A Spectral Masking Approach to Noise-Robust Speech Recognition Using Deep Neural Networks

Bo Li, *Student Member, IEEE*, and Khe Chai Sim, *Member, IEEE*

***Abstract*—Improving the noise robustness of automatic speech recognition systems has been a challenging task for many years. Recently, it was found that Deep Neural Networks (DNNs) yield large performance gains over conventional GMM-HMM systems, when used in both hybrid and tandem systems. However, they are still far from the level of human expectations especially under adverse environments. Motivated by the separation-prior-to-recognition process of the human auditory system, we propose a robust spectral masking system where power spectral domain masks are predicted using a DNN trained on the same filter-bank features used for acoustic modeling. To further improve performance, Linear Input Network (LIN) adaptation is applied to both the mask estimator and the acoustic model DNNs. Since the estimation of LINs for the mask estimator requires stereo data, which is not available during testing, we proposed using the LINs estimated for the acoustic model DNNs to adapt the mask estimators. Furthermore, we used the same set of weights obtained from pre-training for the input layers of both the mask estimator and the acoustic model DNNs to ensure a better consistency for sharing LINs. Experimental results on benchmark Aurora2 and Aurora4 tasks demonstrated the effectiveness of our system, which yielded Word Error Rates (WERs) of 4.6% and 11.8% respectively. Furthermore, the simple averaging of posteriors from systems with and without spectral masking can further reduce the WERs to 4.3% on Aurora2 and 11.4% on Aurora4.**

***Index Terms*—Deep neural network, noise robustness, spectral masking.**

I. Introduction

DEEP neural networks (DNNs) have been adopted in many Automatic Speech Recognition (ASR) systems. Large performance improvements have been reported compared to systems that use Gaussian Mixture Models (GMMs) to represent the state emission probability distributions [1]. Basic DNN-based ASR systems, when trained with large amounts of data, have been found to yield superior performance over advanced GMM-based systems that employ a combination of different optimization techniques [2]. For noisy speech recognition, DNNs have also obtained comparable performance to the best GMM system with various noise reduction, feature enhancement and model-based compensation methods [3]. However, DNNs are still far from reaching humans' expectations and few methods have been developed to further improve DNNs' noise robustness [3]–[6].

To a certain extent, DNNs may be capable of learning some noise-dependent feature normalization effects implicitly through multiple layers of non-linear transformations. However, it will still be useful to explore explicit noise robustness techniques for DNNs. In general, feature compensation methods can easily be applied to DNNs as they are independent of the recognition models [3], [5]. However, these methods have only been found to yield performance gains for clean-trained DNNs due to the huge difference between the clean training data and the noisy testing data. With multi-style data, slight degradations have been observed when enhanced spectral features, which have been found to work well for GMM systems, are used to train DNNs [5]. This may be attributed to the imperfect enhancement process, which can potentially discard useful speech information and bring in unwanted distortions. Similarly, the Vector Taylor Series (VTS)-based feature compensation has also failed to yield gains for acoustic model DNNs [6].

Techniques specific to DNNs were hence developed. In [7], a Deep Recurrent Denoising Autoencoder (DRADE) was trained to reconstruct clean features from noisy ones. It makes no assumptions about how noise affects speech, nor the existence of distinct noise environments. It depends on the training data to provide a reasonable sample of the noise environment. In [8], a Factorial Hidden Restricted Boltzmann Machine (FHRBM) was proposed to explicitly model the noise distribution and how noise affects speech. However, due to the unobserved noise parameters, inference is intractable as the computational complexity scales exponentially with the number of hidden units. Our previous work [6] treated global Mean and Variance Normalization (MVN) as a Gaussian-based normalization front-end for DNNs and applied VTS to transform it towards the target testing scenario. But the oversimplification of the single Gaussian-based compensation limits its effectiveness. It yielded moderate performance gains only when adaptive training was used. In [3], the authors concatenated acoustic features with noise parameters of the corresponding utterance to train a "noise aware" DNN. However, this "noise aware" DNN yielded only a small gain. By further fine-tuning this DNN using dropout techniques, clear improvements were obtained [3].

When searching for methods that can lead to further performance gains, insights from the human speech perception process may be helpful. The human auditory system is capable of efficiently identifying and separating speech and noise prior to understanding [9]. Therefore, in this paper, we

Manuscript received August 01, 2013; revised December 02, 2013; accepted June 02, 2014. Date of publication June 05, 2014; date of current version June 25, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shinji Watanabe.

The authors are with the School of Computing, National University of Singapore, 117417 Singapore (e-mail: li-bo@outlook.com; simkc@comp.nus.edu.sg).

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.2014.2329237

investigate this "separation-prior-to-recognition" process via spectral masking for noise-robust speech recognition. Firstly, a DNN-based Mask Estimator (ME) was developed. Estimated masks were used to transform the noisy speech power spectrum into noise-invariant representations. Due to the use of DNNs for mask estimation, MEs are sensitive to training and testing mismatches. We then proposed to adopt a Linear Input Network (LIN) adaptation technique into our system. A batch mode adaptation was used and one LIN transform was estimated for each test set in an iterative manner. The estimated LINs are effective in reducing training and testing mismatches for Acoustic Models (AMs). But for MEs the LIN estimation requires stereo data, which is not available in practice. To solve this problem, we used the pre-trained Restricted Boltzmann Machine (RBM) weights rather than the fine-tuned ones for the DNN's input layer. This modified DNN is referred to as the RBM-DNN. A LIN transform can then be estimated for the RBM front-end in an unsupervised manner using Contrastive Divergence [10]. To further improve robustness, we also replaced the AM DNN with an RBM-DNN. Most importantly, by sharing the input RBM layer between the AM and the ME, the ME LIN transform can be learned by back-propagating the AM prediction errors.

The rest of the paper is organized as follows. The proposed spectral masking system is first described in Section II and the adaptation using a Linear Input Network to address the mismatch problem for both the AM and ME is discussed in Section III. Experimental results on Aurora2 and Aurora4 are presented in Section IV and we conclude the paper in Section V.

II. Spectral Masking

One of the important properties of auditory nerve responses in human speech perception is that they respond preferentially to certain frequencies [11]. To replicate this phenomenon of masking in human auditory perception, source segregation in computational auditory scene analysis can be achieved by computing a mask to weight the Time-Frequency (T-F) representations, such as the spectrograms of acoustic signals. This mask applies a weight to each T-F unit, such that spectral-temporal regions that are dominated by speech are emphasized, and regions that are dominated by other sources, such as noise, are suppressed. Values of the mask are either binary or real-valued; in the later case, the mask value can be interpreted as the ratio of the target energy to the mixed energy or the probability that specific T-F unit "belongs" to the target speech. A time-frequency weight of this kind was first employed in the binaural source separation algorithm described in [12], and has subsequently been adopted by other researchers [13], [14]. Recently, these methods have also seen many applications in robust ASRs [15], [16].

With stereo data, Ideal Binary Masks (IBMs) [17] have been shown to substantially improve the intelligibility of speech with background noise [18]. IBMs are computed in the power spectrum domain using:

$$m_{t,c}^{(\mathrm{IBM})} = \begin{cases} 1 & \text{if } r_{t,c}^{(\mathrm{SNR})} > LC \\ 0 & \text{otherwise,} \end{cases} \text{ and } r_{t,c}^{(\mathrm{SNR})} = 10\log_{10}\frac{x_{t,c}}{n_{t,c}}, \tag{1}$$

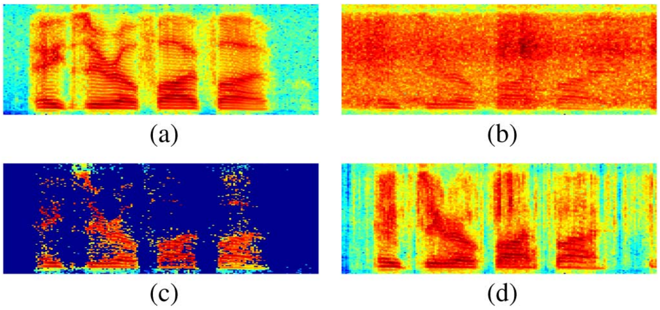


Fig. 1. Spectrogram comparisons for the same speech "8055" under different conditions: (a) clean; (b) with train noise at 0 dB SNR; (c) noisy spectrogram (b) with IBMs; (d) noisy spectrogram (b) with DNN-estimated masks.

where $r_{t,c}^{(\mathrm{SNR})}$ represents the local Signal-to-Noise Ratio (SNR) at the time frame t and the frequency channel c. $x_{t,c}$ and $n_{t,c}$ are the corresponding speech and noise energies. LC is a local SNR criterion [18]. IBMs are used in a direct spectral masking manner to remove noise-dominated T-F units [16], [19]. IBMs cannot be obtained in practical situations for spectral masking since they are computed using stereo data. Therefore, various classification-based algorithms for IBM predictions have been developed [20]–[23]. With the fast adoption of DNNs in various machine learning tasks, the original support vector machines used for mask estimation were replaced by DNNs [23], [24]. In those work, an ensemble of different features were used as DNNs' inputs and the mask estimation was performed in two stages. Firstly, a total number of 27 DNNs were trained using a single-frame input and 1,024 units per hidden layer. In the second stage, a shallow neural network was estimated, to give the final mask prediction by combining multiple frames of output predictions from the first stage DNNs. After masking, another reconstruction DNN was used to convert the masked partial spectral features to clean ones, which were then used as inputs for the final acoustic model DNN.

In this study, we propose to use a single DNN for mask estimation. Our spectral masking system (Fig. 2) is comprised of two DNNs: the Mask Estimation (ME) DNN and the Acoustic Model (AM) DNN. Both DNNs are trained with the same log Mel Filter-Bank (FBank) features so that they can share the same RBM pre-training step. After that, the two DNNs will be fine-tuned with different learning objectives.

The Mask Estimation (ME) DNN is trained using IBM vectors, $\boldsymbol{m}_t$, as supervision labels. The c-th component of $\boldsymbol{m}_t$, *i.e.* $m_{t,c}$, represents the probability, $P(m_{t,c}^{(\mathrm{IBM})} = 1|\boldsymbol{x}_t)$, that the c-th power spectral component of the observation $\boldsymbol{o}_t$ is dominated by speech. The DNN input at time t consists of a window of $2w+1$ adjacent frames, *i.e.* $\boldsymbol{x}_t = [\boldsymbol{o}_{t-w}^T \quad \cdots \quad \boldsymbol{o}_t^T \quad \cdots \quad \boldsymbol{o}_{t+w}^T]^T$. The computation performed by an L-layer ME DNN is as follows:

$$\boldsymbol{h}_{0,t}^{\mathrm{ME}} = \boldsymbol{x}_t, \tag{2}$$

$$\boldsymbol{h}_{l,t}^{\mathrm{ME}} = \sigma(\boldsymbol{W}_l^{\mathrm{ME}}\boldsymbol{h}_{l-1,t}^{\mathrm{ME}} + \boldsymbol{b}_l^{\mathrm{ME}}), \quad \text{for } 1 \leq l < L. \tag{3}$$

$$\boldsymbol{m}_t = \sigma(\boldsymbol{W}_L^{\mathrm{ME}}\boldsymbol{h}_{L-1,t}^{\mathrm{ME}} + \boldsymbol{b}_L^{\mathrm{ME}}), \tag{4}$$

where $\boldsymbol{W}_l^{\mathrm{ME}}$ and $\boldsymbol{b}_l^{\mathrm{ME}}$ are the model parameters for the l-th layer in the ME DNN; $\boldsymbol{h}_{l,t}^{\mathrm{ME}}$ is the input to the $(l+1)$-th layer. $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function. In training,

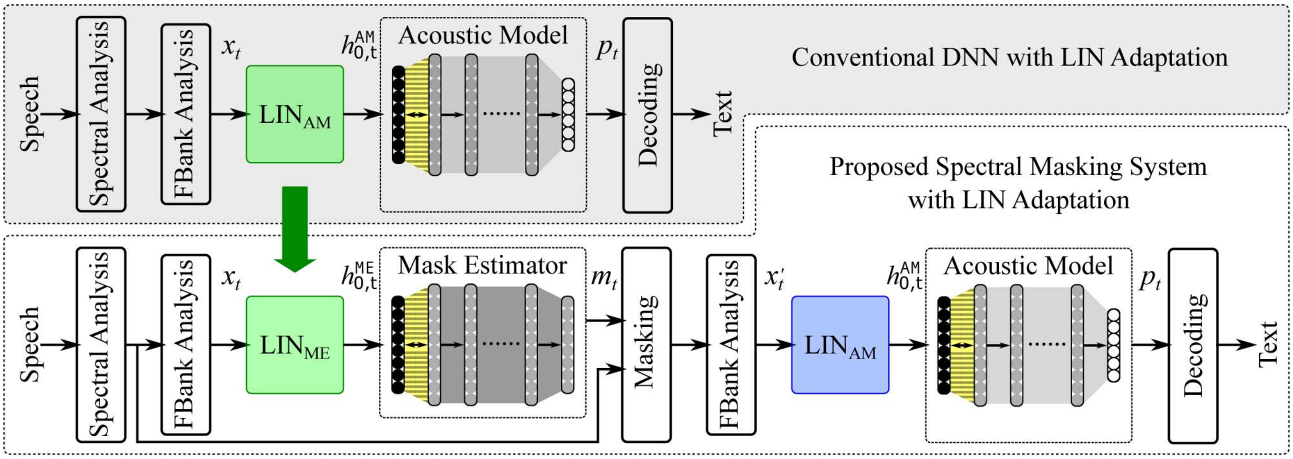


Fig. 2. System architecture comparisons between the conventional DNN-based acoustic model (the lightly shaded upper part) and the proposed spectral masking system (the unshaded lower part). We denote the DNN's input nodes with black circles and the softmax output nodes with white circles. The sigmoid hidden units and the mask estimator's sigmoid output units are denoted with gray circles. The Linear Input Network (LIN) adaptation transforms for the mask estimator and the acoustic model are represented as $\mathrm{LIN}_{\mathrm{ME}}$ and $\mathrm{LIN}_{\mathrm{AM}}$ respectively.

the ME DNN model parameters $\theta^{\mathrm{ME}} = \{\boldsymbol{W}_l^{\mathrm{ME}}, \boldsymbol{b}_l^{\mathrm{ME}} | 1 \le l \le L\}$ were firstly initialized using the pre-trained RBMs and then fine-tuned using the standard Error Back-Propagation (EBP) algorithm [25] to minimize the Mean Square Error (MSE) over the set of training samples $\mathcal{O} = \{\boldsymbol{o}_1, \cdots, \boldsymbol{o}_T\}$:

$$\theta^{\mathrm{ME}} = \arg\min_{\theta^{\mathrm{ME}'}} \frac{1}{2} \sum_{t=1}^{T} \sum_{c} (m_{t,c} - m_{t,c}^{(\mathrm{IBM})})^2. \tag{5}$$

To avoid potential errors brought by binarizing the estimated masks, we directly apply those real-valued masks to noisy speech through a component-wise multiplication. This is usually referred to as soft-masking. An example of the masked power spectrogram obtained using our mask estimator DNN is illustrated in Fig. 1(d). From these masked power spectra, a new set of FBank features, $\boldsymbol{o'}_t$, can be extracted accordingly and are more invariant to noise.

Using these noise-invariant features, $\boldsymbol{x'}_t = [\boldsymbol{o'}_{t-w} \ \cdots \ \boldsymbol{o'}_t \ \cdots \ \boldsymbol{o'}_{t+w}]$, a canonical Acoustic Model (AM) DNN can be trained to yield a more robust phoneme posterior prediction $\boldsymbol{p}_t$, which is computed by:

$$\boldsymbol{h}_{0,t}^{\mathrm{AM}} = \boldsymbol{x'}_t, \tag{6}$$

$$\boldsymbol{h}_{l,t}^{\mathrm{AM}} = \sigma(\boldsymbol{W}_l^{\mathrm{AM}} \boldsymbol{h}_{l-1,t}^{\mathrm{AM}} + \boldsymbol{b}_l^{\mathrm{AM}}), \quad \text{for } 1 \le l < L \tag{7}$$

$$\boldsymbol{p}_t = \delta(\boldsymbol{W}_L^{\mathrm{AM}} \boldsymbol{h}_{L-1,t}^{\mathrm{AM}} + \boldsymbol{b}_L^{\mathrm{AM}}). \tag{8}$$

Different from ME DNNs, the softmax function, $\delta(x) = \exp(x)/\sum_{\bar{x}} \exp(\bar{x})$, is used in the output layer. The s-th component $p_{t,s}$ is an estimate of the posterior probability $P(s|\boldsymbol{x'}_t)$ for the Hidden Markov Model (HMM) state s. Using Bayes' rule, the observation probability of the input $\boldsymbol{x'}_t$ given the state s can be computed by:

$$P(\boldsymbol{x'}_t|s) = P(s|\boldsymbol{x'}_t)P(\boldsymbol{x'}_t)/P(s), \tag{9}$$

$$\propto P(s|\boldsymbol{x'}_t)/P(s), \tag{10}$$

where $P(s)$ is the prior probability of state s calculated from the training data. $P(\boldsymbol{x'}_t)$ is the observation probability that is independent of the state sequence and can be ignored. The likelihood computation $P(\boldsymbol{x'}_t|s)$ is hence commonly simplified to $P(s|\boldsymbol{x'}_t)/P(s)$, for generating the final recognition hypotheses in the HMM framework [1], [26]–[28]. The model parameters of the AM DNN, $\theta^{\mathrm{AM}} = \{\boldsymbol{W}_l^{\mathrm{AM}}, \boldsymbol{b}_l^{\mathrm{AM}} | 1 \le l \le L\}$, are initialized with the same set of pre-trained RBMs and then fine-tuned using the EBP algorithm. However, the cross entropy objective is optimized here:

$$\theta^{\mathrm{AM}} = \arg\min_{\theta^{AM'}} \left(-\sum_{t=1}^{T} \log P(s_t|\boldsymbol{x'}_t) \right), \tag{11}$$

where s_t is the reference state label for the observation at time t obtained by a forced-alignment of the speech signal with its corresponding transcription.

III. Adaptation using a Linear Input Network

In noisy speech recognition, large mismatches between training and testing data are usually unavoidable due to the inherent variability of noise. Performance degradations are expected when the system is used in unknown noise conditions. This has also been observed for DNN-based ASR systems in [4]–[6]. In this research, a Linear Input Network (LIN) adaptation technique [29]–[33] is used to address the mismatch issue. Firstly, the mismatch problem affects the DNN-based ME. Erroneous mask estimations dramatically degrade the system performance, as observed in [20]–[23]. However, it is impossible to directly estimate LINs for ME DNNs during testing because the required IBM supervisions are not available in practice. In this study, we propose two approaches to solve this problem: namely RBM-based LIN adaptation and LIN sharing. Secondly, the mismatch also happens in the masked feature domain. Although masking aims to remove noise such that features are made more similar to the clean speech, it usually cannot achieve this objective because of mask estimation errors (Fig. 1(d) *vs.* Fig. 1(a)). Moreover, ideally masked features (Fig. 1(c)) are different from clean speech (Fig. 1(a)). Retraining the AM with masked features is crucial. However, the different mask estimation accuracies of the ME DNN on the training and testing data may also cause potential mismatches among masked features. Adopting additional adaptation transforms for the AM DNN is necessary and beneficial. Our final spectral masking system with LIN adaptation is depicted in Fig. 2. For comparisons, a conventional DNN-HMM system with LIN adaptation is also illustrated in Fig. 2. In this section we first review LIN adaptation for the AM DNN and then present the proposed ME DNN adaptation.

A. Acoustic Model Adaptation

Linear Input Network (LIN) adaptation [29]–[33] represents training and testing feature mismatches with a weight matrix $\boldsymbol{T}^{\mathrm{LIN}}$ and a bias vector $\boldsymbol{b}^{\mathrm{LIN}}$. Instead of directly forwarding the observation to DNNs, the LIN-transformed one is used:

$$\boldsymbol{h}_{0,t} = g_{\mathrm{LIN}}(\boldsymbol{x}_t) = \boldsymbol{T}^{\mathrm{LIN}}\boldsymbol{x}_t + \boldsymbol{b}^{\mathrm{LIN}}. \tag{12}$$

It effectively adds an additional input layer to the original model with a linear activation function, which is why it is referred to as the Linear Input Network transform. The estimation of LIN transforms is based on EBP and hence follows exactly the same procedure as the AM DNN training:

$$\boldsymbol{T}^{\mathrm{LIN}}(\tau+1) = \boldsymbol{T}^{\mathrm{LIN}}(\tau) + \eta * \Delta\boldsymbol{T}^{\mathrm{LIN}}(\tau), \tag{13}$$

$$\boldsymbol{b}^{\mathrm{LIN}}(\tau+1) = \boldsymbol{b}^{\mathrm{LIN}}(\tau) + \eta * \Delta\boldsymbol{b}^{\mathrm{LIN}}(\tau) \tag{14}$$

and

$$\Delta\boldsymbol{T}^{\mathrm{LIN}}(\tau) = \frac{\partial\theta^{\mathrm{AM}}}{\partial\boldsymbol{T}^{\mathrm{LIN}}(\tau)}, \quad \Delta\boldsymbol{b}^{\mathrm{LIN}}(\tau) = \frac{\partial\theta^{\mathrm{AM}}}{\partial\boldsymbol{b}^{\mathrm{LIN}}(\tau)}, \tag{15}$$

where $\tau \in 0,\ldots,N$ is the update iteration index and η is the learning rate. Commonly, we start with $\boldsymbol{T}^{\mathrm{LIN}}(0) = \boldsymbol{I}$ and $\boldsymbol{b}^{\mathrm{LIN}}(0) = \boldsymbol{0}$. Supervision labels are required for the gradient computation. For unsupervised AM adaptation, recognition hypotheses are used instead. One potential problem is that the hypothesis errors may impede gains from adaptation.

B. Unsupervised Mask Estimator Adaptation

Unlike in AM adaptation, no proper supervision labels could be used for mask estimator adaptation. To solve this problem, we propose using the pre-trained RBM weights instead of those fine-tuned weight parameters for the DNN's first layer. These pre-trained parameters are estimated to minimize the RBM energy function, which is equivalent to maximize the data likelihood, using Contrastive Divergence (CD) [10]. No supervision labels are required in this process. To distinguish from the standard DNN, we will refer to this modified DNN as the RBM-DNN. With this modification, we do not need any supervision labels to adapt the input layer parameters using LIN transforms. The RBM energy function with LIN is:

$$\begin{aligned} E(g_{\mathrm{LIN}}(\boldsymbol{x}_t), \boldsymbol{h}_{1,t}) = -\boldsymbol{h}_{1,t}^T\boldsymbol{W}_1 g_{\mathrm{LIN}}(\boldsymbol{x}_t) - \boldsymbol{b}_1^T\boldsymbol{h}_{1,t} \\ - \boldsymbol{a}_1^T g_{\mathrm{LIN}}(\boldsymbol{x}_t), \end{aligned} \tag{16}$$

where $\boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{a}_1$ are the RBM parameters and the subscript 1 implies it is the first layer of the RBM-DNN. $\boldsymbol{a}_1$ is the input bias. The update of the LIN parameters by optimizing the testing data log likelihood using CD is:

$$\Delta\boldsymbol{T}^{\mathrm{LIN}}(\tau) = \langle \boldsymbol{x}_t\boldsymbol{h}_{1,t}^T\boldsymbol{W}_1\rangle_{data} - \langle \boldsymbol{x}_t\boldsymbol{h}_{1,t}^T\boldsymbol{W}_1\rangle_{model}, \tag{17}$$

$$\Delta\boldsymbol{b}^{\mathrm{LIN}}(\tau) = \langle \boldsymbol{W}_1^T\boldsymbol{h}_{1,t}\rangle_{data} - \langle \boldsymbol{W}_1^T\boldsymbol{h}_{1,t}\rangle_{model}, \tag{18}$$

where the operator "$\langle\cdot\rangle$" computes the expectation value with respect to either the "data" or the "model".

Inspired by the success of reusing feature-space Maximum Likelihood Linear Regression (fMLLR) transforms from a GMM-HMM system to both shallow [33] and deep [34] neural network acoustic models, we propose adapting the ME by sharing transforms. Unlike systems in [23], [24], the use of the same input between our AM and ME allows the exchange of feature transforms. We hence investigate reusing adaptation transforms estimated for the AM DNN to the ME DNN. Furthermore, to ensure consistency in sharing transforms among different models, we adopt the RBM-DNN for both acoustic modeling and mask estimation, and further constrain them to use the same set of input layer parameters, which are obtained from the RBM pre-trained on the same input features using Contrastive Divergence. These parameters are estimated to generate generic hidden representations that are capable of capturing the input data distribution and are independent of both tasks. Empirically, we show that the LINs estimated for the AM RBM-DNN perform much better for MEs than those estimated for the pure DNN-based AM.

C. Structure Constraints for LIN

The use of long-span acoustic features in DNNs is important to its superior performance; but it also causes a large increase in the number of adaptation parameters in the LIN transforms. For example, for a conventional 39-dimensional MFCC-based GMM system, the feature-space Maximum Likelihood Linear Regression (fMLLR) transform has around 1.5k ($39 * 39 + 39$) parameters. For an 11-frame input window DNN using the same features, the LIN transform will have 184.5k ($11 * 39 * 11 * 39 + 11 * 39$) parameters. With the same limited amount of enrollment data, an estimation of the maximum likelihood based fMLLR is undoubtedly more reliable than that of the discriminative LIN. For a window of $2w+1$ frames' input, *i.e.* $\boldsymbol{x}_t = [\boldsymbol{o}_{t-w}^T \;\cdots\; \boldsymbol{o}_t^T \;\cdots\; \boldsymbol{o}_{t+w}^T]^T$, the LIN transform also has a similar block structure:

$$\boldsymbol{T}^{\mathrm{LIN}} = \begin{bmatrix} \boldsymbol{T}_{-w,-w} & \cdots & \boldsymbol{T}_{-w,0} & \cdots & \boldsymbol{T}_{-w,w} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{T}_{0,-w} & \cdots & \boldsymbol{T}_{0,0} & \cdots & \boldsymbol{T}_{0,w} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{T}_{w,-w} & \cdots & \boldsymbol{T}_{w,0} & \cdots & \boldsymbol{T}_{w,w} \end{bmatrix}, \tag{19}$$

where each $\boldsymbol{T}_{i,j}$ is a transform similar to an fMLLR transform. $\boldsymbol{T}_{i,i}$ models the intra-frame correlation and $\boldsymbol{T}_{i,j}, i \neq j$ models the inter-frame correlation between frame i and j. To reduce the number of parameters in LIN, we can first remove all the inter-frame correlations by constraining $\boldsymbol{T}_{i,j} = \boldsymbol{0}$ for all $i \neq j$. This kind of LIN is referred to as the block diagonal LIN - LIN(blk). Furthermore, we can constrain all the intra-frame correlations to use the same transform, $\boldsymbol{T}_{i,i} = \boldsymbol{A}$. This is referred to as the shared block diagonal LIN, *i.e.* the LIN(shd). It has a comparable number of parameters to an fMLLR transform. In [24], a diagonal LIN has been adopted, which will be referred to as the LIN(dig). With this strong constraint, LIN has the same effect as MVN and is only used in utterance-based adaptations.

IV. Experiments

In this section, we justify the effectiveness of the proposed adaptive spectral masking based noise-robust ASR system. Experiments are carried out on both Aurora2 [35] and Aurora4 [36] datasets. For the computation of IBMs (*cf.* equation (1)), the stereo training data including both the clean and multi-style data has to be used to derive the noise energy by subtracting the clean speech energy from the noisy speech energy. Other than that, the AM training uses only the multi-style training data.

A. Aurora2

The multi-style training data of Aurora2 comprises 8,440 utterances with 4 different noise scenarios (train, babble, car and exhibition hall) at 4 different SNRs (20 dB, 15 dB, 10 dB, 5 dB), as well as in clean conditions. All the test sets are grouped into 3 broad sets, A, B and C. They are all used for evaluation. Set A has the same noise as the multi-style training data and set B has four new noise types (restaurant, street, airport and train station). For set C, there are only two noise scenarios (train and street) but with additional channel distortions. For all the test sets, 5 different SNRs are used for evaluation, with one additional 0 dB SNR compared to the training set.

A standard complex back-end GMM-HMM system was built using the per-utterance Cepstral Mean and Variance Normalized (CMVN) MFCC features by maximizing the training data likelihood. The 16-state word-based HMM and the 5-state silence model are adopted, leading to a total number of 181 HMM states. This GMM-HMM system is used to generate the per-frame DNN training labels. For DNN systems, we used 24-dimensional FBank features together with their first- and second-order derivatives as inputs. A per-utterance MVN was also adopted for input feature normalization. A consecutive 11 frames of the acoustic features were concatenated as the input to the DNNs. No language model was used for this task and an equal probability digit-loop was adopted for decoding. The open source Kaldi toolkit [37] was used to train both the GMM-HMM systems and the DNN-HMM systems. A threshold of 0 dB was used as the LC parameter in equation (1) for the computation of IBMs on the training data. By applying IBMs on the testing data, the WER lower bounds obtained on set A, B and C were 1.1%, 1.0% and 1.2% respectively [38].

1) Finding the Optimal DNN Setup: For the DNN baseline setup, we pre-trained a stack of 8 RBMs and then initialized 8 different DNNs with a different number of hidden layers. 2,048 hidden units were used for each hidden layer. The output softmax layer was randomly initialized. Each RBM was trained using a momentum of 0.5 for 10 iterations and followed by a momentum of 0.9 for another 40 iterations. A constant 0.001 learning rate was used for the first layer Gaussian-Bernoulli RBM and all the other Bernoulli-Bernoulli RBMs used a learning rate of 0.1. An L2 weight penalty of 0.0002 was also used. During fine-tuning, the "newbob" learning schedule [39] was used. The learning rate was initially set to 0.015. After each iteration of training, the learning rate was halved if the frame accuracy improvement on a held-out cross-validation set was less than 0.5%. The whole fine-tuning process stopped when the learning rate fell below 0.0001. The best number of hidden layers for this task is 4, which yields the best average WER of 5.0%. We hence used this 4-hidden-layer DNN system as our AM DNN baseline.

TABLE I
Aurora2 WER (%) Performance of Different RBM-DNN Configurations ("gen" for Pre-Trained RBM Layers and "dis" for Discriminatively Fine-Tuned DNN Layers)

# of Hidden Layers			Test Set			Avg.
Total	gen	dis	A	B	C	
4	0	4	4.6	5.3	5.1	5.0
4	1	3	**4.5**	**5.1**	**5.0**	**4.9**
	2	2	4.9	5.3	5.2	5.1
	3	1	5.7	5.6	5.8	5.7
	4	0	7.4	6.8	7.6	7.2
5	0	5	4.5	5.5	5.3	5.0
	1	**4**	4.7	**5.1**	5.1	**4.9**
6	0	6	4.6	5.4	5.2	5.0
	2	**4**	4.7	5.2	5.1	5.0
7	0	7	4.5	5.5	5.2	5.0
	3	**4**	5.1	5.4	5.4	5.3

2) RBM-DNN vs. DNN: Although the RBM-DNN is proposed for adapting the ME, it is interesting to understand the effect of using an RBM input layer. We hence experimented with different combinations of generative ("gen") and discriminative ("dis") depths for the AM DNN. We use the term "generative" only to suggest that the layers are obtained from the unsupervised pre-training rather than the discriminative fine-tuning. Experimental results are tabulated in Table I with the first row as the baseline DNN system. The RBM-DNN with only 1 RBM performs the best among those with the same number of hidden layers. It has lower WERs on all the test sets compared to the standard DNN. With the same number of discriminative layers, adding one and only one RBM input layer is also the best. We hence take the RBM-DNN with 1 pre-trained RBM layer and 3 discriminatively tuned DNN layers as the new baseline, which is both faster (1 less layer for fine-tuning) and more robust than the standard DNN.

3) Spectral Masking: In the proposed spectral masking system, we adopted the same model structure, *i.e.* the RBM-DNN, for both mask estimation and acoustic modeling with a shared RBM front-end. The baseline RBM-DNN is denoted as system S1. Firstly the masked features were directly decoded with the baseline acoustic model RBM-DNN (*i.e.* system S2 in Table II). The mismatches between the noisy features and the masked partial features led to a WER performance degradation from 4.9% to 6.9%. After retraining the AM RBM-DNN with the masked training data, *i.e.* system S3 in Table II, the performance improved to a WER of 5.2%. However, it is still worse than system S1. From Fig. 3 with detailed WER changes from system S1 to S3, masking helps in reducing WERs on matched conditions (such as subsets A1, A3 and A4) and degrades for all the unknown conditions (such as subsets B1, B2, B3 and B4). The only exceptional case is subset A2, which has speech-like babble noise. This kind of noise has distributions similar to the target speech and is hence much more difficult for machines to identify, and mask out, from the noisy speech mixtures. For most of the matched noise types, the masking system S3 has larger improvements on lower SNRs (such as 5 dB and 0 dB). For test set C, the performance is improved on subset C1 with matched additive

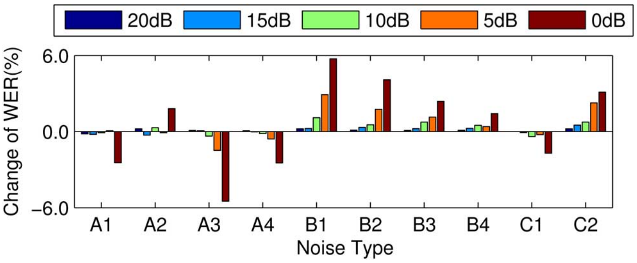


Fig. 3. WER changes from system S1 to S3.

TABLE II
AURORA2 WER (%) PERFORMANCE OF MASKED FEATURES

System	Masking		Test Set			Avg.
	Train	Test	A	B	C	
S1	×	×	4.5	**5.1**	**5.0**	**4.9**
S2	×	✓	5.2	8.6	6.9	6.9
S3	✓	✓	**3.9**	6.3	5.4	5.2

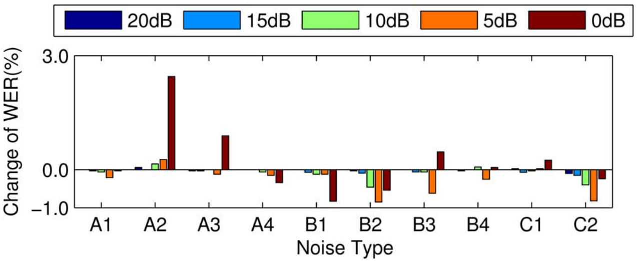


Fig. 4. WER changes from system S1 to S1 + LIN.

TABLE III
AURORA2 WER (%) PERFORMANCE OF AM ADAPTATION WITH LINs

System	LIN	Test Set			Avg.
		A	B	C	
S1	×	**4.5**	5.1	5.0	4.9
S1+LIN	✓	4.6	**5.0**	**4.9**	**4.8**

TABLE IV
AURORA2 PERFORMANCE OF ME ADAPTATION USING GENERATIVE LINs

Task	LIN	Test Set			Avg.
		A	B	C	
Mask Estimation MSE	×	5.49	6.95	5.97	6.17
	ora.	**5.08**	**5.98**	**5.38**	**5.50**
	est.	5.52	7.04	6.01	6.23
Speech Recognition WER(%)	×	3.9	6.3	5.4	5.2
	ora.	**2.8**	**4.1**	**3.6**	**3.5**
	est.	4.0	6.3	5.4	5.2

noise, and degraded on subset C2 with unknown additive noise, regardless of the channel distortions. This may suggest that our masking system is more sensitive to additive noise rather than channel distortions.

4) Acoustic Model Adaptation with LINs: To address the training and testing data mismatch problem, we investigated the effectiveness of LINs for AMs. An initial decoding was required to generate adaptation hypotheses. One LIN was estimated for each noise condition. For Aurora2, each test set contains 1,001 utterances from the same 104 speakers. It adds up to approximately half an hour of speech data. All the utterances in one test set share the same noise condition and SNR. The estimated LIN transforms are hence noise- and SNR-dependent and speaker-independent. The LINs were initialized to be identity matrices and iteratively updated using EBP. To avoid over-fitting, 10% of the adaptation data was held out as the validation set. The same "newbob" training strategy [39] was used for LIN estimations. The evaluation criterion used on this validation set was the frame prediction accuracy. Experimental results in Table III show that the LIN adaptation can slightly improve the baseline system S1 on both set B and set C; but a slight degradation on set A can be observed (from 4.5% to 4.6%). From Fig. 4, LINs could hardly give improvements on matched conditions, as the AM RBM-DNN had already captured those variations from the training data. They are also sensitive to hypothesis errors, as they degraded dramatically for speech at 0 dB in subsets A2 and A3. For mismatched noise types, LINs are effective in improving performance by minimizing training and testing feature mismatches.

5) Mask Estimator Adaptation with Generative LINs: To understand the effectiveness of adapting the ME, we first estimated LINs using oracle IBM supervisions of the test data. The results are tabulated in rows with "ora." under the "LIN" column of Table IV. The clear error reductions in all the test sets for both the mask estimation and the word recognition suggest the necessity of adapting the ME. In practice, one approach of adapting the ME without IBM supervision labels is to estimate LINs within the RBM learning framework as discussed in Section III-B. The results are tabulated in rows with "est." under the "LIN" column of Table IV. These estimated LINs increase mask estimation errors and have no significant effect on the recognition performance. One explanation is that the LINs were optimized for data reconstruction, rather than mask or phoneme predictions.

6) Mask Estimator Adaptation using LIN Sharing: Another approach of adapting the ME is to reuse LINs estimated for the AM. Similarly, we first justified the potential of LIN sharing. Oracle word transcriptions of the test data were used to estimate LINs for the AM. These LINs, referred to as the "ora." LINs, were then directly applied to the ME. Both the standard DNN and the proposed RBM-DNN were evaluated. Results in Table V show that the "ora." AM LINs are effective for our RBM-DNN based masking system by reducing the average WER of 5.2% to 4.3%, although they are not as good as the "ora." LINs directly estimated for the ME (*cf.* Table IV). In practice, we do not have oracle word transcriptions for the test data. Recognized erroneous hypotheses have to be used for the LIN estimation. LINs obtained in this way are referred to as the "est." LINs in Table V. Degradations due to supervision errors were observed for both systems. But for our RBM-DNN, the "est." LINs still improved both the mask estimation and speech recognition performance on set B and C over the unadapted system. The average WER of 5.2% obtained from the spectral masking system S3 with the unadapted ME was reduced to 4.9% by using the "est." LIN (the last row in Table V, which will be referred to as system S4). From Fig. 5 with detailed WER changes from S3 to S4, it can be seen that the degradation on set A mainly happens on speech with 0 dB SNR. It is probably due to the high error rate of the hypotheses generated for those 0 dB speech data. In general, the shared LINs do

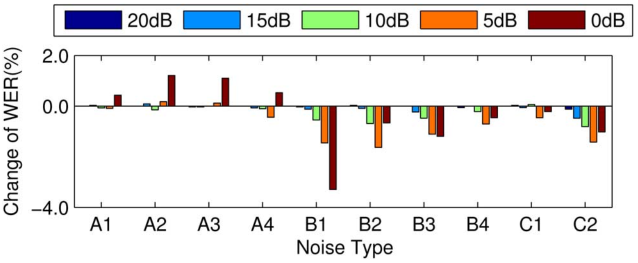


Fig. 5. WER changes from system S3 to S4.

TABLE V
AURORA2 PERFORMANCE OF ME ADAPTATION USING LIN SHARING

Task	Model	LIN	Test Set A	B	C	Avg.
Mask Estimation MSE	DNN	×	5.58	6.97	6.05	6.23
		ora.	5.84	7.21	6.14	6.45
		est.	5.67	6.95	6.02	6.25
	RBM-DNN	×	**5.49**	6.95	5.97	6.17
		ora.	5.58	6.91	**5.88**	6.17
		est.	5.58	**6.89**	**5.88**	**6.16**
Speech Recognition WER(%)	DNN	×	4.1	6.3	5.4	5.2
		ora.	4.5	6.6	5.4	5.5
		est.	4.3	6.0	5.3	5.2
	RBM-DNN	×	3.9	6.3	5.4	5.2
		ora.	**3.5**	**5.0**	**4.5**	**4.3**
		est.	4.1	5.7	5.0	4.9

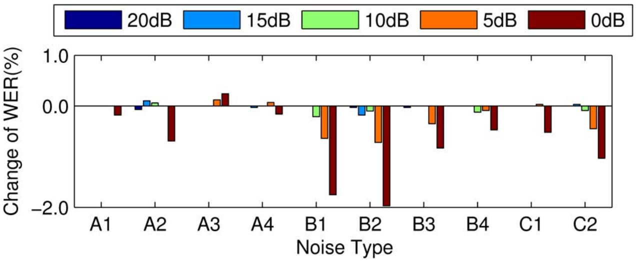


Fig. 6. WER changes from system S4 to S4 + LIN.

address the training and testing data mismatch problem. Our RBM-DNN is not only more robust than the standard DNN, but also more reliable in sharing transforms.

7) AM Adaptation with Spectral Masking: Next, we addressed the potential mismatches between the masked training and testing data, by adapting the AM of system S4 with another set of LINs. It will be referred to as system S4 + LIN. The results are listed in Table VI and the detailed WER reductions are illustrated in Fig. 6. The best average WER of 4.7% was achieved, which is also better than system S1 + LIN. Most of the gains come from lower SNRs. In Table VI, we also included the results of adapting only the AM in system S3, which is referred to as system S3 + LIN. Comparing system S1 + LIN (Table III), system S3 + LIN (Table VI) and system S4 + LIN (Table VI), the use of masks enables a more effective LIN adaptation.

8) Constraining LIN Transforms: Adaptation with LINs has improved system performance consistently. But errors in the supervision hypotheses for LIN estimations have always caused degradations on set A. To improve the adaptation robustness against supervision errors, we propose to reduce the number of parameters by constraining the LINs. This leads to the block diagonal LIN - LIN(blk), and the shared block diagonal LIN - LIN(shd). They were firstly evaluated in system S1 + LIN. Results in Table VII show that the constraints fail to render any improvements. Despite this, we nonetheless used those transforms for our ME. The results in Table VII for system S4 show that these constraints do improve our masking system. The gains may have come from the ME's high sensitivity to input feature mismatches. For the ME, each sigmoid output is independent of the others. In the case of the AM, due to the softmax function, shifts in the final prediction are probably normalized away. Moreover, by further adapting the AM in our masking system with constrained LINs, the best average WER of 4.6% was achieved (the system S4 + LIN in Table VII).

TABLE VI
AURORA2 WER (%) PERFORMANCE OF SPECTRAL MASKING WITH LIN ADAPTATION

System	LIN ME	AM	Test Set A	B	C	Avg.
S3	×	×	**3.9**	6.3	5.4	5.2
S3+LIN	×	✓	**3.9**	5.6	5.0	4.8
S4	✓	×	4.1	5.7	5.0	4.9
S4+LIN	✓	✓	4.1	**5.3**	**4.8**	**4.7**

TABLE VII
AURORA2 WER (%) PERFORMANCE COMPARISONS OF LINS WITH DIFFERENT STRUCTURE CONSTRAINTS

System	LIN ME	AM	Type	Test Set A	B	C	Avg.
S1+LIN	–	✓	full	4.6	**5.0**	4.9	4.8
			blk	4.8	5.1	4.9	4.9
			shd	4.9	5.1	4.9	5.0
S4	✓	×	full	4.1	5.7	5.0	4.9
			blk	**3.9**	5.5	4.9	4.7
			shd	**3.9**	5.6	4.9	4.8
S4+LIN	✓	✓	full	4.1	5.3	4.8	4.7
			blk	**3.9**	5.2	**4.7**	**4.6**
			shd	**3.9**	5.2	**4.7**	**4.6**

9) Posterior Interpolation: Comparing the WER breakdowns of the conventional system S1+LIN (blk), *i.e.* without masking, and the proposed masking system S4 + LIN (blk) in Table VII, performance complementariness is observable. System S1 + LIN (blk) performed the best on set B, while system S4 + LIN (blk) had the best performance on set A and C. In this experiment, we simply averaged the posteriors generated from these two systems and an average WER of 4.3% was achieved. No further gain could be obtained by tuning the posterior interpolation weight from 0.0 to 1.0 by 0.1. The detailed SNR-dependent WER breakdowns for system S1 + LIN (blk), system S4 + LIN (blk) and the posterior interpolation system indicated as "PostInter" are listed in Table VIII.

B. *Aurora4*

To justify the effectiveness of our proposed spectral masking system, we tested it on the Aurora4 dataset, as explained in this section. It is a medium vocabulary noisy speech recognition task. The multi-style training data consists of one half of the total 7,138 utterances recorded by the primary Sennheiser microphone and the other half recorded using one of the 18 different secondary microphones. Both halves include a combination of clean speech and speech corrupted by one of the six different noise types (street traffic, train station, car, babble, restau-

TABLE VIII
AURORA2 WER (%) PERFORMANCE BREAKDOWNS OF SYSTEM "S1 + LIN (BLK)", SYSTEM "S4 + LIN (BLK)" AND THE POSTERIOR INTERPOLATION SYSTEM "POSTINTER"

System	A (SNR (dB))					B (SNR (dB))					C (SNR (dB))					Avg.
	20	15	10	5	0	20	15	10	5	0	20	15	10	5	0	
S1+LIN(blk)	0.4	0.6	1.4	4.3	17.4	0.4	0.6	1.3	4.7	18.3	0.5	0.8	1.7	4.8	16.6	4.9
S4+LIN(blk)	0.4	0.6	1.2	**3.5**	**13.5**	0.5	0.8	1.5	5.3	17.9	0.6	0.8	1.6	4.9	15.6	4.6
PostInter	**0.3**	**0.5**	**1.1**	3.6	15.0	**0.3**	**0.5**	**1.0**	**4.3**	**16.8**	**0.4**	**0.7**	**1.4**	**4.2**	**14.8**	**4.3**

rant, airport) at 10-20 dB SNR range. The clean training data, which has the same number of utterances, was only used to compute the training IBMs with a threshold of $LC = -6$ dB. There are 14 test sets with the same six types of noise but at a 5-15 dB SNR range. They are further grouped into 4 broad sets for easy comparisons: clean, noisy, clean with channel distortions, noisy with channel distortions, which will be referred to as A, B, C and D, respectively.

1) Baseline System: A context-dependent GMM-HMM system with 3,356 senones was trained using maximum likelihood estimation on the per-utterance CMVN normalized 39-dimensional MFCC features. It was also used to create senone labels for training the hybrid systems. Decoding was performed with the standard WSJ0 bigram language model. DNNs were trained using 24-dimensional FBank features together with the first and second order derivatives. An utterance level MVN was adopted. A context window of 11 adjacent frames were used as the DNN inputs. The RBM training configuration was the same as the one used for Aurora2 except that we trained 200 iterations for the Gaussian-Bernoulli RBM and 100 iterations for all the Bernoulli-Bernoulli RBMs with a momentum value of 0.9. The same fine-tuning process used previously in Aurora2 was adopted here. The DNN with six 2,048-dimensional hidden layers yielded the best WER of 13.8%. With two additional iterations of re-alignment and re-training, we can reduce the average WER to 13.4% and no further improvement could be obtained by doing more iterations. By applying IBMs on the testing data, the WER lower bounds obtained for set A, B, C and D were 4.9%, 6.5%, 8.0% and 12.2% respectively. It gave a lower bound of 8.9% for the averaged WER on Aurora4 [38].

2) RBM-DNN vs. DNN: We first justified the effect of using RBM-DNN versus DNN on Aurora4. The results are listed in Table IX. Similarly, the two RBM-DNN systems that had one RBM front-end performed the best, at 13.2% and 13.1%. This further verifies that adopting the generatively trained RBM front-end is helpful, but too many RBMs also degrade performance. The RBM-DNN with 1 RBM layer and 6 discriminatively tuned DNN layers is then used as our baseline system.

TABLE IX
AURORA4 WER (%) PERFORMANCE OF DIFFERENT RBM-DNN SETUPS ("GEN" FOR PRE-TRAINED RBM LAYERS AND "DIS" FOR DISCRIMINATIVELY FINE-TUNED DNN LAYERS)

# of Hidden Layers			Test Set				Avg.
Total	gen	dis	A	B	C	D	
6	0	**6**	5.0	8.8	9.0	20.1	13.4
	1	5	**4.9**	**8.6**	9.1	19.8	13.2
	2	4	5.6	9.0	10.3	20.1	13.6
	3	3	6.5	9.7	11.5	21.1	14.5
7	0	7	5.0	8.8	**8.8**	20.1	13.3
	1	**6**	5.1	**8.6**	9.6	**19.4**	**13.1**
8	0	8	**4.9**	8.7	8.9	20.3	13.4
	2	**6**	5.4	8.9	9.7	19.8	13.4

TABLE X
AURORA4 WER (%) PERFORMANCE OF AM ADAPTATION WITH DIFFERENT LINS

System	LIN	Test Set				Avg.
		A	B	C	D	
S1	×	5.1	8.6	9.6	19.4	13.1
S1+LIN	full	4.8	8.1	**8.2**	18.7	12.4
	blk	**4.6**	8.0	8.3	18.3	12.2
	shd	**4.6**	**7.9**	**8.2**	**18.2**	**12.1**

TABLE XI
AURORA4 WER (%) PERFORMANCE OF SPECTRAL MASKING WITH DIFFERENT LIN ADAPTATIONS

System	LIN			Test Set				Avg.
	ME	AM	Type	A	B	C	D	
S3	×	×	-	4.7	9.2	8.7	20.2	13.5
S3+LIN	×	✓	full	4.5	8.9	8.2	19.6	13.1
			blk	4.5	8.6	8.1	18.8	12.7
			shd	4.6	8.5	8.1	18.7	12.5
S4	✓	×	full	4.7	9.0	8.5	20.0	13.4
			blk	4.6	9.1	8.5	19.8	13.3
			shd	4.6	8.9	8.5	19.6	13.1
S4+LIN	✓	✓	full	4.7	8.1	**7.3**	18.1	12.1
			blk	4.5	**7.9**	7.5	**17.7**	**11.8**
			shd	**4.4**	8.0	7.6	**17.7**	11.9

3) Acoustic Model Adaptation: We then evaluated the performance of different LIN adaptations on this AM RBM-DNN. One LIN transform was estimated for each test set using EBP with recognition hypotheses. Each set has 330 utterances, corresponding to 40 minutes of speech. 10% of them were held out as the validation set, which also uses recognition hypotheses as references to guide the learning process. The utterances are from 8 different speakers. A small difference from Aurora2, however, is that they have different SNRs. Hence, the estimated test set-dependent LINs captured only the noise type mismatches and are speaker- and SNR-independent. Results in Table X show that all the LINs are effective in reducing WERs on all the test sets, including clean set A. This could be attributed to the multi-style training data. Compared to Aurora2, the acoustic modeling complexity is much higher for this task. The AM cannot maintain both a superior clean performance and better generalization for noisy speech. Degradations on clean speech of the multi-style model are hence expected. The LIN transforms seem capable of fixing this problem. The largest relative improvement of 14.6% (from 9.6% to 8.2%) was obtained on set C. It clearly suggests the effectiveness of LINs in addressing channel mismatch. Although for the best LIN(shd), the absolute gain on set D (from 19.4% to 18.2%) is much larger than that on set B (from 8.6% to 7.9%). The relative improvement is almost the same, 8.3% on set D *vs.* 8.2% on set B.

4) Spectral Masking with LIN Adaptation: In this section, we justify our proposed spectral masking system. Firstly, the

TABLE XII
AURORA4 WER (%) PERFORMANCE BREAKDOWNS OF SYSTEM "S1 + LIN (BLK)", SYSTEM "S4 + LIN (BLK)" AND THE POSTERIOR INTERPOLATION SYSTEM "POSTINTER"

System	A	B						C	D						Avg.
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	
S1+LIN(blk)	4.6	5.2	7.8	9.3	9.0	7.6	9.0	8.3	10.6	19.7	22.0	19.8	18.0	19.8	12.2
S4+LIN(blk)	**4.5**	**4.8**	**7.6**	9.6	8.9	7.6	9.0	**7.5**	9.9	**18.3**	21.5	20.4	16.8	19.1	11.8
PostInter	4.6	4.9	**7.6**	**9.0**	**8.4**	**7.3**	**7.9**	7.6	**9.4**	**18.3**	**20.5**	**18.7**	**16.3**	**18.8**	**11.4**

direct use of ME RBM-DNN degraded the performance from 13.1% (system S1 in Table X) to 13.5% (system S3 in Table XI). However, our masking system did give improvements on set A (from 5.1% to 4.7%) and C (from 9.6% to 8.7%). The gains on set A come from the retraining of the AM on the masked training data. With multi-style data, the AM has to compromise between the clean and noisy data, which usually leads to degradations on clean test data compared to the AM trained purely on the clean training data. Although masking cannot completely remove the noise, it does reduce feature variations (Fig. 1(b) *vs.* Fig. 1(d)). This may lower the modeling complexity and improve the performance of the multi-style trained AM on clean speech. It is interesting to see improvements on speech with only channel distortions as masks are defined to remove additive noise. One probable explanation is that the scaling of the component-wise soft-masking is effectively doing a mean and variance normalization in the power spectrum domain. For set B and D, unreliable mask estimation was probably the reason for degradations. To improve the performance, on one hand, we can address the mismatches between the masked training and testing data by adapting the AM of system S3, *i.e.*, system S3 + LIN, which gives the WER of 12.5% (LIN(shd)). On the other hand, we can address the mismatches between the original training and testing features by adapting the ME of system S3 with LIN transforms estimated for the AM, *i.e.* system S4. Although system S4 only brings the performance back to the baseline performance (13.1%), the WER breakdowns differ a lot. This is similar to what we have observed on Aurora2. By further adapting the AM of system S4, which leads to system S4+LIN, we can achieve the best average WER of 11.8% (LIN(blk)) with spectral masking. Comparing S1 + LIN in Table X and our S4 + LIN, it can be seen that WER reductions are relatively small. But the differences in the generated hypotheses are statistically significant [40]. For LIN and LIN(blk), the p-values are all smaller than 0.001 and for LIN(shd), it is 0.018. These suggest there are statistically significant differences between the recognition hypotheses generated by these two systems despite their similar average WER performance.

5) Posterior Interpolation: To exploit differences between system S1 + LIN and system S4 + LIN, we simply averaged the two sets of posteriors. Only the block-diagonal version of LIN was experimented with and the results were tabulated in Table XII. The average WER of 11.4% and the performance gains on almost all the test sets clearly indicated the complementariness between these two systems. Adjusting the interpolation weight from 0.0 to 1.0 by 0.1 did not give any further improvement. To the best of our knowledge, this WER of 11.4% is currently the best reported performance on Aurora4.

6) Utterance-Based Adaptation: Throughout the research thus far, we have estimated the LIN from a set of adaptation data. Relaxing this requirement is more desirable for real world applications. In this experiment we justify the effectiveness of our proposed spectral masking system in an utterance-based adaptation scenario. One LIN was estimated for each test utterance. The learning was exactly the same as previous cases, except for the fact that no cross validation was used. Only one iteration of LIN estimation on each test utterance was carried out as no further gain can be obtained by doing more. Due to the rather limited data, only the LIN(shd) was evaluated. Results in Table XIII show that with LIN(shd), we can adapt system S1 from an average WER of 13.1% to 13.0%. To further reduce the number of model parameters, we kept only diagonal elements of the LIN transform [24], which is referred to as "dig". A slightly better AM adaptation performance (12.9%) can be achieved by doing this. Using LIN(dig) in our proposed S4 + LIN system, we reduced the average WER to 12.3%. Similarly, the posterior averaging further reduced it to 12.1%. Compared to using adaptation only (S1 + LIN), the masking system is much more effective.

TABLE XIII
AURORA4 WER (%) PERFORMANCE OF UTTERANCE-BASED LIN ADAPTATION

System	LIN Type	Test Set				Avg.
		A	B	C	D	
S1+LIN	shd	5.0	8.6	9.6	19.4	13.0
S1+LIN	dig	5.1	8.5	9.3	19.2	12.9
S4+LIN		**4.8**	8.2	**8.2**	18.3	12.3
PosterInter		5.1	**8.0**	8.8	**17.9**	**12.1**

V. CONCLUSIONS

In this paper, we have proposed an adaptive spectral masking system that consists of a mask estimation component and an acoustic model component, which are all based on Deep Neural Networks (DNNs). Spectral masking is used together with Linear Input Network (LIN) adaptation to achieve robust noise reduction. Since the estimation of LINs for mask estimation DNNs requires stereo data, the LINs estimated for acoustic model DNNs were used to adapt mask estimators during testing. For the reusing of the LINs to work well, the first layers of both DNNs were constrained to share the same parameters, which were learned during the pre-training stage. Besides improving the reliability of transformation sharing, the so-called "RBM-DNN" was also found to give a better recognition performance compared to the pure DNN-based acoustic models on noisy speech. By combining spectral masking for noise removal and Linear Input Network adaptation for mismatch reduction, we achieved the best average Word Error Rate (WER) performance of 4.6% on Aurora2 and 11.8% on Aurora4. The combination of our proposed spectral masking system and the baseline system through a simple posterior averaging further reduced the WERs to 4.3% on Aurora2 and 11.4% on Aurora4.

REFERENCES

[1] G. E. Dahl and D. Yu *et al.*, "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.

[2] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath *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.

[3] M. L. Seltzer, D. Yu, and Y. Q. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398–7402.

[4] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Proc. ICASSP*, 2012, pp. 4085–4088.

[5] B. Li, Y. Tsao, and K. C. Sim, "An investigation of spectral restoration algorithms for deep neural networks based noise robust speech recognition," in *Proc. Interspeech*, 2013, pp. 3002–3006, ISCA.

[6] B. Li and K. C. Sim, "Noise adaptive front-end normalization based on vector taylor series for deep neural networks in robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7408–7412.

[7] A. L. Maas and Q. V. Le *et al.*, "Recurrent neural networks for noise reduction in robust ASR," in *Proc. Interspeech*, 2012, pp. 22–25, ISCA.

[8] S. J. Rennie, P. Fousek, and P. L. Dognin, "Factorial hidden restricted Boltzmann machines for noise robust speech recognition," in *Proc. ICASSP*, 2012, pp. 4297–4300.

[9] J. Boldt, "Binary masking & speech intelligibility," Ph.D. dissertation, Aalborg Univ., Aalborg, Denmark, 2011.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] D. L. Wang and G. J. Brown *et al., Computational auditory scene analysis: Principles, algorithms, and applications*. New York, NY, USA: Wiley/IEEE, 2006.

[12] R. Lyon, "A computational model of binaural localization and separation," in *Proc. ICASSP*, 1983, pp. 1148–1151.

[13] G. J. Brown, "Computational auditory scene analysis: A representational approach," Ph.D. dissertation, University of Sheffield, Sheffield, U.K., 1992.

[14] D. L. Wang and G. J. Brown, "Separation of speech from interfering sounds based on oscillatory correlation," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 684–697, May 1999.

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

[16] W. Hartmann and A. Narayanan *et al.*, "A direct masking approach to robust ASR," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 21, no. 10, pp. 1993–2005, Oct. 2013.

[17] 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. Norwell, MA: Kluwer, 2005, ch. 12.

[18] D. L. Wang and U. Kjems *et al.*, "Speech intelligibility in background noise with ideal binary time-frequency masking," in *J. Acoust. Soc. Amer.*, 2009, pp. 2336–2347.

[19] W. Hartmann and A. Narayanan *et al.*, "Nothing doing: Re-evaluating missing feature ASR," *Reconstruction*, 2011, Article ID OSU-CISRC-7/11-TR21.

[20] M. L. Seltzer, B. Raj, and R. M. Stern, "A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition," *Speech Commun.*, pp. 379–393, 2004.

[21] S. Keronen and H. Kallasjoki *et al.*, "Mask estimation and imputation methods for missing data speech recognition in a multisource reverberant environment," *Comput. Speech Lang.*, pp. 798–819, 2012.

[22] J. F. Gemmeke and Y. J. Wang *et al.*, "Application of noise robust MDT speech recognition on the SPEECON and speechdat-car databases," in *Proc. Interspeech*, 2009, ISCA.

[23] A. Narayanan and D. L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7092–7096.

[24] A. Narayanan and D. L. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," in *Proc. OSU-CISRC-6/13-TR14*, 2013.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 232, 6088, pp. 533–536, 1986, Nature Publishing Group.

[26] H. Bourlard and N. Morgan, "Continuous speech recognition by connectionist statistical methods," *IEEE Trans. Neural Netw.*, vol. 4, no. 6, pp. 893–909, Nov. 1993.

[27] H. Bourlard and N. Morgan*, Connectionist speech recognition: A hybrid approach*. New York, NY, USA: Springer, 1994, vol. 247.

[28] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 1, pp. 161–174, Jan. 1994.

[29] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Proc. Eurospeech*, 1995, pp. 2183–2186.

[30] J. P. Neto, C. Martins, and L. B. Almeida, "Speaker-adaptation in a hybrid HMM-MLP recognizer," in *Proc. ICASSP*, 1996, vol. 6, pp. 3382–3385.

[31] J. P. Neto, C. Martins, and L. B. Almeida, "An incremental speaker-adaptation technique for hybrid HMM-MLP recognizer," in *Proc. ICSLP*, 1996, vol. 3, pp. 1293–1296.

[32] R. Gemello, F. Mana, and D. Albesano, "Linear input network based speaker adaptation in the dialogos system," in *Proc. IJCNN*, 1998, vol. 3, pp. 2190–2195.

[33] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *Proc. Interspeech*, 2010, pp. 526–529, ISCA.

[34] Y. Q. Wang and M. J. F. Gales, "TANDEM system adaptation using multiple linear feature transforms," in *Proc. ICASSP*, 2013, pp. 7932–7936.

[35] H. G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. Autom. Speech Recogn. (ASR'00): Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)* , 2000.

[36] N. Parihar and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," Inst. for Signal and Inf. Process, Mississippi State Univ., Tech. Rep, 2002.

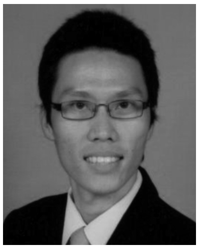
[37] D. Povey and A. Ghoshal *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, IEEE.

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

[39] D. Johnson, "Quicknet," ICSI [Online]. Available: http://www1.icsi.berkeley.edu/Speech/qn.html

[40] S. L. Chow, *Statistical significance: Rationale, validity and utility*. Thousand Oaks, CA, USA: Sage, 1996, vol. 1.

Bo Li is a research student at the School of Computing, National University of Singapore. He received the B.Eng. degree from the School of Computer, Northwestern Polytechnical University, China in 2008. His current research interests include machine learning, deep neural networks and robust automatic speech recognition.



Khe Chai Sim is an Assistant Professor at the School of Computing, National University of Singapore. He received the B.A. & M.Eng. degrees in electrical and information sciences from the University of Cambridge, U.K., in 2001. He then received the M.Phil. degree in computer speech, text and internet technology in 2002 from the same university. He joined the Machine Intelligence Laboratory, Cambridge University Engineering Department as a research student and completed his doctoral dissertation: "Structured Precision Matrix Modelling for Speech Recognition" in 2006 under the supervision of Professor Mark Gales. He has worked on the DARPA-funded EARS project from 2002-2005 and the GALE project from 2005-2006. He has also participated in various NIST evaluations: Rich Transcription (2004), Machine Translation (2006), Language Recognition (2007) and Speaker Recognition (2008). He is the recipient of the Google Faculty Research Award 2014.