	**10.39**	**8.67**	**6.86**	**6.61**	**10.63**

higher. One possible reason is that, when re-training is used, the separation frontend significantly reduces the variations seen by the acoustic model at the training stage [41]. In addition, the distortion it introduces for the training utterances may be different from that for the test utterances. Another possible explanation is related to overfitting. Since in this study[4], the separation frontend is also trained on the multi-conditional training data. We can reasonably assume that the separation frontend performs better on the training set than on the development and test set. Therefore, if the enhanced training data is subsequently used to re-train the acoustic models, overfitting would likely happen. This is exactly what we encountered in our experiments. For the re-training approach, the loss of the acoustic model on the development set is much better than that of the plug-and-play or the direct multi-condition training approach; however it gives us worse performance after decoding.

C. Joint Training

Considering that more variations would be seen by the acoustic models trained on noisy-reverberant utterances and the plug-and-play approach normally gets better performance on the development set as shown in Tables II and III, we use the parameters in the acoustic models from this approach, together with the separation frontend, to initialize the corresponding parameters in the joint-training DNN, and then perform joint training. When joint training is done, sMBR training and run-time adaptation are conducted. Note that for the run-time adaptation, we learn a linear transformation of the input of the separation frontend. The number of parameters to be learned for each utterance is 322 (161 + 161).

As reported in Table II, after joint training, the performance can be improved from 14.77% to 13.99% average WER. After sMBR training, the performance is improved to 12.07%. The performance is further pushed up to 11.23% after run-time unsupervised adaption, which is helpful especially in low SNR conditions. For example, when the input SNR is -6 dB, the WER is reduced from 20.44% to 18.72%.

If we do not use extra robust features for acoustic modeling, compared with plug-and-play, we reduce the average WER by absolute 0.78% or relative 5.3% (14.77% to 13.99%) if only the cross-entropy criterion is used for joint training. The performance gap is enlarged to absolute 1.35% or relative 10.06% (13.42% to 12.07%) after sequence training is applied. If we further perform unsupervised adaptation at the test stage, the performance difference is further increased to absolute 1.39% or relative 11.01% (12.62% to 11.23%). Interestingly, the relative improvement becomes larger after sequence training and unsupervised adaptation are applied to the joint-training DNN. This trend can also be observed by comparing the first entry with the fourth entry in Table III, where more features are used for acoustic modeling. This is desirable since, in joint modeling, the noise compensation module can be seamlessly combined with other ASR techniques, such as sequence training and adaptation, to obtain further improvement.

As presented in the fourth and fifth entry of Table II, co-adapting the filterbank with the separation frontend and acoustic model can give us slightly better results. If the parameters in the filterbank are co-adapted, the performance is 0.24% (14.23% to 13.99%) average WER better after joint training, 0.12% (12.19% to 12.07%) better after sMBR training, and 0.1% (11.33% to 11.23%) better after run-time adaptation.

These results clearly demonstrate the effectiveness of joint training. We think that it is due to the reduction of the distortion problem and the linguistic information back-propagated from the acoustic model to the separation frontend. In addition, the separation frontend used in this study treats all the frames and time-frequency units equally important, without considering the underlying linguistic information that is critical for senone states discrimination. In contrast, with joint modeling, the separation frontend can be somehow informed by the acoustic model to produce more discriminative enhancement results.

The best performance we obtained on the test set is 11.23% average WER if no extra robust features are used. With extra robust features, the performance can be further improved to 10.63%. With more sophisticated training and adaptation techniques, the effectiveness of extra features is reduced. This would be welcome as using a small number of features, such as log mel-spectrogram, is favored in industry. On the other hand, incorporating more robust features for acoustic modeling is a simple and effective technique towards improved robustness of ASR systems.

[4] This underlying problem also exists in many other studies.

TABLE IV
PERFORMANCE (% WER) COMPARISON OF THE PROPOSED APPROACH WITH OTHER STUDIES

Study	dev. set	test set						
	Average	-6dB	-3dB	0dB	3dB	6dB	9dB	Average
Weng *et al*. [52]	-	38.11	29.07	22.98	17.92	14.96	13.60	22.77
Chen *et al*.[3]	20.11	-	-	-	-	-	-	16.04
Narayanan-Wang [31]	-	25.1	19.2	15.1	12.8	10.5	9.5	15.4
Weninger *et al*. [53]	17.87	23.48	17.02	13.71	10.72	8.95	8.67	13.76
sMBR+joint training+multi-stream+run-time adaptation (proposed)	13.81	18.23	13.02	10.39	8.67	6.86	6.61	10.63

It might be argued that the joint training approach just performs acoustic modeling multi-conditionally by training a very deep and large DNN on a combination of features. To address this possibility, we train a DNN with 12 $(4 + 1 + 7)$ hidden layers, each with 1600 ReLUs, on the combination of the log power spectrogram and MEL features (without robust features) using multi-condition training directly. Note that the number of parameters in this new DNN is almost the same as that in the joint training DNN. The performance, shown in the last entry of Table II, is much worse than that of joint training. This is likely because the joint training approach has better network architecture and better parameter initialization.

D. Comparison With Other Studies

In Table IV, we list the results of several other studies that report competitive results on the same dataset. All of them use the DNN-HMM hybrid approach and clean alignments from clean utterances as the labels to train their acoustic models. The system described in [52] employs an RNN to perform acoustic modeling on the noisy-reverberant training data and does not use any speech enhancement or separation. Chen *et al.* [3] utilize LSTM for both speech separation and acoustic modeling. Their ASR systems follow the re-training approach, and an iterative strategy using alignment information from their ASR system is proposed to improve speech separation and recognition simultaneously. Weninger *et al.* [53] build their frontend by training an RNN with the LSTM activation function to predict a phase-sensitive spectrum approximation objective function. They also use re-training and additional alignment information from ASR systems to boost the performance of speech separation. Their DNN based acoustic models are built in a way similar to the standard recipes in the Kaldi toolkit. Both enhanced and unenhanced log mel-filterbank features without delta components are utilized for acoustic modeling, and no extra robust features are used in their study. Han *et al.* [13] use a spectral mapping based separation frontend to enhance both the training and test set first, and perform acoustic modeling on the enhanced training set using the standard DNN training recipes in the Kaldi toolkit. Their overall WER is 15.6%, which is slightly worse than obtained by Narayanan and Wang [31]. To our knowledge, the results by Weninger *et al.* [53] are the best on the CHiME-2 dataset reported in the literature so far. As shown in Table IV, we have now pushed the performance to 10.63% average WER. This represents a 22.75% relative error reduction over [53], and the best result to date.

IV. CONCLUDING REMARKS

Moving forward, we plan to employ sequence models, such as the RNN or LSTM, for speech separation and acoustic modeling since they have been shown to capture temporal dynamics well [20]. How to effectively perform joint training on two RNNs is an open question.

Speech separation and recognition are two closely related problems. In this study, a joint training strategy is presented to integrate speech separation and acoustic modeling at the training stage. By further applying sequence training and run-time adaptation, the performance advantage of the joint modeling approach becomes even larger. Still, speech separation is done in a bottom-up fashion at the test stage. How to leverage top-down information, such as the knowledge from language models, to help speech separation at the test stage is an interesting direction for future research. We think that the joint modeling approach presented in this paper is an important step towards this goal, because language models are about the relations among words, or in a wider sense, among phonemes or states, while speech separation is commonly done in the time-frequency domain or at the signal level [51]. There is clearly a gap between them. The joint modeling approach utilizes acoustic models to bridge these two modules so that the information can be potentially flowed back and forth.

ACKNOWLEDGMENT

We would like to thank Yanzhang He for providing Kaldi training recipes and anonymous reviewers for their constructive comments.

REFERENCES

[1] D. Bagchi, M. Mandel, Z. Wang, Y. He, A. Plummer, and E. Fosler-Lussier, "Combining spectral feature mapping and multi-channel model-based source separation for noise-robust automatic speech recognition," in *Proc. IEEE Workshop Autom. Speech Recog. Understanding*, 2015.

[2] J. Chen, Y. Wang, and D. L. Wang, "A feature study for classification-based speech separation at low signal-to-noise ratios," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 12, pp. 1993–2002, Dec. 2014.

[3] 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 *Proc. Interspeech*, 2015, pp. 3274–3278.

[4] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?," in *Proc. Interspeech*, 2013, pp. 2992–2996.

[5] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 2133–2136.

[6] J. Du, Q. Wang, T. Gao, and Y. Xu, "Robust speech recognition with speech enhanced deep neural networks," in *Proc. Interspeech*, 2014, pp. 616–620.

[7] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[8] H. Erdogan, J. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 708–712.

[9] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 4375–4379.

[10] J. Geiger, F. Weninger, J. Gemmeke, M. Wollmer, B. Schuller, and G. Rigoll, "Memory-enhanced neural networks and NMF for robust ASR," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 6, pp. 1037–1046, Jun. 2014.

[11] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[12] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 6645–6649.

[13] K. Han, Y. He, D. Bagchi, E. Fosler-lussier, and D. L. Wang, "Deep neural network based spectral feature mapping for robust speech recognition," in *Proc. Interspeech*, 2015, pp. 2484–2488.

[14] K. Han, Y. Wang, D. L. Wang, W. S. Woods, and I. Merks, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 6, pp. 982–992, Jun. 2015.

[15] A. Hannun *et al.*, "Deepspeech: Scaling up end-to-end speech recognition," 2014, arXiv preprint arXiv: 1412.5567.

[16] E. Healy, S. Yoho, Y. Wang, and D. L. Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners," *J. Acoust. Soc. Amer.*, vol. 23, no. 6, pp. 3029–3038, 2013.

[17] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[18] J. Hershey, S. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Comput. Speech Lang.*, vol. 24, no. 1, pp. 45–66, 2010.

[19] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 4101–4104.

[22] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2009, pp. 3761–3764.

[23] B. Kollmeier and R. Koch, "Speech enhancement based on physiological and psychoacoustical models of modulation perception and binaural interaction," *J. Acoust. Soc. Amer.*, vol. 95, no. 3, pp. 1593–1602, 1994.

[24] B. Li and K. C. Sim, "A spectral masking approach to noise-robust speech recognition using deep neural networks," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 8, pp. 1296–1305, Aug. 2014.

[25] F. Li, P. Nidadavolu, and H. Hermansky, "A long, deep and wide artificial neural net for robust speech recognition in unknown noise," in *Proc. Interspeech*, 2014, pp. 358–362.

[26] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 4, pp. 745–777, Apr. 2014.

[27] M. Mimura, S. Sakai, and T. Kawahara, "Deep autoencoders augmented with phone-class feature for reverberant speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 4365–4369.

[28] S. Mirsamadi and J. Hansen, "A study on deep neural network acoustic model adaptation for robust far-field speech recognition," in *Proc. Interspeech*, 2015, pp. 2430–2434.

[29] 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. Process.*, vol. 22, no. 4, pp. 826–835, Apr. 2014.

[30] A. Narayanan and D. L. Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 2504–2508.

[31] A. Narayanan and D. L. Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 1, pp. 92–101, Jan. 2015.

[32] A. Narayanan and D. L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 7092–7096.

[33] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.

[34] D. Povey, A. Ghoshal, and G. Boulianne, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recog. Understanding*, 2011.

[35] S. J. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014, pp. 159–164.

[36] T. N. Sainath, B. Kingsbury, A. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *Proc. IEEE Workshop Autom. Speech Recog. Understanding*, 2013, pp. 297–302.

[37] T. N. Sainath *et al.*, "Deep convolutional neural networks for large-scale speech tasks," *Neural Netw.*, vol. 64, pp. 39–48, 2015.

[38] G. Saon and H. Soltau, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. IEEE Workshop Autom. Speech Recog. Understanding*, 2013, pp. 55–59.

[39] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE Workshop Autom. Speech Recog. Understanding*, 2011, pp. 24–29.

[40] M. L. Seltzer, "Robustness is dead! Long live robustness!," in *Proc. Reverb Challenge Workshop*, 2014 [Online]. Available: http://reverb2014.dereverberation.com/workshop/slides/mseltzer-reverb2014-keynote-share.pdf.

[41] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 7398–7402.

[42] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013, pp. 2345–2349.

[43] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: an overview of challenge systems and outcomes," in *Proc. IEEE Workshop Autom. Speech Recog. Understanding*, 2013, pp. 162–167.

[44] D. L. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. New York, NY, USA: Springer, 2005, pp. 181–197.

[45] D. L. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley-IEEE Press, 2006.

[46] Y. Wang, J. Chen, and D. L. Wang, "Deep neural network based supervised speech segregation generalizes to novel noises through large-scale training," Ohio State Univ., Columbus, OH, USA, OSU-CISRC-3/15-TR02, 2015.

[47] Y. Wang, K. Han, and D. L. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 2, pp. 270–279, Feb. 2013.

[48] Y. Wang, A. Misra, and K. Chin, "Time-frequency masking for large scale robust speech recognition," in *Proc. Interspeech*, 2015, pp. 2469–2473.

[49] Y. Wang, A. Narayanan, and D. L. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 12, pp. 1849–1858, Dec. 2014.

[50] Z.-Q. Wang and D. L. Wang, "Joint training of speech separation, filterbank and acoustic model for robust automatic speech recognition," in *Proc. Interspeech*, 2015, pp. 2839–2843.

[51] Z.-Q. Wang, Y. Zhao, and D. L. Wang, "Phoneme-specific speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2016, to be published.

[52] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 5532–5536.

[53] F. Weninger *et al.*, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, 2015, pp. 91–99.

[54] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2014, pp. 577–581.

[55] Y. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, Jul. 2013.

[56] 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. Process.*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[57] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks—Studies on speech recognition tasks," 2013, arXiv preprint arXiv:1301.3605.

[58] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 7893–7897.

Zhong-Qiu Wang (S'16) received the B.E. degree in computer science and technology from Harbin Institute of Technology, Harbin, China, in 2013. He is currently pursuing the Ph.D. degree at the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA. His research interests include robust automatic speech recognition, speech separation, machine learning, and deep learning.

DeLiang Wang, photograph and biography not provided at the time of publication.