Improving Robustness of Deep Neural Network Acoustic Models via Speech Separation and Joint Adaptive Training

Arun Narayanan, *Student Member, IEEE*, and DeLiang Wang, *Fellow, IEEE*

***Abstract*—Although deep neural network (DNN) acoustic models are known to be inherently noise robust, especially with matched training and testing data, the use of speech separation as a frontend and for deriving alternative feature representations has been shown to improve performance in challenging environments. We first present a supervised speech separation system that significantly improves automatic speech recognition (ASR) performance in realistic noise conditions. The system performs separation via ratio time-frequency masking; the ideal ratio mask (IRM) is estimated using DNNs. We then propose a framework that unifies separation and acoustic modeling via joint adaptive training. Since the modules for acoustic modeling and speech separation are implemented using DNNs, unification is done by introducing additional hidden layers with fixed weights and appropriate network architecture. On the CHiME-2 medium-large vocabulary ASR task, and with log mel spectral features as input to the acoustic model, an independently trained ratio masking frontend improves word error rates by 10.9% (relative) compared to the noisy baseline. In comparison, the jointly trained system improves performance by 14.4%. We also experiment with alternative feature representations to augment the standard log mel features, like the noise and speech estimates obtained from the separation module, and the standard feature set used for IRM estimation. Our best system obtains a word error rate of 15.4% (absolute), an improvement of 4.6 percentage points over the next best result on this corpus.**

***Index Terms*—CHiME-2, joint training, ratio masking, robust ASR, time-frequency masking.**

I. Introduction

THE introduction of deep neural network based acoustic models (DNN-AM) [11] has had a substantial impact on improving current ASR systems. Not only has DNN-AMs improved performance in relatively clean conditions [34], it has also boosted noise robustness [47], [35]. Our recent study investigates a feature based technique to improve ASR performance in noise [29]. In contrast to model adaptation, feature based ASR techniques keep the ASR backend unchanged during recognition, and perform speech/feature enhancement to handle noise. Such techniques can therefore be used with any ASR backend, be it Gaussian mixture model (GMM) based or DNN based.

The systems presented in [29] uses a time-frequency (T-F) masking based speech separation frontend. In T-F masking, the mixture energy at each T-F unit of the signal is weighted using a gain that is inversely proportional to the amount of noise it contains. The matrix of gains, typically called a time-frequency mask, is estimated directly from the input signal. The ideal ratio mask, which is defined as the ratio of the clean signal energy to the noisy signal energy at each T-F unit [37], is used in [29], and is estimated using DNNs. The results show that using such a speech separation system with DNN-AMs works reasonably well, but improvements compared to GMM based acoustic models (GMM-AM) are not as large.

With DNNs, it has also been observed that using a speech separation frontend does not always improve performance [35], [29]. This is especially true when log mel-spectral features, which have been shown to work better than cepstral features [25], [29], are used as input and when testing conditions are similar to the training conditions. A strategy commonly used with GMMs is retraining the ASR system using the enhanced features to minimize mismatch [44]. But frontends invariably introduce distortions, and with DNN-AMs retraining can sometimes negatively affect performance [35], [29]. An alternative strategy with GMM-AMs has been joint or adaptive training–the enhancement and the recognition modules are optimized jointly [2], [23], [15]. A probabilistic formulation of both the enhancement frontend and the GMM-AMs lends itself to expectation maximization (EM) style iterative training.

In adaptive training, the acoustic models are trained in a 'canonical' feature space. The frontend feature/model transformation morphs the original input features to this canonical space. This has two advantages. First, the canonical acoustic models are more amenable to adaptation. If the feature/model transforms are learned at test time, like in the case of vector Taylor series (VTS) model adaptation, transforming the canonical models performs better than transforming the models trained in the original input space [2], [15]. Second, the errors made by the frontend are modeled much better.

Manuscript received June 29, 2014; revised October 14, 2014; accepted November 11, 2014. Date of publication November 20, 2014; date of current version January 14, 2015. This work was supported in part by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-12-1-0130, in part by the National Institute on Deafness and Other Communication (NIDCD) under Grant R01 DC012048, and in part by the Ohio Supercomputer Center. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shinji Watanabe.

A. Narayanan was with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA. He is now with Google, Inc., Mountain View, CA 94043 USA (e-mail: narayaar@cse.ohio-state.edu).

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

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2014.2372314

An example of joint/adaptive training is VTS based noise adaptive training (VTS-NAT), which uses VTS model adaptation to deal with noise and channel mismatch [15]. VTS-NAT is an iterative two-stage EM algorithm. It first trains a GMM-AM using clean or noisy data. In the first stage of the algorithm, the parameters of the GMM-AM are kept fixed and the distortion parameters (noise and channel statistics) for VTS-model adaptation are learned from the data. In the second stage, the distortion parameters are kept fixed and the parameters of the GMM-AM are updated so as to maximize the likelihood of the data given the adapted models. The algorithm iterates between these two stages until convergence.

Most adaptive training algorithms are tailored for GMM-AMs. In the context of DNN-AMs, a speaker adaptive training strategy was proposed in [1], where a discriminative speaker code and a non-linear feature transform are learned to transform the original features to the representation in the first hidden layer of the DNN-AM. The acoustic model parameters remain unchanged during adaptive training. The canonical feature space is implicitly defined by the representation learned by the first hidden layer of the DNN-AM. Such joint adaptive training strategies, to the best of our knowledge, have not been proposed in the context of noise robustness.

This paper proposes a strategy for jointly training a ratio-masking based speech separation frontend and a DNN-AM. We call the proposed framework *joint adaptive training* for DNN-AMs (DNN-JAT). Even though the model parameters are not updated during test time, we call the proposed training strategy 'adaptive' to differentiate it from the simpler approach of retraining the acoustic model using the transformed features without modifying the transformation in itself–an approach that has been classified as joint training [20]. Joint training schemes have been used successfully with GMM-AMs [44], but not so much with DNN-AMs [35], [7].

Since separation as a frontend gives limited performance gains when using DNN-AMs, we also study how ASR performance can be improved by augmenting the traditional log mel spectra with additional features when training the acoustic models. For example, noise-aware training (NAT) proposed in [35] uses a crude estimate of noise obtained by averaging the first and the last few frames of each utterance as an additional input. This improved performance on the Aurora-4 corpus [31] by 3.9% (relative). It was shown in [30] that instead of using such an estimate, speech separation can be used to obtain a more accurate estimate of noise. Additionally, features like speech estimate and residual noise estimate, which can both be derived from speech separation, are shown to further improve performance. We analyze the performance that can be obtained using these alternative feature representations in the context of joint adaptive training. We also study whether the feature set that is traditionally used as input by supervised mask estimation algorithms adds any further information when they are used for acoustic modeling. We note that [30] presents a preliminary version of this work. Apart from studying additional features for acoustic modeling, this article presents extended analysis and results compared to [30].

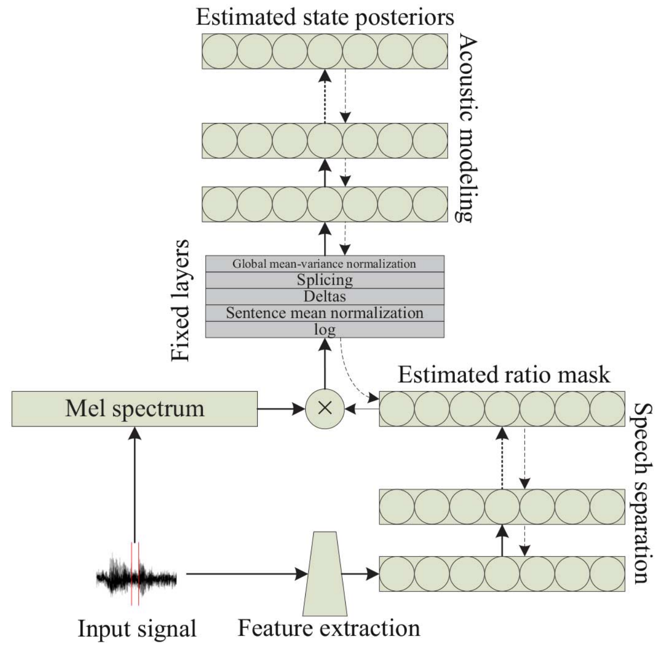

Fig. 1. (Color online) A block diagram of joint noise adaptive training. The two main components of the system, speech separation and acoustic modeling, are shown along with how they are joined into a single framework.

This paper is organized as follows. The proposed systems are presented in Section II, followed by an in-depth evaluation in Section III. We conclude with a discussion in Section IV.

II. System Description

For the DNN-JAT system that we present here, it is assumed that there is not a lot of channel mismatch between training and testing, and that speech separation primarily addresses background noise. Channel mismatch, like those caused by microphone characteristics and room reverberation, is typically addressed by learning the channel impulse response within a model based probabilistic framework when using GMMs [21]. With DNNs, channel effects can, arguably, be handled by collecting more training data (see, e.g., [47]). For example, additional recordings can be made using a new cell phone with previously unseen microphone characteristics if an ASR system has to be deployed on it. Alternatively, feature mapping can also be used [29]. Unlike channel mismatch, background noise can be more uncertain and harder to deal with.

We present joint adaptive training in the context of a speech separation frontend that handles background noise via time-frequency masking. A block diagram of the proposed system is shown in Fig. 1. The goal of the frontend is to estimate a time-frequency mask, an ideal ratio mask in our case, which is used to weight the energy at each T-F bin of the noisy spectra. The ideal mask is defined as the ratio of the speech to mixture energy at each T-F bin, assuming that speech and noise are uncorrelated [35], [7]:

$$\mathbf{M}_t^{(r)}(c) = \frac{\mathbf{X}_t(c)}{\mathbf{X}_t(c) + \mathbf{N}_t(c)} \quad (1)$$

Here, $\mathbf{M}^{(\mathbf{r})}$ is the IRM, $\mathbf{X}$ and $\mathbf{N}$ are the clean and the noise (mel) spectrogram, respectively, and t and c index time and frequency; and we assume that the mixture energy $\mathbf{Y}_t(c) = \mathbf{X}_t(c) + \mathbf{N}_t(c)$. It is easy to see that, with perfect estimation, masking using the IRM restores the clean spectra. The components of the system are described in detail below.

A. Speech separation

As mentioned, speech separation is done in a supervised fashion via ratio masking using DNNs. DNNs have now become the standard in supervised speech separation [24], [29], [44], [19], [45], [41], [13]. Apart from mask estimation, binary [42] or ratio [27], [29], [13], they have also been used in other ways. In [24], recurrent neural networks (RNNs) are used to denoise cepstral features. Weninger *et al.* use bidirectional long short-term memories, a more sophisticated variety of recurrent nets, for denoising and show that they work better than plain RNNs [44]. Xu *et al.* use DNNs to estimate the log power spectral coefficients and found them to work better than statistical speech enhancement algorithms [45]. In [41], it was shown that, in the context of speech separation, ratio masking is perhaps the most effective way to handle background noise when using DNNs. This is largely because 1) ratio masks are inherently normalized and bounded targets, and 2) denoising via masking is less sensitive to estimation errors.

We perform IRM estimation in the 26-channel mel spectrogram domain spanning frequencies in the range 50 Hz to 7 kHz. A window size of 20 msec and a hop size of 10 msec are used. The IRM is estimated using a system similar to the one presented in [29], but is simplified so that it can be easily incorporated into the joint framework. The following feature set is extracted at every time frame from the noisy input signal:

- 13 dimensional relative spectral filtered perceptual linear predictive cepstral coefficients (RASTA-PLP) [10]. The features for 7 contiguous frames are spliced together to add context.
- Amplitude modulation spectrograms (AMS) [18]. 15-dimensional AMS features are extracted separately for each of the 26 frequency bands in the mel spectrogram. They are then concatenated to form the input feature for a time-frame.
- 31 dimensional broadband and narrowband mel frequency cepstral coefficients (MFCC). Narrowband MFCCs, which are extracted using an analysis window of 200 msec [5], add a lot more context than broadband MFCCs, which use a 20 msec window. Similar to RASTA-PLPs, the MFCC features of 7 contiguous frames are spliced together to form the input representation.

The context size while forming features is fixed to 7 to keep a check on the input dimensionality. The above features when concatenated together forms a 915-dimensional ($13 \times 7 + 15 \times 26 + 31 \times 2 \times 7$) input feature, which is fed to a 3 hidden layer DNN that simultaneously estimates the IRM for all 26 frequency channels. Each hidden layer has 1024 nodes. Through cross-validation, we found 3 layers and 1024 nodes to be sufficient; decreasing the number of layers increased estimation error, and increasing both the number of layers and the number of nodes did not significantly improve performance. The DNN is trained with a dropout rate of 0.3 for both the input and the hidden layers. The hidden nodes use rectified linear activations (ReLU) [26] and the output nodes sigmoidal activations. The weights are learned using mini-batch stochastic gradient descent with AdaGrad [8] and momentum. The momentum is linearly increased from 0.1 to 0.5 over the first 5 epochs after which it is set to 0.9. Mini-batch size is set to 256. The weights are initialized to small random values. We also normalize the L_2 norm of incoming weights of the hidden nodes to 1 after each update [12]. The DNN is trained for 50 epochs to minimize the cross-entropy error criterion. The learning rate is set to 0.01 for the first 10 epochs, 0.005 for the next 20 epochs, and 0.001 for the last 20 epochs.

Note that the IRM estimation algorithm in [29] estimates masks at the subband and the fullband levels, and then combines these estimates over a window in the post-processing stage to explicitly incorporate context. The proposed system, on the other hand, incorporates context at the feature level.

B. Acoustic Modeling

Motivated by the studies in [35], [30], we experiment with multiple input feature representations apart from the commonly used noisy log mel spectrogram (NMS). Some of these features are derived using the IRM estimated by the speech separation module. Given an estimated IRM, $\widehat{\mathbf{M}}^{(\mathbf{r})}$, we can obtain an estimate of both speech and noise as follows:

$$\widehat{\mathbf{N}}_t = (1 - \widehat{\mathbf{M}}_t^{(\mathbf{r})}) \odot \mathbf{Y}_t, \tag{2}$$

$$\widehat{\mathbf{X}}_t = (\widehat{\mathbf{M}}_t^{(\mathbf{r})})^\alpha \odot \mathbf{Y}_t. \tag{3}$$

Here, $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{N}}$ correspond to estimates of the clean and noise mel spectrogram. A tunable parameter α (≤ 1) exponentially scales-up IRM estimates to reduce the speech-distortion introduced by masking at the expense of retaining some noise. Note that to obtain noise estimate, the mask is not scaled. Exponentiation is done point-wise. $\odot$ denotes point-wise multiplication.

We use the following input feature representations based on the above estimates:

- NMS: Noisy log-compressed mel spectrogram along with deltas and double deltas. After sentence level mean normalization, features from 11 contiguous time frames are spliced together, as is commonly done, to incorporate context. The dimensionality of this feature is 858.
- NMS + SNE: This feature uses the NMS feature and appends it with a 'stationary' noise estimate obtained by averaging the first and the last 15 frames of the noisy mel spectrogram. This feature set replicates the system proposed in [35]. The dimensionality of this feature is 884.
- NMS + DNE: This feature uses the NMS feature and appends it with a 'dynamic' noise estimate obtained using the estimated IRM. The noise estimate is obtained as per Eq. (2), and is smoothed using a 9th order ARMA filter [4]. A 9th order filter roughly smooths over a window of 19 time-frames, which is the same as the number of frames needed to create an input feature representation for NMS features after the context for deltas and double deltas are

taken into consideration. The dimensionality of this feature is 884.

- NMS + DNE + SE: Same as above, but also appends an estimate of speech obtained by smoothing $\widehat{\mathbf{X}}$ in Eq. (3) using a 2nd order ARMA filter. 2nd order filters were suggested in [4], [5]. The dimensionality of this feature is 910.
- NMS + fIRM: The NMS feature appended with the features defined in Section II-A for IRM estimation. The dimensionality of this feature is 1773.
- NMS + fIRM + DNE + SE: Same as above, but with smoothed speech and noise estimates as additional features. The dimensionality of this feature is 1825.

The DNN-AMs consist of 7 hidden layers, each with 2048 nodes, and are trained in a way similar to the DNNs used for IRM estimation. Dropout rate is fixed to 0.3 for all layers. The hidden nodes use ReLU activations and the output layer uses softmax activation. Mini-batch stochastic gradient descent with AdaGrad and momentum is used for weight optimization, and the L_2 norm of the weights of each node is normalized to 1. The network is trained for 50 epochs. The learning rate is set to 0.005 for the first 30 epochs and 0.001 for the final 20 epochs. Momentum is linearly updated from 0.1 to 0.5 over the first 5 epochs, and is set to 0.9 starting the 6th. The weights are initialized to random values; no pre-training is used. A subset of the development set was chosen for cross-validation during training, to ensure that the models converge and that they do not overfit to the training data. We found that both AdaGrad and dropout are necessary when training 7-layer ReLU models as they tend to overfit easily. In general, obtaining an appropriate hyper-parameter setting for the simpler stochastic gradient descent based optimization was found to be especially hard.

C. Joint Adaptive Training

The main goal behind joint adaptive training is to unify separation and acoustic modeling. Typically, the output of separation undergoes further processing before it is fed to the acoustic model. In our joint system, we model these processing steps as *fixed* hidden layers of a single deep network. They are shown in darker gray in Fig. 1 and includes operations like log-compression, feature normalization, delta calculation, and feature splicing. Interestingly, all of these operations can be performed within a DNN framework using appropriate weights and/or network architecture. For example, delta features can be calculated using a linear activation layer with weights as below [39]:

$$\begin{bmatrix} \mathbf{O}_t \\ \Delta\mathbf{O}_t \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \mathbf{I} & 0 \\ -\mathbf{I} & 0 & \mathbf{I} \end{bmatrix}}_{\mathbf{W}_\Delta} \begin{bmatrix} \mathbf{O}_{t-1} \\ \mathbf{O}_t \\ \mathbf{O}_{t+1} \end{bmatrix}. \quad (4)$$

Here, $\mathbf{O}_t$ is the static feature at time t, $\mathbf{I}$ is the identity matrix, and $\mathbf{W}_\Delta$ is the weight matrix of the hidden layer. The above formulation calculates deltas over a window of 3 frames, but it can be extended trivially to more than 3 frames and for calculating double-deltas. Further, the connections from the preceding layer can be modified so that this layer receives static features from multiple, consecutive time frames as is necessary for delta calculation. The other fixed layers can be modeled in a similar fashion. We outline the steps briefly below, assuming an NMS based input feature representation for the acoustic model:

1) Given the noisy mel spectrogram, $\mathbf{Y}$, and the estimated IRM, $\widehat{\mathbf{M}}^{(\mathbf{r})}$, we start with masking to obtain an estimate of the clean (log compressed) mel spectrogram, $\widehat{\mathbf{X}}$, using Eq. (3).
2) Next, we append deltas and double-deltas by multiplying the features with the delta weight matrix, $\mathbf{W}_\Delta$. Since delta components are a temporal feature, the static features from contiguous time-frames are spliced together before multiplying them with $\mathbf{W}_\Delta$.
3) Finally, we do sentence level mean and global variance normalization followed by a another splicing operation to form the input to the DNN-AM. The mean and the variance are re-estimated after every epoch.

With the above formulation, it is easy to see that the two modules can now be trained jointly; with the fixed hidden layers cast as a 'neural network,' the error gradients from the DNN-AM can flow through them back to the speech separation module. While training such a system, we initialize both the acoustic model and the IRM estimator using the independently trained DNNs. The trainable weights of both networks are then tuned together for a few additional epochs. The number of additional epochs is set using cross-validation. In our recipe, we run the joint DNN for 10 epochs; the WER on the development set is calculated after every epoch and the model that gives the least WER is chosen. We found that DNN-JAT converges within the 7-9 epochs in most of our experiments. Training the joint DNN with randomly initialized weights is an extremely difficult optimization task; it is unreasonable to expect a randomly initialized network to implicitly learn an appropriate masking function. An important detail that needs to be considered during training is the gradient of the masking operation. This gradient involves a term that is inversely proportional to $\widehat{\mathbf{M}}_t^{(\mathbf{r})}$ because of log compression. This value can easily dominate the gradients that reach the mask estimation module of the joint framework, especially when the estimated IRM values are close to zero. To prevent such a scenario, we explicitly clip the gradients so that they do not exceed a preset maximum. Via cross-validation, we found this parameter to be not too critical in terms of final performance; most values between 2 and 100 seem to work reasonably well. In our experiments, we set it to 5. The other hyperparameters during joint adaptive training are the same as those used during the final epoch of independent training.

III. Results

A. Experimental Setup

The proposed system is evaluated on the CHiME-2 corpus [40], which is a medium-large vocabulary corpus based on WSJ0-5k. The corpus simulates a family living room; the utterances are reverberant, and are artificially mixed with real recorded noise at signal-to-noise ratios (SNRs) in the range $[-6, 9]$ dB. Even though the corpus is binaural, our system is monaural; we simply average the left and right ear recordings for all experiments[1]. The IRM estimator needs noisy and the corresponding clean recordings to define the targets at the time of training. Since such recordings are not provided with the corpus, we artificially mix the reverberant noise-free utterances in the training set with randomly selected segments of the noise recordings provided with the corpus. The fraction of recordings at each SNR is the same as in the official noisy-reverberant training set. This new set is used *only* to train the IRM estimator when it is trained independent of the DNN-AM.

In order to obtain senone (or tied-triphone state) labels for training the DNN-AM, a maximum-likelihood trained GMM-HMM system is used. The clean training set of WSJ0 is used to train and subsequently align the utterances. Based on the pruning parameters, the system ended up with 3298 senones. The DNN-AMs are trained using the official noisy-reverberant training set. The same set is also used for joint adaptive training–after initialization the joint model is optimized solely to improve the ASR criterion (i.e., cross-entropy).

All systems are implemented using the HTK Toolkit [46]. During decoding, we use a bigram language model unless stated otherwise, and the CMU pronunciation dictionary. The standard GMM-based Viterbi decoder implemented in HTK is modified to function as a hybrid decoder.

B. Development Set Results

We first present results on the development set (si_dt_05) of CHiME-2. Experiments on this set are used to 1) choose an appropriate value for α in Eq. (3), 2) decide whether or not to retrain the system using masked features, 3) finalize what additional features to include as input to the DNN-AM, and 4) finalize the setup of the joint adaptive training framework. All parameters are chosen to optimize performance on the development set. Note that the recipe for training the DNNs is fixed as described before. No attempt was made to optimize it separately for each subtask.

Choosing an Appropriate Value for α : The parameter α controls how the estimated ratio mask is scaled. Values > 1 would shrink the estimated IRM closer to 0, thereby increasing noise suppression while at the same time introducing some distortion to speech. On the other hand, values < 1 will scale IRM estimates towards 1, reducing speech distortion at the expense of leaving some noise in the masked mel-spectrogram. The chosen value reflects a tradeoff between residual noise and speech distortion. The WER obtained for various values of α are shown in Table I. The NMS feature is used to train the DNN-AM for the systems described in this section.

Our baseline, which is to directly use the noisy features as input, corresponds to the system that uses a value of 0 for α. An average WER of 28.9% is obtained using such a system. With masking used in the traditional fashion, i.e., with $\alpha = 1$, an average WER of 27.8% is obtained; an improvement of 1.1 percentage points over the baseline. But more interestingly, using a smaller value for α further improves performance. Although most values between 0 and 1 give improvements, the value 0.5 is found to be the best; $\alpha = 0.5$ improves the WER by 2.3 percentage points compared to the baseline. As has been noted in other studies, using separation as a frontend does not always yield performance gains with DNN-AMs [35], [29]. This is perhaps because of the distortions introduced in speech while attempting to remove noise. Reducing distortion at the expense of retaining residual noise seems more helpful when using DNN-AMs. Setting α to a value < 1 also has the added advantage of shrinking the gradients of the log operation during joint adaptive training. For the rest of the systems that we present, unless otherwise stated, the value of α will be set to 0.5 while performing masking.

TABLE I
WER ON THE DEVELOPMENT SET (SI_DT_05) OF THE CHiME-2 CORPUS FOR VARIOUS VALUES OF α. THE NMS FEATURE REPRESENTATION IS USED. NOTE THAT $\alpha = 0$ CORRESPONDS TO DIRECTLY USING NOISY FEATURES, WHICH FORMS THE BASELINE. THE BEST PERFORMANCE IN EACH CONDITION IS MARKED IN **BOLD**

α	si_dt_05						
	-6 dB	**-3 dB**	**0 dB**	**3 dB**	**6 dB**	**9 dB**	**Average**
0.0	43.3	34.3	29.0	25.4	21.6	19.6	28.9
0.3	39.7	31.8	27.1	23.4	20.6	**18.7**	26.9
0.5	**39.4**	31.2	**27.0**	**23.0**	**20.4**	18.8	**26.6**
0.7	**39.4**	**30.8**	**27.0**	23.7	20.5	19.2	26.8
1.0	40.9	32.6	27.6	24.2	21.5	19.9	27.8

Next, we experiment with the conventional joint training framework which simply retrains the acoustic models using masked features. We obtain masked features for training using Eq. (3) with α set to 0.5. As is typically done, the same training recipe as was used to train the original acoustic models is used but with masked features instead of the noisy ones. The cross-validation set is used to ensure that there was no overfitting. The WERs obtained after retraining is shown in Table II. As before, we measure WERs as a function of α. When α $=$ 0.5, the value used for obtaining masked features while training, we get an average WER of 27.3 percent. It improves to 26.6 when α is set to 0.7. The difference in performance is mainly in the lower SNR conditions. In general, it is observed that when α is set to a value between 0.5 and 1, better results are obtained. While it is surprising that α s other than 0.5 perform better, it may be because, when trained using masked features, it is better to remove more noise during testing. From the results it is clear that retraining using masked data does not outperform the system that is trained on noisy data, when using masked features. The noisy dataset contains signals at (long-term) SNRs ranging from -6 dB to 9 dB. And we can see from the results that, as the input SNR of the signal improves, the WER decreases. When we use masked features with a carefully tuned α, it is likely that some noise gets removed without significantly distorting speech. In other words, masking improves SNR of the input signal without introducing additional speech distortions. We believe this is the reason why using masked features with noisy acoustic models yields similar performance as the retrained models. Based on these results, the acoustic models are not retrained using masked features in the rest of the experiments.

From the presented results, it is clear that estimated IRMs significantly improve performance. To check whether masking

[1]The target azimuth is always 0°, so averaging the left and right ear recordings improves SNR.

TABLE II
WER ON THE DEVELOPMENT SET (SI_DT_05) OF THE CHIME-2 CORPUS USING ACOUSTIC MODELS RETRAINED ON MASKED DATA, WITH α SET TO 0.5. THE NMS FEATURE REPRESENTATION IS USED. THE BEST PERFORMANCE IN EACH CONDITION IS MARKED IN **BOLD**

α	si_dt_05						
	-6 dB	**-3 dB**	**0 dB**	**3 dB**	**6 dB**	**9 dB**	**Average**
0.3	43.9	34.7	29.1	25.7	21.1	19.4	29.0
0.5	41.0	32.7	27.6	23.8	**20.3**	18.5	27.3
0.7	**39.8**	**31.5**	27.1	**23.0**	**20.3**	**18.3**	**26.6**
1.0	39.7	31.7	**26.9**	23.3	**20.3**	18.6	26.7
1.5	43.1	33.7	28.9	24.5	21.8	19.6	28.6

TABLE III
WER ON THE DEVELOPMENT SET (SI_DT_05) OF THE CHIME-2 CORPUS WHEN THE SAME DATA IS USED TO TRAIN THE MASK ESTIMATOR AND THE ACOUSTIC MODEL

α	si_dt_05						
	-6 dB	**-3 dB**	**0 dB**	**3 dB**	**6 dB**	**9 dB**	**Average**
0.0	42.7	33.9	28.9	25.2	21.7	19.2	28.6
0.5	38.4	30.5	26.5	22.9	20.8	18.9	26.3

TABLE IV
WER ON THE DEVELOPMENT SET (SI_DT_05) OF THE CHIME-2 CORPUS USING AN 8-LAYER DNN FOR ACOUSTIC MODELING

α	si_dt_05						
	-6 dB	**-3 dB**	**0 dB**	**3 dB**	**6 dB**	**9 dB**	**Average**
0.0	43.2	34.7	29.4	25.1	21.9	19.7	29.0
0.5	39.5	31.4	26.7	23.3	20.6	19.4	26.8

helps because it is trained on data that the acoustic model has not seen, we trained the acoustic model using the same data that was used to train the IRM estimator. Results are shown in Table III. As can be seen, the obtained performance is very similar to those obtained using the original system (see Table I). Masking still improves performance when the same data is used to train the DNN-AM.

Finally, to confirm that improvements due to masking is not because the full model now has more tunable parameters, we trained an 8 hidden layer DNN-AM instead of a 7 layer model that was used to generate the results in Table I. Adding another layer to the DNN-AM adds more parameters than what gets added by having a masking frontend. From the results, which are shown in Table IV, we can see that simply making the system more complex does not improve performance, and that masking still provides gains. Based on these results we can conclude that masking, by itself, adds value to the system. It is likely because masking, conceptually, is an apt frontend 'feature transformation' to remove noise; it is hard to discover it by simply using more data and (or) additional parameters.

Performance with Additional Features: Given that masking improves performance, we now experiment with the additional features described in Section II-B. The results are shown in Table V.

It can be seen from the results that when using SNE, no significant improvements are obtained over the baseline in Table I, with and without masking. Using DNE performs slightly better. This is expected as the noises in CHiME-2 are highly non-stationary. It is unlikely that the first and the last 15 frames are representative enough of the noise characteristics in the middle of an utterance. Adding speech estimate in addition to DNE performs the best, improving performance by 1.7 and 0.7 percentage points, respectively, compared to the baseline. When the features used for IRM estimation are appended to NMS, a WER of 26.3% is obtained which is close to the performance obtained after masking when only the NMS feature is used. After enhancing the NMS feature, performance of ' NMS + fIRM' further improves to 25.2 percent. Adding speech and noise estimates derived from the separation frontend does not improve performance any further. It is worth noting that using only the 'fIRM' features resulted in an average WER of 37.1% (not shown in the table), which is worse than the systems presented here.

From the results, we can conclude that: 1) Masking consistently helps improve performance in all conditions. 2) Using fIRM features in combination with the traditional NMS features performs the best; adding speech and noise estimates further does not help reduce WER.

Joint Adaptive Training: We now apply joint adaptive training to the following systems:

- DNN-AM that uses the 'NMS' feature as its input. The IRM estimated by the separation module enhances the NMS features using Eq. (3) which is then used as input to the DNN-AM. Joint adaptive training is used to further enhance the NMS feature and adapt the DNN-AM (NMS + JAT)[2].
- DNN-AM that uses the 'NMS + DNE + SE' feature as its input. The noise and speech estimates are obtained using the initial IRM estimator that is trained independent of the DNN-AM. Joint adaptive training is used to enhance the NMS feature and the DNN-AM (NMS + DNE + SE + JAT(1)).
- DNN-AM that uses the 'NMS + DNE + SE' feature as its input. Joint adaptive training is used to enhance all three input features and the acoustic model (NMS + DNE + SE + JAT(2)).
- DNN-AM that uses the 'NMS + fIRM' feature as its input. Joint adaptive training is used to enhance the NMS feature and the DNN-AM (NMS + fIRM + JAT).

All joint systems improve performance over their independently trained counterparts, as can be seen from the results in Table VI. As before, using additional features continues to perform better than only using the NMS feature as input to the DNN-AM. Interestingly, using the noise and speech estimates obtained from the initial estimate of the IRM ('NMS + DNE + SE + JAT(1)' in the table) performs better than recalculating them from the mask estimated by the joint system ('NMS + DNE + SE + JAT(2)' in the table). The reason for this becomes clear when we look at the masks generated by the joint system. An example is shown in Fig. 2. As can be seen, the mask generated by the jointly trained model (Figs. 2(f) and (h)) attenuates noise a lot more than those generated by the independently trained models (Figs. 2(e) and (g)). Since joint adaptive training

[2]Whenever the suffix ' + JAT' is used, it means that the NMS component of the feature is enhanced via masking and that the mask is obtained through joint adaptive training.

TABLE V
WER ON THE DEVELOPMENT SET (SI_DT_05) OF THE CHIME-2 CORPUS USING ADDITIONAL FEATURES

System	si_dt_05						
	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
NMS + SNE	43.2	34.0	29.1	25.0	21.7	19.0	28.7
Masked NMS + SNE	38.8	30.9	26.8	22.8	20.3	18.4	26.3
NMS + DNE	42.1	33.8	28.5	24.7	21.5	19.1	28.3
Masked NMS + DNE	38.3	30.3	26.5	22.7	20.1	18.4	26.0
NMS + DNE + SE	40.3	33.0	27.4	23.1	21.0	18.2	27.2
Masked NMS + DNE + SE	38.3	31.0	26.5	22.4	19.7	17.8	25.9
NMS + fIRM	39.6	31.7	26.7	22.5	19.9	17.5	26.3
Masked NMS + fIRM	37.9	29.9	25.2	21.9	19.4	17.2	25.2
NMS + fIRM + DNE + SE	40.7	32.9	27.3	23.2	20.2	18.1	27.0
Masked NMS + fIRM + DNE + SE	37.9	29.9	25.2	21.6	19.1	17.5	25.2

TABLE VI
WER ON THE DEVELOPMENT SET (SI_DT_05) OF THE CHIME-2 CORPUS USING JOINT ADAPTIVE TRAINING

System	si_dt_05						
	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
NMS + JAT	37.9	30.5	26.1	22.0	20.0	18.2	25.8
NMS + DNE + SE + JAT(1)	37.2	29.5	25.8	21.5	19.4	17.5	25.1
NMS + DNE + SE + JAT(2)	37.4	29.8	25.6	21.9	19.8	17.8	25.4
NMS + fIRM + JAT	37.1	29.4	24.9	20.9	18.8	17.3	24.7

improves IRM estimation by optimizing the ASR loss, the resultant mask preserves spectro-temporal patterns that are most important for recognition. The independently trained model, on the other hand, tries to estimate a mask that can transform the noisy spectrogram to the clean spectrogram as closely as possible. Note that these two criteria are neither mutually exclusive nor the same: Even signals that do not 'sound' the same as the underlying clean speech signal may be perfectly recognizable by an ASR system just as binary masked noise signals are recognizable to both humans and machines [17], [28]. Given that the IRM estimated by the joint system retains only the essential spectro-temporal patterns, the quality of the speech and noise estimate obtained using it will not necessarily be as good as those obtained from the original estimated IRM. This is likely the reason why 'NMS + DNE + SE + JAT(1)' performs slightly better than 'NMS + DNE + SE + JAT(2)'. Consistent with the results presented in the previous section, 'NMS + fIRM + JAT' performs the best with an average WER of 24.7% on the development set.

C. Test Set Results

We now report performance on the test set using the best performing systems on the development set. The first set of results is generated using a bigram language model, which was also used for evaluations on the development set. The results are shown in Table VII. As can be seen, our baseline system trained using the NMS features gives an average WER of 25.0 percent, which is in itself better than the previous best results on this corpus using a GMM-HMM system by 6.7% (relative) [44]. The system in [44] uses bidirectional long short-term memory based feature enhancement and a discriminatively trained, speaker adapted GMM-HMM system. Using the estimated ratio mask to enhance the noisy speech improves performance by 2.4 percentage points compared to the baseline ('Masked

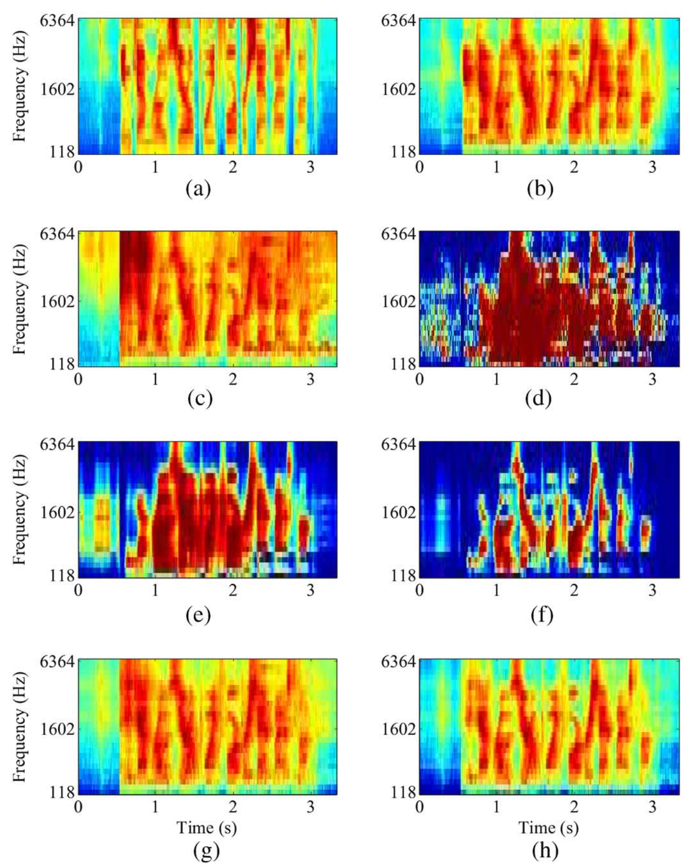

Fig. 2. (Color online) Example of masking: (a) Log-mel spectrogram of a clean utterance. (b) Log-mel spectrogram of the utterance with reverberation. (c) Log-mel spectrogram of the utterance with noise and reverberation. The SNR, with respect to reverberant speech, is -3 dB. (d) The ideal ratio mask. (e) The IRM estimated by the independently trained mask estimator. (f) The IRM estimated by the joint model. (g) The noisy log-mel spectrogram enhanced using the estimated IRM. (h) The noisy log-mel spectrogram enhanced using the IRM estimated by the joint model.

TABLE VII
WER ON THE TEST SET (SI_ET_05) OF THE CHIME-2 CORPUS USING A BIGRAM LANGUAGE MODEL

System	si_et_05						
	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
NMS	37.9	30.1	25.9	21.1	18.3	16.5	25.0
Masked NMS	33.0	26.6	23.9	19.5	16.6	15.8	22.6
NMS + JAT	32.1	25.6	23.1	18.6	16.1	15.1	21.7
Masked NMS + DNE + SE	33.2	26.7	22.4	18.7	15.4	14.5	21.8
NMS + DNE + SE + JAT(1)	31.4	24.8	21.4	18.1	15.1	14.1	20.8
NMS + fIRM	32.9	25.6	21.8	18.5	15.5	14.2	21.4
Masked NMS + fIRM	31.2	24.9	20.7	18.0	15.0	13.6	20.6
NMS + fIRM + JAT	30.5	24.5	21.2	17.5	14.4	13.5	20.3

TABLE VIII
WER ON THE TEST SET (SI_ET_05) OF THE CHIME-2 CORPUS USING A TRIGRAM LANGUAGE MODEL

System	si_et_05						
	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
NMS	31.8	25.0	21.0	16.5	13.9	12.9	20.2
Masked NMS	27.2	21.6	18.9	15.6	13.0	11.6	18.0
NMS + JAT	26.0	20.8	18.1	14.8	12.3	11.5	17.3
Masked NMS + DNE + SE	26.2	20.7	18.0	14.7	11.7	11.0	17.0
NMS + DNE + SE + JAT(1)	25.6	19.6	16.8	13.8	10.7	10.6	16.2
NMS + fIRM	26.6	20.7	16.8	14.2	11.3	10.4	16.7
Masked NMS + fIRM	25.6	19.9	15.4	13.4	10.8	10.1	15.9
NMS + fIRM + JAT	25.1	19.2	15.1	12.8	10.5	9.5	15.4

NMS' in the table). Note that the ASR models are not retrained using masked speech. 'NMS + JAT', which jointly trains the IRM estimator and the DNN-AM that is trained on the NMS feature, improves performance by another 0.9% compared to 'Masked NMS'. The 'NMS + DNE + SE' system obtains WERs similar to 'NMS + JAT' on average, after the NMS features are enhanced using the estimated IRM. NMS + JAT seems to do better in low SNR conditions (-6 dB and -3 dB) and 'NMS + DNE + SE' works better in high SNR conditions (0 dB, 6 dB and 9 dB). 'NMS + DNE + SE + JAT(1)' produces an average WER of 20.8 percent, 1 percentage point better than the corresponding system that does not use joint adaptive training. NMS + fIRM performs comparably to 'NMS + DNE + SE + JAT(1)', obtaining an average WER of 21.4 percent. Masking and joint adaptive training improves performance to 20.6% and 20.3 percent, respectively. 'NMS + fIRM + JAT' obtains the lowest WER and is 18.8% (relative) better than our baseline (19.5% (relative) better at -6 dB).

Table VIII shows the performance obtained when a trigram language model is used. Performance of each system improves by roughly 5 percentage points. The baseline WER improves from 25% to 20.2 percent. 'NMS + DNE + SE + JAT(1)' obtains a WER of 16.2 percent. 'NMS + fIRM + JAT' improves it further to 15.4 percent, 23.8% better (relative) than the baseline. Upon comparing performance at different SNR conditions, it can be inferred that the final system improves SNR by 3 to 6 dB in terms of ASR performance: The WERs that 'NMS + fIRM + JAT' obtains at -6 dB and 3 dB are similar to those obtained by the baseline at -3 dB and 9 dB, respectively.

IV. DISCUSSION

We list a few results from literature on the CHiME-2 corpus in Table IX. The systems described in [38] and [44] are GMM-HMM systems, and use a feature enhancement frontend, along with discriminative features and discriminative acoustic modeling [32]. As can be seen, the system proposed in this paper outperforms them by a large margin. The recently proposed system in [3] uses recurrent nets and NMS features without any enhancement. The joint system outperforms it by 7.4 percentage points, highlighting the utility of the proposed robust feature frontend. It is likely that RNNs and discriminative training strategies can further improve performance of the proposed framework. The system in [9] proposes model combination–it uses an NMF feature enhancement frontend and discriminatively trained GMM-AMs, and combines it with a long short-term memory based acoustic model. It obtains an average WER of 20.0, 4.6 percentage points worse than the proposed system. We note that, the system described in this paper, and those in [44] and [3] make use of aligned clean and noisy data which was disallowed in the original CHiME-2 challenge. To overcome this limitation, it will be worth investigating in the future whether mask estimators trained using other speech corpora can be used for initializing the joint systems; joint adaptive training can then be performed only using the noisy utterances.

As we noted before, the masks generated by the jointly trained model attenuate noise a lot more than those generated by the independently trained models, while preserving spectro-temporal patterns that are important for recognition. There appears to be a disconnect between the objectives commonly used for mask estimation (SNR improvement) and ASR (WER reduction), and our past work has shown that SNR improvements and ASR improvements (or speech intelligibility) are not fully correlated [28]. Joint adaptive training, on the other hand, directly optimizes a criterion that is important to improve ASR. Since ASR and speech intelligibility tend to correlate [28], such joint adaptive training schemes may provide an alternative, more appropriate criterion to optimize for speech separation algorithms that

TABLE IX
A FEW PUBLISHED RESULTS ON THE CHiME-2 TEST SET (SI_ET_05). ALL SYSTEMS USE A TRIGRAM LANGUAGE MODEL, EXCEPT FOR [38] WHICH USES DISCRIMINATIVE LANGUAGE MODELS AND MINIMUM BAYES RISK DECODING

System	si_et_05						
	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
Discriminative features, models, decoding [38]	44.1	35.5	28.1	21.2	17.4	14.8	26.9
BLSTM + discriminative models [44]	42.7	33.9	27.5	21.8	18.4	16.2	26.7
RNN [3]	38.1	29.1	23.0	17.9	15.0	13.6	22.8
LSTM + discriminative GMMs + NMF [9]	33.8	25.7	20.3	15.5	13.0	11.9	20.0
DNN-JAT (this paper)	25.1	19.2	15.1	12.8	10.5	9.5	15.4

focus on intelligibility. Moreover, the proposed joint adaptive training strategy is quite flexible, and can easily be adapted to work with other separation frontends like feature mapping [24], [44], [29].

The fact that an IRM estimator can be optimized using an ASR criterion may be leveraged to adapt an already trained mask estimator to unseen conditions. With minimal noisy adaptation data and no aligned clean utterances, the weights of the IRM estimator can be updated by using the joint framework. Note that the necessary senone labels for training the joint system can be obtained by aligning the adaptation data using an unadapted ASR model–a commonly used strategy in ASR. In the event that the mask labels for training the IRM estimator can be obtained or generated from data, the loss function can be easily modified to weight the contribution of the acoustic model loss and the mask estimator loss during adaptation. With limited adaptation data, model adaptation with additional regularization as has been proposed for speaker adaptation [22] will also be an interesting approach to consider.

An interesting future study will be the incorporation of discriminative training [16], [14] into our framework to allow higher-level sequence structure to influence separation and acoustic modeling. Having language model level sequence information influence speech separation has always been a challenge. Systems like convolutional NMF [36] and structure-preserving models [43] only account for short term sequence structure. Discriminative modeling, on the other hand, improves performance by optimizing model parameters to maximize the posterior probability of the correct word sequence. With joint adaptive training, it becomes fairly straightforward to use such well studied sequence training strategies to improve performance of speech separation.

The senone labels for training DNN-AMs were obtained by aligning the clean training data. When only the noisy data is available, such high quality labels cannot be directly obtained from the training set. An alternative is the iterative procedure outlined in [6]: First, train the best GMM-HMM system from the data, and use it to generate senone labels and train a DNN-AM. Next, use the trained DNN-AM to re-align the training data and update senone labeling. The DNN-AM can now be retrained with the revised alignments. Training can iterate between the last two steps multiple times, as long as improvements in performance are observed. Using such a strategy, we were able to train a DNN-AM that closely matched the performance obtained by the system presented here using the scripts provided with the KALDI ASR toolkit [33]. Compared to the baseline error rate of 20.2% as shown in Table VIII, the system obtains an average WER of 21.4 percent. With off-the-shelf sequence discriminative training [33], [14], the performance further improves to 19.5 percent. Note that a similar strategy was proposed in [3].

To conclude, we have proposed novel ways for improving the state-of-the-art in noise robust ASR using time-frequency masking. The results show that by using speech separation to provide smooth estimates of speech and noise to a DNN-AM, substantial improvements in performance can be obtained. Moreover, appending the features that are commonly used for separation and acoustic modeling to form the input of a DNN-AM also helps improve performance. Finally, a joint adaptive training framework is proposed for DNN-AMs. DNN-JAT unifies separation and acoustic modeling and consistently improves performance over the corresponding independently trained models.

ACKNOWLEDGMENT

We would like to thank Shinji Watanabe and Francesco Nesta for providing KALDI training and evaluation scripts.

REFERENCES

[1] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 2013, pp. 7942–7946.

[2] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. 4th Int. Conf. Spoken Lang.*, 1996, vol. 2, pp. 1137–1140.

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

[4] C.-P. Chen and J. A. Bilmes, "MVA processing of speech features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 257–270, Jan. 2007.

[5] J. Chen, Y. Wang, and D. L. Wang, "A feature study for classification-based speech separation at very low signal-to-noise ratio," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 2014, pp. 7089–7093.

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[7] 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.

[8] 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, 2010.

[9] J. T. 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.

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

[11] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. W. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[13] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 2014, pp. 1581–1585.

[14] L. B. K. Vesely, A. Ghoshal, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013.

[15] O. Kalinli, M. L. Seltzer, and A. Acero, "Noise adaptive training using a vector Taylor series approach for noise robust automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 2009, pp. 3825–3828.

[16] 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.

[17] U. Kjems, J. Boldt, M. Pedersen, T. Lunner, and D. Wang, "Role of mask pattern in intelligibility of ideal binary-masked noisy speech," *J. Acoust. Soc. Amer.*, vol. 126, no. 3, pp. 1415–1426, 2009.

[18] 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, pp. 1593–1602, 1994.

[19] B. Li and K. C. Sim, "Improving robustness of deep neural networks via spectral masking for automatic speech recognition," in *Proc. IEEE Workshop Autom. Speech Recogn. Understand.*, 2013, pp. 279–284.

[20] 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.

[21] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," *Comput., Speech, Lang.*, vol. 23, pp. 389–405, 2009.

[22] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 2013, pp. 7947–7951.

[23] H. Liao and M. J. F. Gales, "Adaptive training with joint uncertainty decoding for robust recognition of noisy data," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 2007, vol. 4, pp. 389–392.

[24] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Proc. Interspeech*, 2012.

[25] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[27] 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.

[28] A. Narayanan and D. L. Wang, "The role of binary mask patterns in automatic speech recogniton in background noise," *J. Acoust. Soc. Amer.*, vol. 133, pp. 3083–3093, 2013.

[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. 2523–2527.

[31] N. Parihar and J. Picone, "Analysis of the Aurora large vocabulary evaluations," in *Proc. Eur. Conf. Speech Commun. Technol.*, 2003, pp. 337–340.

[32] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Univ. of Cambridge, Cambridge, U.K., 2004.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recogn. Understand.*, 2011.

[34] T. N. Sainath, B. Kingsbury, H. Soltau, and B. Ramabhadran, "Optimization techniques to improve training speed of deep neural networks for large speech tasks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 11, pp. 2267–2276, Nov. 2013.

[35] M. L. Seltzer, D. Yu, and Y.-Q. 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.

[36] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[37] S. Srinivasan, N. Roman, and D. L. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Commun.*, vol. 48, pp. 1486–1501, 2006.

[38] Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey, "Discriminative methods for noise robust speech recognition: A CHiME challenge benchmark," in *Proc. 2nd CHiME Workshop Mach, Listen. Multisource Environ.*, , 2013, pp. 19–24.

[39] R. C. Van Dalen and M. J. F. Gales, "Extended VTS for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 733–743, May 2011.

[40] E. Vincent, J. Barker, S. Watanabe, J. LeRoux, F. Nesta, and M. Matassoni, in *Proc. 2nd CHiME Speech Separat. Recogn. Challenge*, 2012 [Online]. Available: http://spandh.dcs.shef.ac.uk/chime_challenge/chime2_task2.html

[41] 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.

[42] 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.

[43] Y. Wang and D. L. Wang, "A structure-preserving training target for supervised speech separation," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 2014, pp. 6148–6152.

[44] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, "The Munich feature enhancement approach to the 2nd CHiME challenge using BLSTM recurrent neural networks," in *Proc. 2nd CHiME Workshop Mach. Listening Multisource Environ.*, 2013, pp. 86–90.

[45] Y. Xu, J. Du, L. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

[46] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and B. Woodland, *The HTK Book*. Cambridge, U.K.: Cambridge Univ. Press, 2002 [Online]. Available: http://htk.eng.cam.ac.uk

[47] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks - studies on speech recognition tasks," in *Proc. Int. Conf. Learn. Represent.*, 2013.

Arun Narayanan (S'11) received the B.Tech. degree in computer science from the University of Kerala, Trivandrum, India, in 2005, and the M.S. and the Ph.D. degrees in computer science from the Ohio State University, Columbus, USA, in 2012 and 2014, respectively. His research interests include robust automatic speech recognition, speech separation, and machine learning. Since 2014, he has been with Google, Inc.

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