Adaptive Very Deep Convolutional Residual Network for Noise Robust Speech Recognition

Tian Tan, *Student Member, IEEE*, Yanmin Qian, *Member, IEEE*, Hu Hu, Ying Zhou, Wen Ding, and Kai Yu, *Senior Member, IEEE*

***Abstract*—Although great progress has been made in automatic speech recognition, significant performance degradation still exists in noisy environments. Our previous work has demonstrated the superior noise robustness of very deep convolutional neural networks (VDCNN). Based on our work on VDCNNs, this paper proposes a more advanced model referred to as the very deep convolutional residual network (VDCRN). This new model incorporates batch normalization and residual learning, showing more robustness than previous VDCNNs.Then, to alleviate the mismatch between the training and testing conditions, model adaptation and adaptive training are developed and compared for the new VDCRN. This paper focuses on factor aware training (FAT) and cluster adaptive training (CAT). For FAT, a unified framework is explored. For CAT, two schemes are first explored to construct the bases in the canonical model; furthermore, a factorized version of CAT is designed to address multiple nonspeech variabilities in one model. Finally, a complete multipass system is proposed to achieve the best system performance in the noisy scenarios. The proposed new approaches are evaluated on three different tasks: Aurora4 (simulated data with additive noise and channel distortion), CHiME4 (both simulated and real data with additive noise and reverberation), and the AMI meeting transcription task (real data with significant reverberation).The evaluation not only includes different noisy conditions, but also covers both simulated and real noisy data. The experiments show that the new VDCRN is more robust, and the adaptation on this model can further significantly reduce the word error rate (WER). The proposed best architecture obtains consistent and very large improvements on all tasks compared to the baseline VDCNN or long short-term memory. Particularly, on Aurora4 a new milestone 5.67% WER is achieved by only improving acoustic modeling.**

***Index Terms*—Robust speech recognition, convolutional neural network, residual learning, factor aware training, cluster adaptive training.**

I. Introduction

In RECENT years, significant progress has been observed in automatic speech recognition (ASR) due to the introduction of deep neural network based acoustic models. On a wide range of large vocabulary continuous speech recognition (LVCSR) tasks, DNNs have shown great performance improvement over traditional Gaussian mixture models (GMMs) [1]–[4]. However, these systems still perform poorly in noisy environments (e.g., scenarios with additive noise [5]) and a magnified performance degradation has been observed under the distant (far-field) talking condition [6]. The low SNR in these noisy conditions makes DNNs more susceptible to the *mismatch problem*. Previous research has revealed that acoustic mismatch between training and testing still leads to a great performance degradation even with deep learning technologies [7], [8]. Thus, noise robustness is still a critical problem for making ASR widely adopted in real scenarios.

Many technologies [9]–[11] have been proposed to handle the difficult problem of mismatch between training and testing in the noisy speech recognition scenario. Those methods can be grouped into two categories: feature enhancement on the front-end (denoising or dereverberation) and acoustic modeling with adaptation on the back-end. Feature enhancement attempts to remove noise at the signal level [12], [13], while adaptation methods update the model parameters to better fit the unseen condition rather than denoise the features [5], [14].

Over the years, many techniques have been developed to adapt DNNs. For example, transformation based adaptation is an important category in DNN adaptation, e.g., linear input network (LIN) [15], feature discriminative linear regression (fDLR) [2], and linear output network (LON) [16]. The transformation could also be applied at a non-linear layer such as in LHUC [17], [18]. The adaptive training techniques have also been developed for DNN. In [19], layers are split into speaker-dependent and speaker-independent parts. Another idea is factor aware training, in which auxiliary features representing the non-speech variability are incorporated into DNNs. Features include i-vectors [20], [21], environment features [22], speaker codes [23], [24] and bottleneck features [25]–[28]. This method can be also treated as using a speaker or condition dependent bias [29]. Cluster adaptive training (CAT) has also been developed for DNNs: in [30]–[32], bases are built for DNNs to represent non-speech characteristics.

In this paper, we focus on the technologies on the back-end to improve the robustness of ASR systems. These technologies can be divided into two approaches. The first one is exploring more robust acoustic models, which can inherently limit the mismatch between training and testing. The second one is model

Manuscript received August 5, 2017; revised December 7, 2017 and February 18, 2018; accepted February 23, 2018. Date of publication April 12, 2018; date of current version May 8, 2018. This work was supported in part by China NSFC projects (U1736202 and 61603252), and in part Shanghai Sailing Program 16YF1405300. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sin-Horng Chen. *(Tian Tan and Yanmin Qian contributed equally to this work.) (Corresponding authors: Yanmin Qian and Kai Yu.)*

The authors are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: tantian@sjtu.edu.cn; yanminqian@gmail.com; mihawk@sjtu.edu.cn; zhouy49@sjtu.edu.cn; wen.ding@sjtu.edu.cn; kai.yu@sjtu.edu.cn).

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.2018.2825432

adaptation, constructed based on the new model structure, which can further improve the system performance in noisy conditions. For the more robust acoustic model, we focus on the convolutional neural network, which has been explored for acoustic model and yields a lower word error rate (WER) than standard fully connected feed-forward DNNs in many tasks [33], [34]. Recently, [35], [36] have designed very deep CNNs for speech recognition and gotten a significant WER reduction on telephone speech. Furthermore, in [37]–[39], our previous works have shown that VDCNNs particularly show noise robustness superior to other models in noisy scenarios and have also revealed some natural properties of VDCNNs. In this work, we introduce batch normalization [40] and residual learning [41] into our previous VDCNN structure. We discover that using these methods can further improve model robustness and reduce the mismatch in noisy scenarios. Various design aspects of the architecture are investigated in detail for the noisy scenarios.

In addition, some new adaptation techniques are developed based on this new very deep convolutional residual network (VDCRN). The first one is factor aware training (FAT). Typically, the auxiliary vector is concatenated with the input feature in DNNs or RNNs. Considering the different properties of the normal spectrum feature (e.g., FBANK) and auxiliary feature (e.g., i-vector), a parallel joint-learning structure is usually designed when applying factor aware training in CNNs [38], [42]. In this work, a unified framework is proposed: an *adaptation neural network* is learned to convert auxiliary features to a factor specific bias vector. Then, this specific bias vector could be added to any layers of a VDCRN including convolution layer and fully connected layer. The second technique developed in this paper is cluster adaptive training (CAT). We previously implemented CAT for DNN [30], whereas in this work we further explore a similar idea to use the filter or feature map as the basis for doing the CAT for VDCRN. Moreover, a factorized CAT structure is designed to incorporate multiple sources of non-speech variability into one complete model. A comprehensive investigation and an in-depth analysis of all those technologies are performed in this paper. Experimental results on the noisy Aurora4, CHiME-4 and AMI meeting transcription tasks show that applying the proposed techniques can obtain promising performance improvements.

The remainder of this paper is organized as follows. In Section II the conventional CNN-HMM hybrid system and structure of the VDCNN are first revisited, then the new very deep convolutional residual network (VDCRN), which shows better noise robustness, is presented. In Section III, we introduce factor aware training and cluster adaptive training based on VDCRN to alleviate the mismatch between the training and testing. Section IV describes the multiple pass scheme applied in our final system. We report experimental results in Section V and conclude the paper in Section VI.

II. Very Deep Convolutional Residual Network

A. Convolutional Neural Network

The convolutional neural network (CNN) has shown better performance than the traditional DNN in many speech recognition tasks [33], [34]. Recall that the inputs and outputs of each convolutional layer are several feature maps. Each feature map is a two-dimensional matrix. In speech recognition, the feature map at the input layer is a time-frequency map. Convolution is an operation that applies a filter to the feature map. The result of a convolution operation is still a feature map. We use $\otimes$ to represent it. A convolution layer consists of $\#outchannel \times \#inchannel$ filters. The i-th output feature map of layer l is given by

$$\mathbf{o}_i^l = \sigma\left(\sum_{j=1}^{N} W_{i,j}^l \otimes \mathbf{o}_j^{l-1} \oplus b_i^l\right) \tag{1}$$

where $\mathbf{o}_i^l$ and $\mathbf{o}_j^{l-1}$ are feature maps in the current layer l and previous layer $l-1$ respectively. $W_{i,j}^l$ is the filter between input feature map j and output feature map i at layer l. b_i^l is a bias applied to the whole feature map, $\oplus$ indicates each element in the feature map plus the same scalar b_i^l. σ is the activation function, which is typically a sigmoid or ReLU. N is the number of output feature maps. A pooling layer is a layer that performs down-sampling on the feature maps of the previous layer. In this work, max-pooling is used.

B. Very Deep Convolutional Neural Network

Recently very deep CNNs, which have many more convolution layers, have been successfully used in speech recognition [35], [36], [43]–[46] and particularly greatly outperformed DNN, RNN and shallow CNN models in noisy scenarios [37]–[39]. Our previous work shows the following key principles for designing very deep CNNs for speech recognition:

- Rather than using a large filter as in shallow CNNs (9×9 or 3×4 in [33], [34], a smaller filter is used in a VDCNN. What's more, zero padding is used before the convolution operation so the feature map resolutions can be preserved. In this way, it is possible to increase the number of convolutional layers.
- Compared to computer vision tasks, the size of the input feature map in speech recognition is relatively small. In addition to the adjustment of the size of filters and pooling, the input size needs to be enlarged to allow more convolution and pooling operations. So a 17×64 feature map is used as model input, i.e., a 17 frames context window with 64-dimension FBANK feature.
- For very deep CNNs, a pooling layer is added after at least two convolutional layers. The size of the output feature map at the top convolutional layer is relatively small, 2×2, which can reduce the model parameters. Moreover, to achieve a better trade-off within model complexity and size, the number of feature maps is increased gradually: which will only be doubled after some pooling layers.

The full structure of the VDCNN is shown in the middle block of Fig. 1 (enclosed with the blue dotted line). It contains 5 blocks separated by the pooling operation, and each block contains two convolutional layers and one pooling layer. There are 4 FC (fully connected) layers with 2048 nodes in each layer after the convolutional part. The model configuration, such as

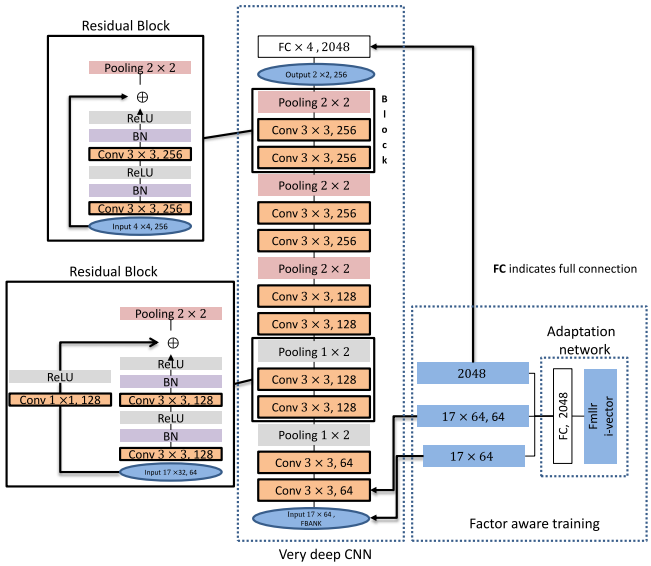

Fig. 1. The structure of VDCRN and FAT-VDCRN.

$[3 \times 3, 64]$ indicates that the layer uses a 3×3 filter and the output contains 64 feature maps.

C. Very Deep Convolutional Residual Network

Based on the VDCNNs, further extensions are developed in this work to better train the model with increased depth (a similar idea is also explored in other recent works [44]–[46]). Batch normalization (BN) and residual learning are mainly incorporated, which are motivated by the great success of ResNets in the image community.

- *Batch Normalization:* BN is an additional layer to normalize the mean and variance of a feature map within a mini-batch before the activation function is applied. After introducing BN, the gradients are well-behaved which allows the use of a larger learning rate and the training of deeper and more complicated networks. Furthermore, using BN can reduce the non-speech variability because it normalizes the activations over different frames.
- *Residual Learning:* Residual learning is proposed to ease the training of very deep neural networks. The key idea is to use a residual block (denoted as res-block for simplicity), which incorporates BN and skip connections across several convolutional layers. The very deep convolutional residual network used in this work is shown in Fig. 1. A residual block replaces each block in previous VDCNN (two convolutional layers with one pooling layer in VDCNNs). Two kinds of res-block are designed here, as shown in the left part of Fig. 1. Since the number of feature maps doubles in the first, second and fourth blocks, a convolutional layer with a 1×1 filter is applied in the corresponding skip connection to double the number of feature maps, as shown at the bottom left of Fig. 1. For the blocks with the same number of feature maps as the previous block, e.g., the third and fifth block, a direct skip connection can be used, as shown at the top left of Fig. 1.

In this work, we use this new structure for noise robust speech recognition, and we named our proposed model very deep convolutional residual network (VDCRN). The following experiments reveal that this new model particularly shows greater robustness than our previous VDCNN under noisy conditions, leading to a significant mismatch reduction within the clean and noisy data. More details will be given in Section V.

III. Adaptation and Adaptive Training for VDCRN

Adaptation and adaptive training are effective methods to reduce the mismatch between training and testing conditions. In this work, adaptation technologies are developed for the advanced VDCRN proposed in the previous Section. More specifically, factor aware training (FAT) and cluster adaptive training (CAT) are designed individually, and these adaptations can be performed on the different levels, such as speakers, noises, or channels, to enhance the system robustness.

A. Factor Aware Training - FAT

Factor aware training normally incorporates a vector that represents the acoustic condition information into the network training process to normalize the non-speech variability. In a DNN or LSTM, it works well by simply concatenating the auxiliary factor representations with the acoustic features, such as the i-vector for speaker [20], [21], noise energy for noise [22] and T60 for room reverberation [28]. Then the newly concatenated feature is used as the NN input. In previous works, auxiliary features have been applied to the CNN. However, since auxiliary vectors have no topography, they cannot be used as concatenated feature inputs for a CNN. In [38], [42], FAT was used with either shallow or very deep CNN structures, where the auxiliary feature was only connected to the fully-connected MLP layers. In [47], I-vectors were also incorporated into the convolutional layer by concatenating the feature with every localized frequency patch, so that the new filter size becomes 9×109 (with 100-dim I-vectors).

In contrast to their works, in this paper, auxiliary features are used in a unified framework, that is, using them to estimate a speaker dependent bias rather than concatenating them with acoustic features. Then the speaker dependent bias can be incorporated into any positions in a neural network, e.g., a convolutional layer or a fully connected layer. Compared to another recent work in [44], a shallow adaptation network is used to estimate the bias and then the speaker dependent bias is applied into different positions of the neural network in our approach, while they used a linear transformation and only integrated it into the convolutional layer.

The formulation of the proposed unified factor aware training is as follows:

$$\mathbf{b}^{sl} = V_2^l \sigma(V_1^l \mathbf{y}^s + \mathbf{p}_1^l) \tag{2}$$

$$\mathbf{o}^{sl} = \sigma(W^l \mathbf{o}^{l-1} + \mathbf{b}^{sl} + \mathbf{b}^l) \tag{3}$$

where $\mathbf{o}^{sl}$ is the speaker adapted hidden output of layer l. $\mathbf{b}^{sl}$ is the speaker dependent bias. In this work, a shallow adaptation NN with 1 hidden layer is used. $V_2^l, V_1^l, \mathbf{p}_1^l$ are the weight matrices and bias for the adaptation NN. $\mathbf{y}^s$ is the auxiliary feature. W^l is the weight matrix applied to the output from the previous layer and $\mathbf{b}^l$ is speaker independent bias. The FAT-DNN is a special case of this framework when $\mathbf{b}^{sl} = G^l\mathbf{y}^s$ where G^l is a linear transformation. The structure of the FAT-VDCRN is depicted in the bottom right of Fig. 1, a shallow adaptation neural network is learned to take the auxiliary vectors, e.g., i-vector, noise energy or T60, as the input and output a factor-specific bias. Then, this output is used as a bias so it can be integrated into the VDCRN at different layers. In this work, we compared three positions, including 1) acoustic feature input layer; 2) the output of the first convolutional layer; and 3) the output of the first fully connected layer, which follows the whole CNN block. When the bias is added to a convolutional layer, it should be reshaped to a 3-dim tensor. For example, if we apply this bias at the convolutional layer with size $[17 \times 64, 64]$, which means it contains 64 feature maps and each feature map is a 17×64 matrix. The dimension of the speaker dependent bias $\mathbf{b}^{sl}$ is 69632 and the adapted output feature map is given by

$$\mathbf{b}^{sl} = V_2^l \sigma(V_1^l \mathbf{y}^s + \mathbf{p}_1^l) \tag{4}$$

$$\mathbf{r}^{sl} = reshape(\mathbf{b}^{sl}, 17, 64, 64) \tag{5}$$

$$\mathbf{o}_i^{sl} = \sigma\left(\sum_{j=1}^{N} W_{i,j}^l \otimes \mathbf{o}_j^{l-1} \oplus b_i^l + \mathbf{r}_i^{sl}\right) \tag{6}$$

where $\mathbf{o}_i^{sl}$ is the ith feature map of speaker adapted hidden output of layer l. $\mathbf{r}^{sl}$ is the reshaped tensor. $\mathbf{r}_i^{sl}$ is the ith element in tensor, $\mathbf{r}^{sl}$: it is a matrix and its size is 17×64.

In this work, the new architecture is mainly evaluated with fMLLR features and i-vectors as auxiliary features. Feature-space Maximum Likelihood Linear Regression (fMLLR), also known as constrained MLLR (cMLLR), is an affine feature transform based adaptation. For each speaker s, an affine transform will be estimated by standard maximum likelihood estimation [48]. It has been shown that using fMLLR transformed features can improve the performance of DNNs compared to FBANK features [2]. I-vectors are another popular technique for speaker verification and recognition. They capture the most important information of a speaker in a low-dimensional representation. For each speaker, a vector is estimated which represents the coordinate of speaker s in the total variability subspace [49]. The detailed analysis and experiments with these auxiliary features can be found in the experimental section.

B. Cluster Adaptive Training - CAT

Different from doing adaptation with auxiliary features as in the previous section, several factorized adaptation methods have been proposed in the past few years to directly adapt the neural network in the model domain [30]–[32], [50]–[53]. A speaker dependent matrix is used to form a speaker dependent hidden layer. The difference between those methods is the structure used to estimate the speaker dependent matrix.

In CAT-DNN (Cluster adaptive training on DNN) [30]–[32], [50], [51], a basis structure was used, i.e., multiple matrices were used to form a matrix basis. So there are two sets of parameters in CAT-DNN: the canonical model and transforms.

- *Canonical model:* weight matrix bases

$$\mathcal{M} = \left\{\{M^1, ..., M^L\}, \{W^1, ..., W^K\}\right\} \tag{7}$$

  where $M^l = [W_1^l, ..., W_P^l]$ is the weight matrix basis of layer l, and P is the number of clusters. L is the total number of CAT-layers. W^k is the weight matrix of non-CAT layer k and K is the total number of non-CAT layers.
- *Transformations:* The speaker dependent interpolation vector is $\boldsymbol{\lambda}^{sl}$

$$\boldsymbol{\lambda}^{sl} = [\lambda_1^{sl}, ..., \lambda_P^{sl}]^\top \tag{8}$$

  where λ_c^{sl} is the interpolation weight for the cth cluster (base). The final adapted weight matrix for a given speaker s and the output are given by

$$W^{sl} = \sum_{c=1}^{P} \lambda_c^{sl} W_c^l \tag{9}$$

$$\mathbf{o}^{sl} = \sigma(W^{sl}\mathbf{o}^{l-1} + \mathbf{b}^l) \tag{10}$$

In [52], another structure using matrix decomposition was proposed. The weight matrix was first decomposed by singular value decomposition (SVD), $W_{n\times m} \approx U_{n\times P} V_{P\times m}$, then a speaker dependent square linear layer was applied to the bottleneck. So a speaker adapted weight matrix can be obtained by

$$W_{n\times m}^s = U_{n\times P} S_{P\times P}^s V_{P\times m} \tag{11}$$

where $S_{P\times P}^s$ is a speaker dependent square matrix. In [53], the speaker dependent square matrix was further decomposed to a diagonal matrix plus a low rank matrix.

$$S_{P\times P}^s = D_{P\times P}^s + P_{P\times c}^s Q_{c\times P}^s \tag{12}$$

where $D_{P\times P}^s$ is a diagonal matrix and $P_{P\times c}^s, Q_{c\times P}^s$ are two low-rank matrices. If we only look at the diagonal part, the formulation can be rewritten as following:

$$U_{n\times P} D_{P\times P}^s V_{P\times m} = \sum_{c=1}^{P} d_c^s \mathbf{u}_c \mathbf{v}_c{}^\top \tag{13}$$

where $\mathbf{u}_c$ is the cth column of U and $\mathbf{v}_c$ is the cth column of $V^\top$. So this structure can also be interpreted as interpolating multiple rank-1 matrix bases by a speaker dependent vector, which is equivalent to the method proposed in [51].

Motivated by these successful factorization based adaptation methods for DNNs, we intend to further explore its potential ability for noise-robust speech recognition. In this work, we extend the previous work and develop a new CAT architecture for the VDCRN. A similar idea was also proposed in [54], but there are some major differences between our work and theirs: 1) The first one is the choice of basis. In [54], a CNN layer was split into several sub-CNN layers and those sub-CNN layers were considered as the basis. In this work, feature maps and filters are explored as bases for applying CAT in the proposed

VDCRN. The method proposed in [54] is equivalent to using filters as the basis in our framework. 2) Another improvement in this work is that a factorized version of CAT is proposed in this work to model multiple factors simultaneously, subsequently increasing performance. 3) In addition, this work deploys CAT for VDCRN, which is much deeper than the baseline model in [54]. 4) Finally, our proposed methods were evaluated on all kinds of noisy tasks (Aurora4, Chime4 and AMI), including not only different noise types but also simulated and real noisy data. Significant and consistent improvements are obtained over all test conditions, which will be shown in Section V.

1) CAT in VDCRN: The choice of basis for CAT is the most important design decision. Two kinds of bases can be used when applying CAT in the proposed VDCRN.

- *Feature map bases:* At each convolutional layer, an output feature map is generated by summing up over all input feature maps, each convolved by its own filter as shown in equation 1. The first method utilizes each input feature map as a basis and interpolates them with a speaker dependent interpolation vector. The output feature map is given by

$$\mathbf{o}_i^{sl} = \sigma\left(\sum_{j=1}^{N} \lambda_j^{sl}(W_{i,j}^{l} \otimes \mathbf{o}_j^{l-1}) \oplus b_i^l\right) \tag{14}$$

where $\mathbf{o}_i^{sl}, \mathbf{o}_j^{l-1}$ are i-th and j-th feature map in two consecutive layers. λ_j^{sl} is a scalar coefficient for the cluster j at layer l for speaker s. It is worth noting that in this case, the number of clusters is equal to the number of input channels (feature maps).
λ_j^{sl} can be extended to a matrix, which will be applied to the feature maps via element-wise multiplication. The new speaker adapted output feature map is given by

$$\mathbf{o}_i^{sl} = \sigma\left(\sum_{j=1}^{N} \Lambda_j^{sl} \odot (W_{i,j}^{l} \otimes \mathbf{o}_j^{l-1}) \oplus b_i^l\right) \tag{15}$$

where Λ_j^{sl} is a matrix, and $\odot$ indicates element-wise multiplication. The structure is illustrated in the left part of Fig. 2, enclosed in the blue dotted line. There are four input feature maps and one output feature map, $W_{1,j}, 1 \le j \le 4$ is the filter and λ_j^s is the speaker dependent coefficient, which can be a scalar or matrix. Note that extending λ_j^{sl} to a matrix will dramatically increase the number of adaptation parameters. For example, the number of adaptation parameters when applying CAT at the first convolution layer of the VDCRN will increase from 64 to 69632 ($64 \times 17 \times 64$) if λ_j^{sl} is changed from a scalar to a matrix. The choice of using a scalar or a matrix is a trade off between complexity and the model ability, and it usually depends on the amount of adaptation data for each speaker.
- *Filter bases:* Another design uses a filter basis rather than a single filter for each input/output feature map pair. The speaker adapted output feature map is given by

$$W_{i,j}^{sl} = \sum_{k=1}^{P} \lambda_k^{sl} W_{i,j,k}^{l} \tag{16}$$

$$\mathbf{o}_i^{sl} = \sigma\left(\sum_{j=1}^{N} W_{i,j}^{sl} \otimes \mathbf{o}_j^{l-1} \oplus b_i^l\right) \tag{17}$$

where $W_{i,j}^{sl}$ is a speaker dependent filter for layer l given by interpolating the filter bases using a speaker specific vector λ^{sl}, and $W_{i,j,k}^{l}$ is the kth element of the filter bases. P is the number of filter bases.
In addition, equation 17 can also be rewritten as

$$\mathbf{x}_i^{sl} = \left(\sum_{k=1}^{P} \lambda_k^{sl} \sum_{j=1}^{N} W_{i,j,k}^{l} \otimes \mathbf{o}_j^{l-1}\right) \oplus b_i^l \tag{18}$$

$$\mathbf{o}_i^{sl} = \sigma(\mathbf{x}_i^{sl}) \tag{19}$$

where $\mathbf{x}_i^{sl}$ is the speaker adapted output and $\mathbf{o}_i^{sl}$ is the speaker adapted hidden output. So CAT can also be interpreted as splitting a convolutional layer into P sublayers to represent different speakers or environmental characteristics [54].
Similarly, each interpolation scalar coefficient λ_k^{sl} can also be extended to a matrix, which will be applied on the feature maps using element-wise multiplication. The formulation for the adapted output feature map becomes

$$\mathbf{x}_i^{sl} = \left(\sum_{k=1}^{P} \mathbf{\Lambda}_k^{sl} \odot \sum_{j=1}^{N} W_{i,j,k}^{l} \otimes \mathbf{o}_j^{l-1}\right) \oplus b_i^l \tag{20}$$

$$\mathbf{o}_i^{sl} = \sigma(\mathbf{x}_i^{sl}) \tag{21}$$

where $\odot$ denotes element-wise multiplication.

CAT with filter bases is illustrated in the right part of Fig. 2, enclosed in the blue dotted line. $W_{1,j,1}, W_{1,j,2}, 1 \le j \le 4$ are the filter bases. In this work, we use two clusters for filter bases. λ_j^s is the speaker dependent coefficient, which can be a scale or matrix. $W_{1,j}^s$ is a speaker adapted filter.

2) Factorized CAT in VDCRN for Robust ASR: In reality, there are many sources of non-speech variability, e.g., speaker, noise and channel, etc. To reduce the mismatch on all levels, the above CAT is further extended, and a factorized CAT is designed to do adaptive training and adaptation on multiple factors simultaneously. In this work, we focus on modeling both speaker and noise factors.

- The first factorized CAT structure models speaker and noise using separate bases, and the bases are built on filters. The adapted output feature map is given by

$$\mathbf{o}_i^{snl} = \sigma\left(\sum_{j=1}^{N} (W_{i,j}^{sl} \otimes \mathbf{o}_j^{l-1} + W_{i,j}^{nl} \otimes \mathbf{o}_j^{l-1}) \oplus b_i^l\right) \tag{22}$$

where $\mathbf{o}_i^{snl}$ is the ith feature map of speaker s, noise n of layer l. $W_{i,j}^{sl}, W_{i,j}^{nl}$ are a speaker adapted filter and a

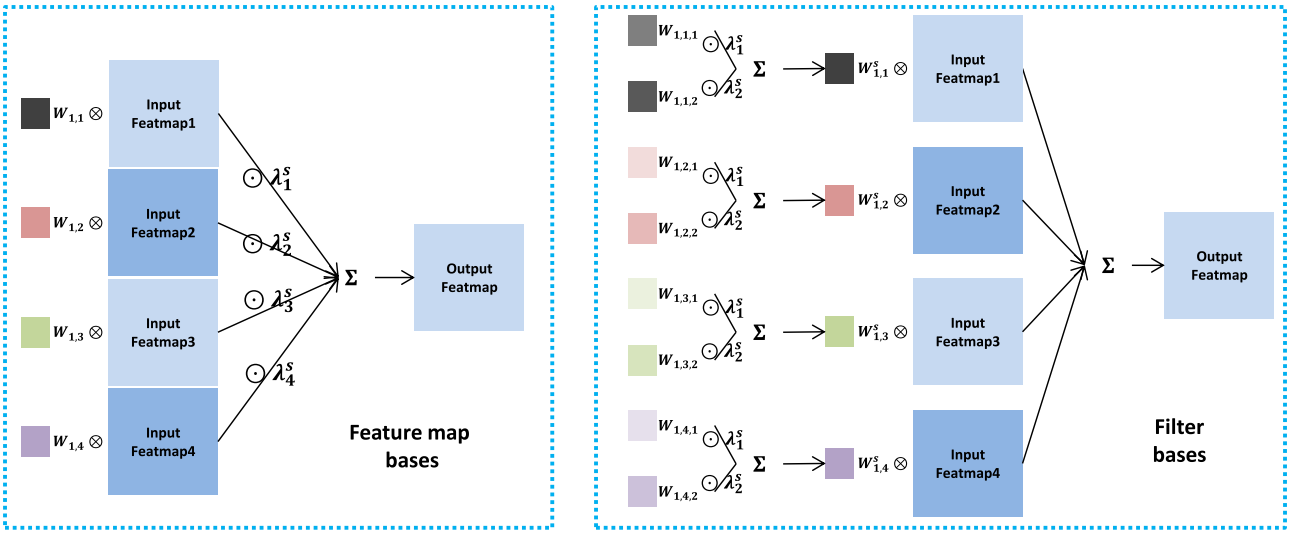

Fig. 2. Proposed cluster adaptive training for VDCRN. Left: CAT-VDCRN with feature map bases; Right: CAT-VDCRN with filter bases.

noise adapted filter respectively. They are calculated by interpolating using a speaker-dependent vector λ_k^{sl} and a noise-dependent vector λ_k^{nl} respectively.

$$W_{i,j}^{sl} = \sum_{k=1}^{P} \lambda_k^{sl} W_{i,j,k}^{sl} \tag{23}$$

$$W_{i,j}^{nl} = \sum_{k=1}^{P} \lambda_k^{nl} W_{i,j,k}^{nl} \tag{24}$$

where $W_{i,j,k}^{sl}, 1 \leq k \leq P$ are filter bases for speakers and $W_{i,j,k}^{nl}$ are filter bases for noises.

- Another method for factorized CAT makes the filter speaker dependent ($W_{i,j}^{sl}$) and layer bias noise dependent ($\mathbf{b}_i^{nl}$). The adapted output feature map is given by the following formula, and $W_{i,j}^{sl}$ is the same as in equation 23.

$$\mathbf{o}_i^{snl} = \sigma\left(\sum_{j=1}^{N}(W_{i,j}^{sl} \otimes \mathbf{o}_j^{l-1}) + \mathbf{b}_i^{nl} \oplus b_i^l\right) \tag{25}$$

IV. Multi-Pass System for Noise-Robust ASR

A multi-pass decoding framework that integrates all the above techniques is proposed for noise-robust speech recognition. The system pipeline is illustrated in Fig. 3. It consists of 5 stages shown as P1$\sim$ P5:

- *P1:* The front-end audio processing, including speech enhancement, such as beamforming, and feature extraction. In this work, beamforming is used for the CHiME4 and AMI tasks which contain multiple microphones, and the single channel audio is used directly for Aurora4. fMLLR transformed features are estimated using a GMM-HMM system. (It is noted that the front-end technique is not the main point of this work, so standard front-end processing is used for all tasks)
- *P2:* Factor aware training based acoustic models are built individually in the first round, including FAT-VDCRN and FAT-LSTM. For FAT-LSTM, we use the same structure as in [26] that directly concatenates the auxiliary features with the acoustic features.
- *P3:* A CAT-VDCRN system is built. The 1-best hypothesis from the FAT-VDCRN system is generated first and then

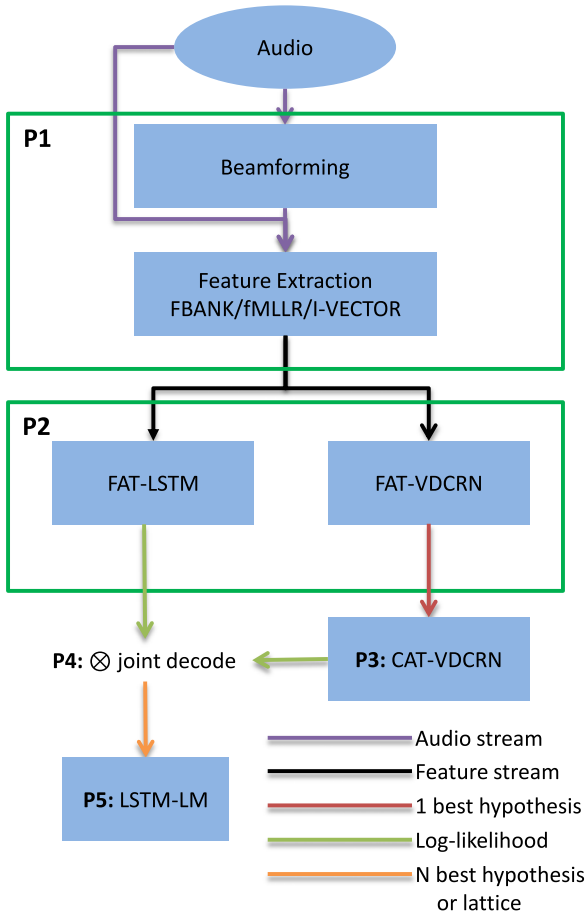

Fig. 3. Proposed multi-pass system within the deep learning framework.

used to do the CAT-based adaptive training on VDCRN model.

- *P4:* The CAT-VDCRN and FAT-LSTM are integrated to perform multi-model joint-decoding [38], which does system combination by interpolating the posteriors.
- *P5:* Based on the n-best hypotheses or word lattices generated from the P4 stage, a more advanced language model, such as LSTM-LM, can be used for rescoring to get a better result.

V. Experiments

In this section the proposed approaches are evaluated on three tasks: Aurora4, CHiME4 and AMI Meeting transcription, which have different noisy scenarios. The reasons we choose these three data set are listed as follows.

- Aurora4 has been a benchmark task for noise-robust speech recognition for a long time (more than a decade). All the noisy data are generated artificially with additive noise and channel distortion.
- The CHiME4 task is popular recently due to the success of the CHiME challenges. It has both simulated and real data with additive noise and reverberation, and multi-microphone audio is available for this task.
- Both Aurora4 and CHiME4 are based on WSJ0, which has a small vocabulary and is read speech. To make the evaluation more realistic, AMI meeting transcription is also used. It has more real spontaneous speech and there is significant reverberation in the recordings.

Accordingly the designed evaluation includes not only the different noisy conditions, but also covers both the simulated and real noisy data.

A. Experimental Setup and Baseline Systems

1) Data set:

- Aurora4 [55] is a medium vocabulary task. Transcriptions are based on the Wall Street Journal corpus (WSJ0) [56]. It contains 16 kHz speech data in the presence of additive noises and linear convolutional channel distortions, which were introduced synthetically to clean speech. The multi-condition training set contains 7138 utterances from 83 speakers, including clean speech and speech corrupted by one of six different noises at 10–20 dB SNR. Some utterances in the training set are from the primary Sennheiser microphone and others are from the secondary microphone. Similar to the training data, the same types of noise and microphones are used to generate the test set, grouped into 4 subsets: clean, noisy, clean with channel distortion, and noisy with channel distortion, which are referred to as A, B, C, and D, respectively.
- CHiME4, likes Aurora4, is based on the speaker-independent medium (5k) vocabulary subset of the Wall Street Journal (WSJ0) corpus, but with a multi-microphone pad on recording. Two types of data are employed: real data, which is recorded in real noisy environments (bus, cafe, pedestrian area, and street junction) from live talkers, and simulated data, which is generated by mixing clean speech data with noise. The training set contains 1600 real utterances from four speakers and 7138 simulated utterances from 83 speakers generated by mixing the WSJ0 SI-84 training set with 4 noisy backgrounds. The development set contains 3280 utterances and the evaluation test set contains 2640 utterances. There are 6 microphone channels that can be used for the front-end audio processing. In this work, the 6 channels data are pooled together for the model training, and CGMM-based minimum variance distortionless response (MVDR) beamforming [57] is used to do the front-end speech enhancement in the test stage.
- AMI [6] contains around 100 hours of meetings recorded in specifically equipped instrumented meeting rooms at three sites in Europe (Edinburgh, IDIAP, TNO). The acoustic signal is captured and synchronized by multiple microphones including individual head microphones (IHM, close-talk), lapel microphones and one or more microphone arrays. For the distant speech recognition in this work, the condition using multiple distant microphone (MDM) is evaluated, and audio in this condition is enhanced using the standard Kaldi recipe with BeamformIt. Our experiments adopted the suggested AMI corpus partition that contains about 80 hours and 8 hours in the training and evaluation sets, respectively [58].

2) Experimental Setup and Baseline Systems: GMM-HMM systems were first built to generate the senone alignments for later neural network training. In all tasks, the GMM-HMM system was built with Kaldi [59] using the standard recipes. The number of states in the Aurora4, Chime4 and AMI models were 2787, 1972 and 3916 respectively. The task-standard WSJ0 bi-gram and tri-gram language models were used for decoding on Aurora4 and CHiME4, respectively, with the WSJ0 5K-word closed vocabulary. A 50K-word dictionary and a tri-gram language model interpolated on the training transcripts and Fisher English transcripts were used for AMI decoding.

Standard DNN, CNN and LSTM models were trained as baselines. Recently, the CNN-LSTM-DNN (CLD) structure has been widely used in ASR [60], we also compared it in this work. All neural network models were trained using CNTK [61], running on one GPU card. The first iteration was trained with a relatively small learning rate of 0.1 and zero momentum. Starting from the second iteration, the learning rate was set to 1.0, with a momentum of 0.9. The learning rate halves when the validation loss stops decreasing. The standard testing pipelines in the Kaldi recipes were used for decoding and scoring.

- The baseline DNN and CNN used 40-dim FBANK features (25 ms frame window with 10ms frame shift) with $\Delta/\Delta\Delta$ and an 11-frame context window. The baseline DNN consists of 6 hidden layers of 2048 nodes. The baseline CNN utilizes the classical CNN configuration as in [33] with 2 convolutional layers and 4 fully-connected MLP layers. The minibatch was set to 256 in training.
- The LSTM was used for comparison in noisy scenarios. The basic LSTM system used a single frame of 40-dim FBANK features. There are 3 long short-term memory with projection (LSTMP) [62] layers in the model, and each

TABLE I
WER (%) COMPARISON OF DIFFERENT VDCRNS ON AURORA4

Systems	#Param	A	B	C	D	Avg.
DNN	30M	4.17	7.46	7.19	16.57	11.11
CNN	20M	4.11	7.00	6.33	16.09	10.64
LSTM	13M	3.92	7.21	6.63	15.94	10.68
CLD	19M	3.64	6.50	6.16	15.30	10.04
VDCNN	23M	3.27	5.61	5.32	13.52	8.81
VDCRN	23M	3.25	5.41	4.75	12.16	**8.10**
+ pre-act	23M	3.57	5.74	4.99	12.72	8.52
+ nomaxp	23M	3.57	5.74	5.38	13.69	8.97
+ gap	23M	3.51	5.87	5.42	13.97	9.14

`pre-act` indicates the pre-activation version res-block, `nomaxp` indicates convolution (stride=2) as a replacement for max-pooling, and `gap` indicates global average pooling instead of fully-connected layers. All modifications are tested independently of each other.

TABLE II
WER (%) COMPARISON OF DIFFERENT MODELS ON CHIME4

Systems	CHiME4			
	dev_real	dev_simu	eval_real	eval_simu
DNN	9.30	8.87	15.31	11.59
CLD	9.29	9.13	15.12	11.38
VDCNN	8.18	8.01	12.83	10.04
VDCRN	**7.48**	**7.23**	**11.60**	**8.69**

TABLE III
WER (%) COMPARISON OF DIFFERENT MODELS ON AMI MDM CONDITION

Systems	AMI	
	dev	eval
DNN	47.5	52.3
CLD	42.4	46.1
VDCNN	42.5	46.9
VDCRN	**40.9**	**44.7**

LSTMP layer has 1024 memory cells and 512 hidden nodes in the projection. In addition, a combination model, CNN-LSTM-DNN (CLD) system used a single frame of 64-dim FBANK features was also built. The CLD structure utilizes the standard configuration in [60], which contains 2 CNN layers, 3 LSTMP layers and 2 DNN layers. The output state label was delayed by 5 frames.Truncated BPTT was used to train these recurrent models with the chunk size set to 20 frames, and 40 utterances were processed in parallel to form a mini-batch. To ensure the stability of training, the gradient was clipped to the range of $[-1, 1]$ during the parameter update.

All neural networks developed in this work were trained using the cross-entropy criterion (CE) with the stochastic gradient descent (SGD) based back propagation (BP) algorithm.

B. Very Deep Convolutional Residual Network

A VDCRN was constructed following the description in Section II. For better comparison, the VDCNN model proposed in our previous work [37] is also built. These models were trained by CNTK. It is observed that VDCRN has the same fast-converge property as VDCNN, so it was trained with only 4 epochs. The learning rate was set as 0.1, 0.1, 0.025, 0.0016 for 4 epochs respectively. The momentum is 0 for the first epoch and 0.9 for the remaining epochs.

1) Evaluation of Different VDCRN Structures on Aurora4: The results on Aurora4 are shown in Table I. It is observed that using more parameter doesn't always improve the performance. The baseline DNN system has more parameters than the baseline CNN system but performed much worse. The VDCNN outperforms the CNN model by more than 20.0% relative WER reduction with only 3M more parameters. In addition, the improvement is quite noticeable when compared to the CLD model (Our experiments show that a larger size CLD model, with the same parameter number as VDCNN, gets no more improvement.). Moreover the VDCRN obtains another 8.0% relative improvement over the VDCNN without increasing the parameter count, and the improvement mainly comes from noisy scenarios.

Three variations on the basic VDCRN network were also evaluated. These are pre-activation, no max-pooling and global average pooling. All of them are added on basic VDCRN individually and tested independently of each other. Pre-activation [63] is a structure that pushes the activation function in front of the convolution operation. No max-pooling means that rather than using pooling, the stride was set to 2 at convolutional layer to perform downsampling. A global average pooling was used to replace the fully connected layer to decrease model complexity. All these variations were evaluated on Aurora4 set, and the results are shown in the bottom part of Table I.

Pre-activation didn't work well in ASR tasks, which contradicts the conclusion regarding image classification. No pooling is worse than using max-pooling layers especially for subsets C and D. The global average pooling layer, which is designed to replace the fully-connected layer, can indeed reduce the model size, but it resulted in a sharp decline in system performance. Accordingly the basic VDCRN was chosen for all the subsequent experiments. In summary, the proposed VDCRN, which is shown in Fig. 1, has a 20.0% relative improvement over the baseline CLD. In detail, the improvement in four subsets of Aurora4 are 12.0%, 17.0%, 23.0% and 21.0%, which means the new VDCRN is superior in noisy conditions.

2) Evaluation of VDCRN on CHiME4 and AMI: The proposed VDCRN was then evaluated on CHiME4 and AMI which contain real noise data. The results are shown in Tables II and III. The proposed VDCRN still can obtain about 10.0% relative gain on CHiME4 and 2.0% absolute WER reduction on AMI when compared to the strong VDCNN, demonstrating the better robustness of VDCRN when applied to both simulated and real noisy data.

3) Noise Robustness of VDCRN: Similar to the analysis on robustness of VDCNN in our previous work [37], we also did an in-depth analysis of the noise robustness of the VDCRN. The differences between the noisy feature outputs and clean feature outputs are measured. More specifically, using data in the test from Aurora4 we compute the average mean square error (MSE) between the outputs in noisy conditions (B, C.D) and the clean condition (A) at the last linear transformation before the softmax. As shown in Table IV, the VDCNN's significantly reduced MSE demonstrates the robustness provided by

TABLE IV
THE MEAN SQUARE ERROR (MSE) OF OUTPUTS BEFORE THE FINAL SOFTMAX OPERATION OF DIFFERENT MODELS IS CALCULATED

Systems	B	C	D
CNN	3.03	2.38	5.16
VDCNN [37]	1.76	1.49	2.91
VDCRN	1.06	0.88	1.80

The MSE is calculated on Aurora4 between the outputs using noisy inputs (B, C and D, respectively) and clean inputs (A).

TABLE V
WER (%) COMPARISON OF DIFFERENT STRUCTURES FOR FACTOR AWARE TRAINING ON AURORA4

Systems	position	A	B	C	D	Avg.
VDCRN	-	3.25	5.41	4.75	12.16	8.10
+ Bias	input	3.38	5.29	4.97	12.07	8.04
	conv	2.90	5.13	4.37	11.51	**7.65**
	mlp	3.01	5.16	4.07	11.78	7.76

bias-input indicates adding bias at the input feature. *bias-conv* indicates adding bias at the first convolutional layer. *bias-mlp* indicates adding bias at the first fully connected layer.

the increased number of convolutional layers, which is also the conclusion in our previous work [37]. The MSE results are also consistent with system performance: smaller MSE values correspond to improved performance in noisy scenarios. Compared to the VDCNN, the new proposed VDCRN achieves another large MSE improvement in all noisy conditions. Moreover, we find that the MSE gain on subset D is larger than the others, and this is also consistent with the WER in Table I, in which the WER reduction on subset D is the largest. All these observations further demonstrate that the proposed VDCRN acoustic model has better noise-robustness.

C. Factor Aware Training

In this subsection, factor aware training of the VDCRN was evaluated.

1) Evaluation of Different Structures for Factor Aware Training: As described in Section III-A, factor aware training can be treated as a factor specific bias. As shown in Fig. 1, factor aware training of three different layers was investigated, including the input feature layer (*bias-input*), the first convolutional layer (*bias-conv*) and the first fully connected layer (*bias-mlp*). 40-dim fMLLR transformed features were generated using all the data from a given speaker (the hypotheses were generated by the GMM-HMM system as the supervisions for estimating transformations), and an 11-frame context window was used to generate the final auxiliary feature. The related structures were evaluated on Aurora4. As shown in Table V, the proposed unified factor aware training framework can consistently improve the system performance. Factor-aware training of the first convolution layer has the best results and obtains a significant gain compared to the un-adapted VDCRN model.

2) Evaluation of Different Auxiliary Feature: Next, different auxiliary features are explored. In addition to the above fMLLR feature, i-vectors are also tested. A GMM with 2048 Gaussians is used to extract a 100-dimensional i-vector for each utterance, and these i-vectors are obtained using 40-dim MFCC features.

TABLE VI
WER (%) COMPARISON OF DIFFERENT AUXILIARY FEATURES ON AURORA4

Systems	A	B	C	D	WER
VDCRN	3.25	5.41	4.75	12.16	8.10
+ fMLLR	2.90	5.13	4.37	11.51	7.65
+ i-vector	3.53	6.00	5.23	12.48	8.55

TABLE VII
WER (%) COMPARISON OF CAT USING DIFFERENT BASES ON AURORA4

Systems	Base	λ^s	#Param	A	B	C	D	Avg.
VDCRN	-	-	-	3.25	5.41	4.75	12.16	8.10
+ CAT	fmap	scalar	64	3.40	5.80	5.68	12.14	8.34
		matrix	69632	2.69	4.17	3.81	8.96	**6.09**
	filter	scalar	2	3.33	5.52	5.31	13.12	8.61
		matrix	139264	2.65	4.06	3.38	8.95	**6.01**

fmap indicates building bases on feature map, *filter* indicates building bases on filters. *scalar* indicates the speaker-dependent interpolation parameter λ^s for each base is a scalar, and *matrix* indicates the speaker-dependent interpolation parameter λ^s for each base is a matrix. #Param indicates the number of adaptation parameter.

The results of factor aware training on the VDCRN with two kinds auxiliary features are shown in Table VI. It is observed that using fMLLR features can achieve a large gain, while in contrast i-vector didn't produce an improvement for VDCRN. This may because an i-vector represents the information about the speaker and acoustic conditions, while the fMLLR feature tries to suppress information about speaker and channel. Since batch normalization is applied in VDCRN which already normalizes the non-speech variability, using i-vectors to provide more speaker and conditions information is not helpful. However, fMLLR features provide another way to normalize the feature which may be complementary to batch normalization. In addition, fMLLR transformed features provide stronger information about the phonetic content of the test speech than FBANK features. Thus, using them in VDCRN also performs feature fusion. These results show that the appropriate auxiliary feature selection is important for the factor aware training.

D. Cluster Adaptive Training

In this subsection, cluster adaptive training of the VDCRN was evaluated. First we need to finish the adaptive training stage in the CAT-VDCRN model training. Then, during testing, the hypothesis was first generated by the above FAT-VDCRN model with fMLLR features. This hypothesis was used to estimate the speaker-specific parameters λ by back-propagation. This adaptation process used the same hyper parameters as training, where 4 epochs were used and the learning rate and momentum were set the same as the training.

1) Evaluation of Different CAT Structures: The experiments are performed to investigate CAT with different bases, different layers and different blocks.

- *Evaluation of CAT Using Different Bases:* As described in Section III-B1, the bases can be defined in two modes, i.e., feature map bases and filter bases. The speaker-dependent interpolation parameter λ^s can have two forms, scalar or matrix. In this experiment, CAT was applied at the first convolutional layer in the first block of the VDCRN, and the related results are illustrated in Table VII. It is observed

TABLE VIII
WER (%) COMPARISON OF APPLYING CAT ON DIFFERENT CONVOLUTIONAL LAYERS IN EACH BLOCK OF THE VDCRN ON AURORA4

CAT	A	B	C	D	Avg.
L1	2.65	4.06	3.38	8.95	6.01
+L2	2.95	4.17	3.70	8.55	**5.92**

L1 indicates only apply CAT on the 1st convolutional layer of the 1st block, and +L2 indicates apply CAT on both the 1st and 2nd convolutional layers of the 1st block.

TABLE IX
WER (%) COMPARISON OF APPLYING CAT ON THE DIFFERENT BLOCKS OF VDCRN ON AURORA4, B1 INDICATES BLOCK 1, B2 INDICATES BLOCK 2 AND B3 INDICATES BLOCK 3

Systems	A	B	C	D	Avg.
B1	2.95	4.17	3.70	8.55	**5.92**
B2	2.95	4.31	3.70	9.06	6.21
B3	2.84	4.28	3.66	9.49	6.37

CAT is applied to both convolutional layers in each block.

TABLE X
WER (%) COMPARISON OF DIFFERENT FACTORIZED CAT STRUCTURES ON CHIME4

Systems	Modes	dev_real	eval_real	Avg.
VDCRN	-	7.48	11.60	9.30
+ CAT	-	5.25	8.06	6.50
+ F-CAT	S1	5.19	7.92	**6.39**
	S2	5.06	8.11	6.40

F-CAT indicates factorized cluster adaptive training. *S1* indicates using separate bases. *S2* indicates using speaker-dependent filters and noise-dependent layer bias.

TABLE XI
WER (%) COMPARISON OF SYSTEM COMBINATION WITHIN CAT-VDCRN AND FAT-LSTM SYSTEMS ON AURORA4

Systems	A	B	C	D	Avg.
CAT-VDCRN (I)	2.95	4.17	3.70	8.55	5.92
FAT-LSTM (II)	3.75	6.87	5.64	13.99	9.61
(I) $\oplus$ (II)	2.84	3.96	3.33	8.49	5.77
(I) $\otimes$ (II)	2.82	3.90	3.53	8.26	**5.67**

$\oplus$ indicates MBR lattice combination, $\otimes$ indicates joint decoding.

that using a scalar interpolation weight λ^s is useless, and in contrast the matrix based speaker-dependent interpolation parameter can get very large improvements within the CAT-VDCRN, but also needs more adaptation parameters. Both basis types can give significant WER reduction with the appropriate interpolation weight λ^s, and the filter base seems slightly better. The system using filter bases with matrix interpolation weight λ^s reduces the WER from 8.10% to 6.01%, which is a 25.0% relative gain. We did further development based on this system in the following experiments.

- *Evaluation of CAT Applied on Different Layers:* The comparison of applying CAT on different convolutional layers of the first block is performed. In addition to doing CAT only on the first convolutional layer, it is extended to both layers in the first block. Table VIII shows that implementing CAT on both convolutional layers can get a further small improvement compared to the single-layer CAT adaptation. The gain is solely from the subset D which is the most mismatched condition. We analyzed and compared the different error distributions, including insertion, deletion and substitution, for individual subsets. It is observed that all degradations on condition A and C come from increased insertion errors. The improvement on condition D mainly comes from reduced deletion errors. This might be because applying CAT at multiple layers can enhance the model capability, so it tries to compensate more in the mismatched noisy environment which leads to recognizing more words during testing.
- *Evaluation of CAT Applied on Different Residual Blocks:* As shown in Fig. 1, the VDCRN is separated into several residual blocks by the pooling operations. The CAT structure can be applied to different blocks. We did experiments with CAT-VDCRN on Blocks 1-3, where CAT is applied to both convolutional layers in each block. The comparison of different block positions is shown in Table IX. It indicates that the performance decreases when applying CAT on higher residual blocks, and the first block is the best position for cluster adaptive training.

  The first line of Table IX is also the best configuration of the CAT-VDCRN on Aurora4. The WER has been reduced below 6.0% on Aurora, which is huge progress compared to the previous work on this task. Particularly, the WER on subsets B and C is close to that on A, and the WER on subset D is also dramatically reduced. This shows that the mismatch between training and testing can be reduced significantly in the noisy scenario and the proposed CAT-VDCRN is very promising.

2) Evaluation of Factorized CAT Model: Factorized CAT is applied to the VDCRN to simultaneously adapt to multiple non-speech variabilities. Because speaker and noise labels can only be obtained in CHiME4, the factorized CAT model was evaluated on the CHiME4 task. As described in Section III-B2, two structures of factorized CAT were investigated and the results are presented in Table X. The proposed CAT technology got a large improvement over the VDCRN baseline, which is the same conclusion as on Aurora4. Compared to the single factor based CAT, the factorized CAT version obtains small but consistent gains on the real noisy data. Moreover, factorization with separate bases seems more stable. More effective and accurate factorization methods need to be further explored in future work.

E. The Multi-Pass System for Noise-Robust ASR

1) System Combination Within Convolutional and Recurrent Models: Because the LSTM is another important acoustic model, the complementarity between VDCRN and LSTM is explored to further improve the performance in noisy conditions. Hence, an LSTM with factor aware training was also built. The auxiliary feature was a 100-dim i-vector, directly concatenated with the acoustic feature as input. The performance of the two single systems, e.g CAT-VDCRN and FAT-LSTM, are listed in the top part of Table XI. Two system combinations are also investigated. One is the normal MBR lattice combination

TABLE XII
WER (%) COMPARISON OF OUR FINAL MULTI-PASS SYSTEM ON AURORA4

Systems	A	B	C	D	Avg.
VDCNN	3.27	5.61	5.32	13.52	8.81
VDCRN	3.25	5.41	4.75	12.16	8.10
CAT-VDCRN (I)	2.95	4.17	3.70	8.55	5.92
FAT-LSTM (II)	3.75	6.87	5.64	13.99	9.61
(I) $\otimes$ (II)	2.82	3.90	3.53	8.26	**5.67**

TABLE XIII
WER (%) COMPARISON OF OUR FINAL MULTI-PASS SYSTEM ON CHIME4

Systems	dev_r	dev_s	eval_r	eval_s
VDCNN	8.18	8.01	12.83	10.04
VDCRN	7.48	7.23	11.60	8.69
CAT-VDCRN (I)	5.25	4.93	8.06	5.57
FAT-LSTM (II)	8.79	7.88	13.14	9.58
(I) $\otimes$ (II)	4.75	4.28	6.98	4.79
+LSTM LM	**2.90**	**2.65**	**4.67**	**2.79**

TABLE XIV
WER (%) COMPARISON OF OUR FINAL MULTI-PASS SYSTEM ON AMI MDM

Systems	dev	eval
VDCNN	42.5	46.9
VDCRN	40.9	44.7
CAT-VDCRN (I)	40.4	43.5
FAT-LSTM (II)	41.6	45.7
(I) $\otimes$ (II)	**38.2**	**41.3**

(denoted as $\oplus$) and another is the joint decoding which uses a weighted combination of state-level acoustic log likelihoods from individual models (denoted as $\otimes$) [38]. The results are shown in the bottom part of Table XI. It is interesting to observe that although the performance gap within the CAT-VDCRN and the FAT-LSTM is huge (3.7% absolute), the combination still can obtain a significant improvement, and the joint decoding scheme achieves a better performance than the MBR lattice combination.

2) Summary of the Final Multi-Pass System on All Tasks: Finally, the multi-pass systems are built for all three tasks, and the complete pipeline is depicted in Fig. 3. All the results are summarized in Tables XII–XIV. It is noted that only the CHiME4 task was performed with all 5 passes described in Section IV, while the other two tasks are only built to the 4th pass, i.e., no LSTM LM is applied on rescoring.[1]

Results show that the conclusions are consistent on all tasks: the proposed VDCRN model is superior to all the previous acoustic models in noisy scenarios, and the proposed adaptation technologies can dramatically reduce the mismatch between training set and testing set, which gives a very large WER reduction. The complete multi-pass system, integrated with all new techniques, performs well on both simulated and real noisy data, and all different noisy conditions. Particularly on Aurora4, a new milestone of 5.67% WER is achieved solely through improvements to the acoustic model.

[1]The LSTM LM rescoring is only given on CHiME4 because most of the CHiME4 challenge papers report this results. Considering that it is a noise-robust problem, acoustic-level technologies are the main focus, so the LSTM LM rescoring results are not given on the Aurora4 and AMI tasks.

VI. CONCLUSION

In this paper, batch normalization and residual learning are introduced into our previous very deep convolutional neural network (VDCNN). This new model, named very deep convolutional residual network (VDCRN), shows better robustness in noisy speech recognition. Then, adaptation using factor aware training and cluster adaptive training are developed for the VDCRN. Adapted VDCRNs can significantly improve system performance under all kinds of noisy conditions. Particularly, the factorized version of CAT-VDCRN, which does the adaptation on multiple levels to compensate multiple non-speech variabilities in one model, shows advanced superiority. Finally a multi-pass system using the proposed new technologies is proposed to achieve robust system performance in noisy scenarios.

The new approaches are evaluated on three noisy tasks: Aurora4, CHiME4 and AMI meeting transcription, which not only include different noisy conditions, but also cover both simulated and real noisy data. The new method obtains consistent and very large improvements on all noisy tasks compared to the previous VDCNN or LSTM. On Aurora4, it achieves a WER of 5.67% only through improvements to the acoustic model. To our knowledge this is the best published result on Aurora4.

ACKNOWLEDGMENT

Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

REFERENCES

[1] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[2] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE Automat. Speech Recognit. Understanding Workshop*, 2011, pp. 24–29.
[3] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. INTERSPEECH*, 2011, pp. 437–440.
[4] 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.
[5] Y. Wang and M. J. Gales, "Speaker and noise factorization for robust speech recognition," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 20, no. 7, pp. 2149–2158, Sep. 2012.
[6] T. Hain *et al.*, "Transcribing meetings with the AMIDA systems," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 20, no. 2, pp. 486–498, Feb. 2012.
[7] Y. Huang, D. Yu, C. Liu, and Y. Gong, "A comparative analytic study on the Gaussian mixture and context dependent deep neural network hidden Markov models," in *Proc. INTERSPEECH*, 2014, pp. 1895–1899.
[8] S.-Y. Chang and S. Wegmann, "On the importance of modeling and robustness for deep neural network feature," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4530–4534.
[9] 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.
[10] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Commun.*, vol. 16, no. 3, pp. 261–291, 1995.
[11] L. Shilin and K. C. Sim, "Joint adaptation and adaptive training of TVWR for robust automatic speech recognition," in *Proc. INTERSPEECH*, 2014, pp. 636–640.

[12] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "A minimum-mean-square-error noise reduction algorithm on MEL-Frequency cepstra for robust speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 4041–4044.

[13] T. Yoshioka and M. J. Gales, "Environmentally robust ASR front-end for deep neural network acoustic models," *Comput. Speech Lang.*, vol. 31, no. 1, pp. 65–86, 2015.

[14] X. Chen *et al.*, "An initial investigation of long-term adaptation for meeting transcription," in *Proc. INTERSPEECH*, 2014, pp. 954–958.

[15] 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.

[16] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2012, pp. 366–369.

[17] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014, pp. 171–176.

[18] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 24, no. 8, pp. 1450–1463, Aug. 2016.

[19] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6349–6353.

[20] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using I-vectors," in *Proc. IEEE Automat. Speech Recognit. Understanding Workshop*, 2013, pp. 55–59.

[21] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using I-Vectors," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 23, no. 11, pp. 1938–1949, Nov. 2015.

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

[23] 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. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7942–7946.

[24] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on Speaker Code," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6339–6343.

[25] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for DNN-Based speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4610–4614.

[26] T. Tan *et al.*, "Speaker-aware training of LSTM-RNNS for acoustic modeling," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 5280–5284.

[27] S. Kundu, G. Mantena, Y. Qian, T. Tan, M. Delcroix, and K. C. Sim, "Joint acoustic factor learning for robust deep neural network based automatic speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 5025–5029.

[28] Y. Qian, T. Tan, and D. Yu, "Neural network based multi-factor aware joint training for robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 24, no. 12, pp. 2231–2240, Dec. 2016.

[29] Y. Liu, K. Penny, and H. Thomas, "An investigation into speaker informed DNN front-end for LVCSR," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4300–4304.

[30] T. Tan, Y. Qian, M. Yin, Y. Zhuang, and K. Yu, "Cluster adaptive training for deep neural network," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4325–4329.

[31] C. Wu and M. J. Gales, "Multi-Basis adaptive neural network for rapid adaptation in speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4315–4319.

[32] M. Delcroix, K. Kinoshita, T. Hori, and T. Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4535–4539.

[33] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2015, pp. 1478–1482.

[34] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8614–8618.

[35] M. Bi, Y. Qian, and K. Yu, "Very deep convolutional neural networks for LVCSR," in *Proc. INTERSPEECH*, 2015, pp. 3259–3263.

[36] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4955–4959.

[37] Y. Qian, M. Bi, T. Tian, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 24, no. 12, pp. 2263–2276, Dec. 2016.

[38] Y. Qian and P. C. Woodland, "Very deep convolutional neural networks for robust speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2016, pp. 481–488.

[39] Y. Qian and T. Tan, "The SJTU CHIME-4 System: Acoustic noise robustness for real single or multiple microphone scenarios," in *Proc. 4th CHiME Speech Separation Recognit. Challenge*, 2016, pp. 70–72.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[42] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The IBM 2015 english conversational telephone speech recognition system," in *Proc. INTERSPEECH*, 2015, pp. 3140–3144.

[43] D. Yu *et al.*, "Deep convolutional neural networks with layer-wise context expansion and attention," in *Proc. INTERSPEECH*, 2016, pp. 17–21.

[44] W. Xiong *et al.*, "The microsoft 2016 conversational speech recognition system," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5255–5259.

[45] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 4845–4849.

[46] G. Saon *et al.*, "English conversational telephone speech recognition by humans and machines," in *Proc. INTERSPEECH*, 2017, pp. 132–136.

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

[48] M. J. Gales, "Maximum likelihood linear transformations for HMM-Based speech recognition," *Comput. Speech Lang.*, vol. 12, no. 2, pp. 75–98, 1998.

[49] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[50] T. Tan, Y. Qian, and K. Yu, "Cluster adaptive training for deep neural network based acoustic model," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 24, no. 3, pp. 459–468, Mar. 2016.

[51] L. Samarakoon and K. C. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 24, no. 12, pp. 2241–2250, Dec. 2016.

[52] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for Deep Neural Network," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6359–6363.

[53] Y. Zhao, J. Li, and Y. Gong, "Low-Rank plus diagonal adaptation for deep neural networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5005–5009.

[54] M. Delcroix, K. Kinoshita, A. Ogawa, T. Yoshioka, D. T. Tran, and T. Nakatani, "Context adaptive neural network for rapid adaptation of deep CNN based acoustic models," in *Proc. INTERSPEECH*, 2016, pp. 1573–1577.

[55] D. Pearce and J. Picone, "Aurora working Group: DSR front end LVCSR evaluation AU/384/02," *Inst. Signal Inf. Process.*, Mississippi State Univ., Mississippi, MS, USA, Tech. Rep, 2002.

[56] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Workshop Speech Natural Lang.*, 1992, pp. 357–362.

[57] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5210–5214.

[58] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Proc. IEEE Automat. Speech Recognit. Understanding Workshop*, 2013, pp. 285–290.

[59] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. IEEE Automat. Speech Recognit. Understanding Workshop*, 2011.

[60] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4580–4584.

[61] D. Yu *et al.*, "An introduction to computational networks and the computational network toolkit," Microsoft, Redmond, WA, USA, Tech. Rep. MSR-TR-2014–112, 2014.
[62] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. INTERSPEECH*, 2014, pp. 338–342.
[63] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

Tian Tan received the B.S degree in 2013 from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is currently working toward the Ph.D. degree in speech recognition. His current research interests include speech recognition and deep learning.

Yanmin Qian received the B.S degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. In 2013, he joined the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is currently an Associate Professor. From 2015 to 2016, he was also an Associate Research with the Speech Group, Department of Engineering, Cambridge University, Cambridge, U.K. His current research interests include the acoustic and language modeling in speech recognition, speaker and language recognition, key word spotting, and multimedia signal processing.

Hu Hu is currently working toward the Undergraduate degree at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include speech recognition and deep learning.

Ying Zhou received the B.S degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2016. She is currently working toward the Postgraduate degree at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. She is working on microphone array signal processing and her current research interests include speech signal processing and deep learning.

Wen Ding received the B.S degree from the College of Software, Jilin University, Changchun, China, in 2016. She is currently working toward the Postgraduate degree at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Her current research interests include speech recognition and deep learning.

Kai Yu received the B.Eng. degree in automation and the M.Sc. degree from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree from Cambridge University, Cambridge, U.K., in 2006. He then joined the Machine Intelligence Laboratory, Department of Engineering, Cambridge University. He is currently a Research Professor with Shanghai Jiao Tong University, Shanghai, China. His main research focuses on speech-based human–machine interaction including speech recognition, synthesis, language understanding, and dialogue management. He was selected into for 1000 Overseas Talent Plan (Young Talent) by the Chinese government and for the Excellent Young Scientists Project of NSFC China. He is a member of the Technical Committee of the Speech, Language, Music and Auditory Perception Branch of the Acoustic Society of China.