Gao *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:86
DOI 10.1186/s13634-016-0384-5

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Joint training of DNNs by incorporating an explicit dereverberation structure for distant speech recognition

Tian Gao[1], Jun Du[1*], Yong Xu[1], Cong Liu[2], Li-Rong Dai[1] and Chin-Hui Lee[3]

Abstract

We explore joint training strategies of DNNs for simultaneous dereverberation and acoustic modeling to improve the performance of distant speech recognition. There are two key contributions. First, a new DNN structure incorporating both dereverberated and original reverberant features is shown to effectively improve recognition accuracy over the conventional one using only dereverberated features as the input. Second, in most of the simulated reverberant environments for training data collection and DNN-based dereverberation, the resource data and learning targets are high-quality clean speech. With our joint training strategy, we can relax this constraint by using large-scale diversified real close-talking data as the targets which are easy to be collected via many speech-enabled applications from mobile internet users, and find the scenario even more effective. Our experiments on a Mandarin speech recognition task with 2000-h training data show that the proposed framework achieves relative word error rate reductions of 9.7 and 8.6 % over the multi-condition training systems for the cases of single-channel and multi-channel with beamforming, respectively. Furthermore, significant gains are consistently observed over the pre-processing approach using simply DNN-based dereverberation.

Keywords: Distant speech recognition, Dereverberation, Joint training, Deep neural network, Beamforming

1 Introduction

With the fast development of mobile internet, hands-free speech interaction with automatic speech recognition (ASR) system is natural and becoming more and more popular. In these application scenarios, speech signal is often corrupted by reverberation and background noise. Reverberation is the collection of reflected sounds from the surfaces in an enclosure like an auditorium. It is a desirable property of auditoriums to the extent that it helps to overcome the inverse square law drop-off of sound intensity in the enclosure. However, if it is excessive, it can make the sounds run together with the loss of articulation, muddy and garbled effects. Human listeners rarely encounter the problem of comprehending speech in reverberant environments. However, the room reverberation leads to the severe degradation of ASR performance compared with the close-talking condition [1, 2]. The word error rate (WER) is highly correlated to the reverberation time, namely T60, which is the time required for reflections of a direct sound to decay 60 dB. Typically, the higher the T60 values are, the more distorted the reverberated speech becomes.

In recent years, substantial progress has been made for distant/reverberant speech recognition by several important challenges, such as REVERB (REverberant Voice Enhancement and Recognition Benchmark) challenge [3], CHiME [4] challenge mainly for solving background noises, and ASpIRE (Automatic Speech Recognition In Reverberant Environments) [5]. Many techniques have been widely investigated, including front-end multi-channel and single-channel dereverberation techniques, and back-end acoustic modeling approaches.

The multi-channel signal processing methods include spatial filtering and channel selection. When the signals from the individual microphone with a known geometry are suitably combined, the array can function as a spatial filter for suppressing noise and reverberation. The signals

*Correspondence: jundu@ustc.edu.cn
[1] National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, JinZhai Road, Hefei, China
Full list of author information is available at the end of the article

are filtered and weighted so as to form a beam of enhanced sensitivity in the direction of the desired source and to attenuate sounds from other directions. Such beamforming techniques have been investigated in [6–8]. However, if the positions of the microphones are neither known nor fixed, the beamforming approaches become less effective. Then the channel selection is an alternative approach. Wolf et al. [9] described, analyzed, and compared several signal-based and decoder-based measures of signal quality and applied them to the problem of channel selection in multi-microphone environments. And Himawan et al. [10] proposed a channel selection approach for selecting reliable channels based on a criterion operating in the short-term modulation spectrum domain for distant speech recognition. More recently, deep learning-based beamforming methods combined with acoustic modeling were investigated in [11, 12].

Inverse filtering [13] is one of the commonly used single-channel speech dereverberation techniques. The dereverberated signal is estimated by convolving the reverberant signal with the inverse filter. However, in many situations, the inverse filter cannot be directly determined or accurately estimated. Furthermore, this approach assumes that the room impulse response (RIR) function is minimum-phase which is not always satisfied in real practice. The recent breakthrough of deep learning [14, 15], and the applications of deep neural networks (DNNs) in speech signal processing area [16–18], creates a new direction of single-channel dereverberation. Han et al. [19] implemented the supervised learning approach based on DNNs to perform speech dereverberation. They used DNNs to learn a spectral mapping from corrupted speech to clean speech for dereverberation and denoising. This supervised learning approach boosted ASR results in a range of reverberant and noisy conditions. In [20], the DNN-based speech dereverberation was verified to be effective to reverberant speech recognition with clean-condition training.

So far, one of the most powerful methods for reverberant speech recognition is the use of multi-condition training. Many results from different research groups at the REVERB [21] and ASpIRE challenges have shown that the increased diversity of reverberation conditions for multi-condition training usually improves the robustness of acoustic models due to a well acoustic-condition match of the training and testing data. Reverberant speech is usually generated by convolving clean speech signals with RIRs measured in the target environment [22, 23]. In [24], some novel methods for taking advantage of reverberant speech training in modern DNN-based hidden Markov model (DNN-HMM) systems were proposed. Based on multi-condition training, feature-level dereverberation by deep autoencoders (DAEs) has been investigated in [25, 26]. In these works, DAEs were trained using reverberant speech features as input and clean speech features as learning targets. Acoustic models were retrained using the reconstructed features. Weninger et al. [27, 28] have shown that deep recurrent neural networks (RNNs) are also suitable for feature enhancement of reverberant speech signals. Recently, Mimura et al. [29] augmented the input of the autoencoder based on long short-term memory (LSTM) [30] with phone-class information (denoted as pLSTM). The results show that DNN-based pDAE (DAE with phone-class information) slightly outperformed pLSTM on real testing data. It is noted that the front-end feature dereverberation and the back-end acoustic modeling of these methods via DNNs or RNNs were trained separately.

In [31], an effective joint training procedure was proposed for noise robust speech recognition. In this DNN-based hybrid framework, the front-end for feature denoising and the back-end for acoustic modeling were jointly optimized. A joint training procedure was also proposed in [32] to combine masking DNN with back-end DNN. However, fixed layers were used to perform middle-stage masking post-processing and dynamic feature calculation operations. Instead, the joint training in [31] can seamlessly connect front-end and back-end DNNs, as the output of feature mapping DNN is exactly the input of acoustic modeling DNN. Multi-task learning (MTL) of DNN is also a machine learning scheme to combine different tasks. The motivation of MTL is to improve the generalization of the target task by leveraging the other tasks. When the tasks are properly chosen, the knowledge learned from one task could be made useful to the other tasks [11, 33, 34]. Compared with joint training which incorporates an explicit functional structure, MTL can be viewed as an implicit method.

In this study, we aim to jointly optimize a front-end regression DNN for feature dereverberation and a back-end classification DNN for acoustic modeling. Traditionally, a multi-condition training set of reverberant data can be simulated by using different RIRs and clean speech data. Furthermore, for DNN-based dereverberation, the learning targets are also clean speech. In both cases, the high-quality clean speech data are necessary and also difficult to be largely collected in real applications. With our joint training strategy, we can relax this constraint by using large-scale diversified real data as the targets which are easy to be collected via many speech-enabled applications from mobile internet users. Surprisingly, our experiments show that the system built with real user data can even outperform that using recorded clean speech in case of the same amount of training data. As for joint training, the new structures by utilizing both dereverberated and original reverberant features can effectively improve recognition accuracy over the conventional one in which only dereverberated features are used. Besides, we also

verify the effectiveness of the proposed ASR system for multi-channel testing data with beamforming front-end. Further gains are consistently observed, indicating that the explicit dereverberation structure in the joint training framework is still effective when combining linear beamforming techniques.

The remainder of the paper is organized as follows. In Section 2, we give a system overview. In Section 3, we adopt a DNN-based speech dereverberation module as the pre-processor for acoustic modeling. In Section 4, we introduce the details of several DNN-based joint training structures. In Section 5, we report experimental results, and finally we conclude the paper in Section 6.

2 System overview

The overall flowchart of our proposed ASR system is illustrated in Fig. 1. According to [35], reverberation is usually formulated as:

$$y(t) = s(t) * h(t) + \alpha n(t) \tag{1}$$

where reverberant signal $y(t)$ is obtained by convolving close-talking speech signal $s(t)$ with RIRs $h(t)$. In the training stage, we first use this simulation method to generate a large amount of reverberant data. Since we mainly focus on the effects of reverberation, white noise $n(t)$ controlled by the gain α in order to obtain different SNRs is added to the reverberant speech to simulate background noise. In this paper, we set SNR at 40 dB. Then the training samples are processed to extract log Mel-filterbank (LMFB) [36] features followed by mean normalization. We also augment the LMFB with pitch-related features [37]. Next, we use a joint training procedure to train DNNs for both front-end feature dereverberation and back-end acoustic modeling. Besides, different joint training structures are explored for distant speech recognition.

In the recognition stage, two cases are considered. The first one is a conventional single-channel speech recognition system as shown in Fig. 1. The other one is a multi-channel speech recognition system. In [38, 39], weighted prediction error (WPE) plus beamforming yielded good performances for REVERB challenge and CHiME-3 challenge [40], respectively. Based on this, WPE-based dereverberation [41] is first carried out with a linear time-invariant filter. Next, we use a traditional beamformer [6] to extract beamformed speech signal from the multi-channel dereverberated signals. This is a linear and multiple input single output process. After beamforming, the one-channel beamformed signal will be used to verify the effectiveness of the proposed ASR system which incorporates a nonlinear front-end. For both the single-channel and multi-channel systems, normal recognition is conducted with hybrid DNN-HMM.

3 DNN as a Pre-processor for ASR

For comparison with our joint training method, we first adopt a DNN-based pre-processor which has been used in [42] for noisy speech recognition and [19, 20] for reverberant speech recognition. In detail, this DNN is trained to predict clean log-power spectral (LPS) features given the input reverberant LPS features with acoustic context,

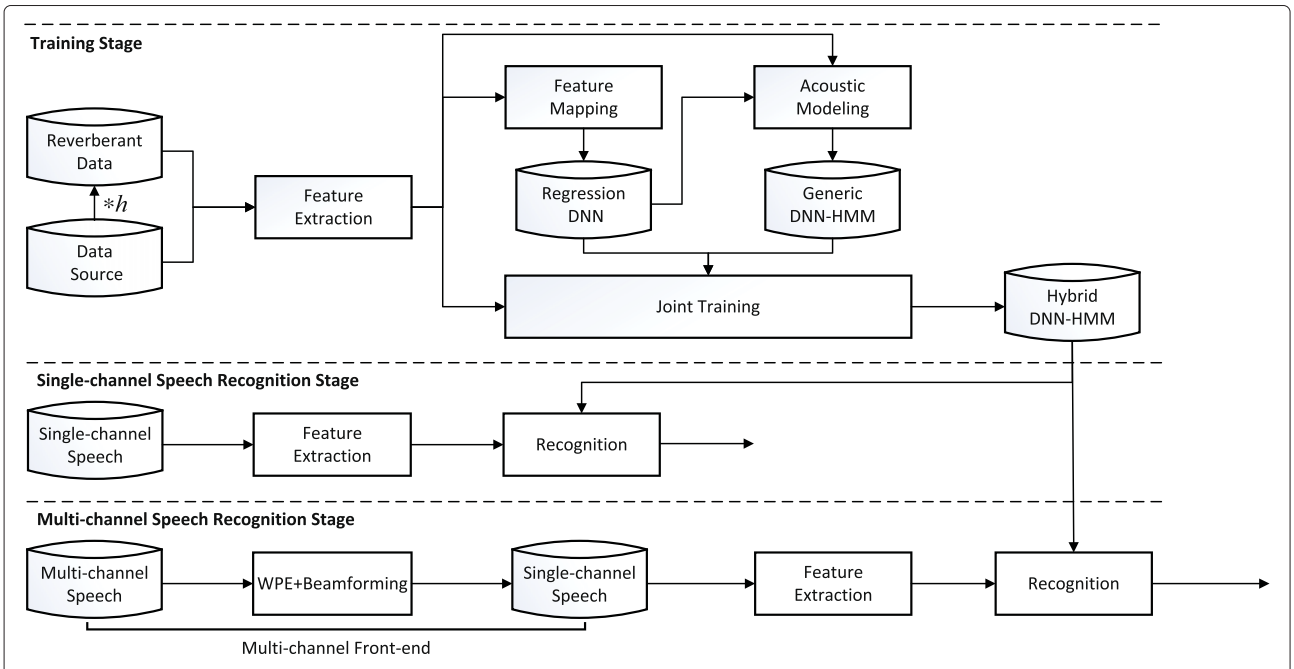

Fig. 1 Overall development flow and architecture of our proposed solution

which is shown in Fig. 2. The reason why we use LPS features rather than LMFB features is that all speech information can be retained in this domain and good listening quality can be obtained from the reconstructed clean speech according to [16]. The acoustic context information along both time axis (with multiple neighboring frames) and frequency axis (with full frequency bins) can be fully utilized by DNN to improve the continuity of estimated clean speech, and context information is more important in reverberant condition.

In the training of this regression DNN, we aim at minimizing mean squared error between the DNN output and the reference clean features based on randomly initialized weights. In the dereverberation stage, the well-trained DNN model is fed with the features of reverberant speech to generate dereverberated LPS features. The additional phase information is calculated from original reverberant speech. Finally, an overlap-add method is used to synthesize the waveform of the estimated clean speech.

The well-trained speech dereverberation DNN will be used as a pre-processor for acoustic modeling. We extract acoustic features from the dereverberated training speech, and the DNN acoustic model constructed using reverberant speech features can be further optimized by using the dereverberated features as input. This simple fine-tuning procedure of DNN is not only faster than retraining from scratch but also generates better recognition performance in [42]. In the recognition stage, after DNN pre-processing and feature extraction of unknown utterance, normal recognition is conducted. We denote this method as DNN-PP. To better understand this training strategy, we further illustrate the system procedure in Algorithm 1.

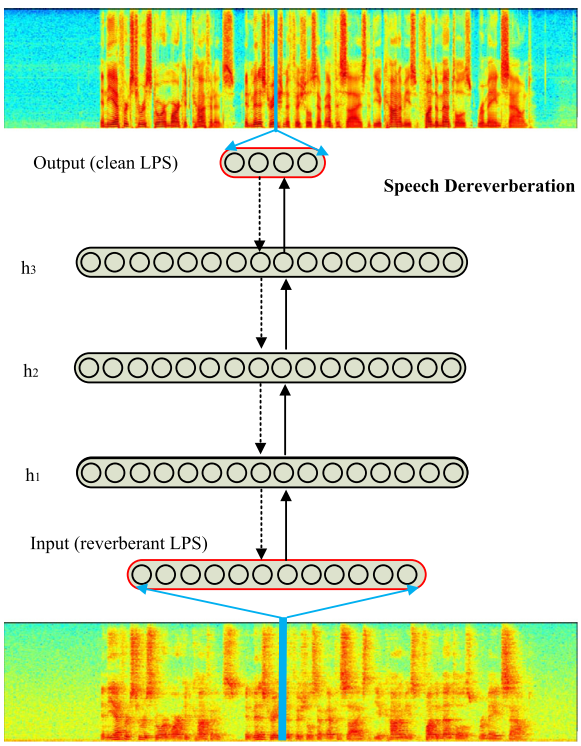

Fig. 2 DNN-based speech dereverberation

Algorithm 1 : DNN as a pre-processor for ASR (DNN-PP)

Step1: Pre-processor training

1. Extract reverberant and clean log-power spectral (LPS) features from all training utterances.
2. Train dereverberation DNN with reverberant-clean LPS feature pairs under minimum mean square error (MMSE) criterion.

Step2: Acoustic model training

1. Train a baseline DNN acoustic model using reverberant acoustic features based on randomly initialized weights.
2. Generate dereverberated speech waveforms using DNN-based pre-processor, and then extract LMFB acoustic features from all training utterances.
3. On top of the baseline DNN as an initialization, the acoustic model of dereverberated speech can be further optimized by only changing the input of DNN from original reverberant features to dereverberated features.

Step3: Recognition

1. Generate dereverberated testing speech using DNN-based pre-processor, and then extract acoustic features.
2. Feed the acoustic features through DNN acoustic model to generate senone posterior probability.

4 New structures for joint training

4.1 Conventional joint training structure

In [31], joint training framework was verified more effective than DNN-PP for noisy speech recognition, in which the front-end for feature denoising and the back-end for acoustic modeling were jointly optimized. To address reverberant condition, we adopt the front-end and back-end DNNs as shown in Fig. 3. Specifically, the front-end is a DNN-based feature level dereverberation module which maps the input reverberant features to the desired clean features. In the supervised learning stage, we aim at minimizing mean squared error between the DNN output and the reference clean features based on randomly initialized weights:

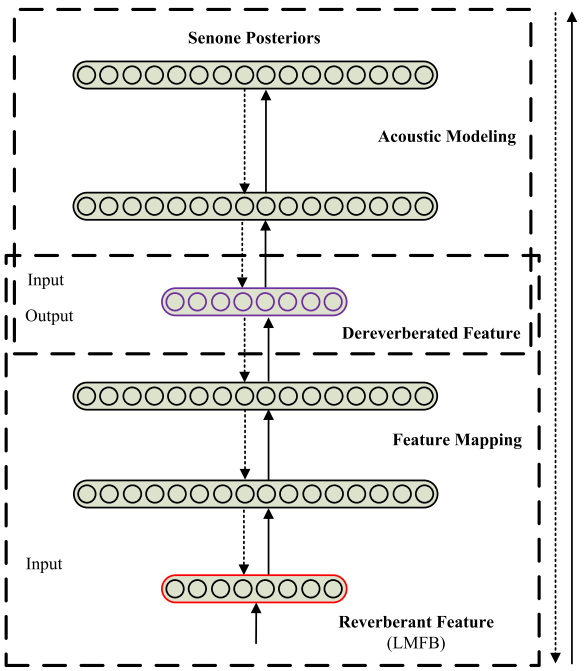

Fig. 3 Conventional joint training of front-end and back-end DNNs (DNN-JT1)

$$E = \frac{1}{N}\sum_{n=1}^{N} \|\hat{\boldsymbol{x}}_{n-\tau}^{n+\tau}(\boldsymbol{y}_{n-\tau}^{n+\tau}, \boldsymbol{W}, \boldsymbol{b}) - \boldsymbol{x}_{n-\tau}^{n+\tau}\|_2^2 + \kappa\|\boldsymbol{W}\|_2^2 \quad (2)$$

where $\hat{\boldsymbol{x}}_{n-\tau}^{n+\tau}$ and $\boldsymbol{x}_{n-\tau}^{n+\tau}$ are the nth $D(2\tau+1)$-dimensional vectors of estimated and reference features, respectively. $\boldsymbol{y}_{n-\tau}^{n+\tau}$ is a $D(2\tau+1)$-dimensional vector of input reverberant features with neighboring left and right τ frames as the acoustic context. $\boldsymbol{W}$ and $\boldsymbol{b}$ denote all the weight and bias parameters, respectively. κ is the regularization weighting coefficient to avoid overfitting. The objective function is optimized using back-propagation with a stochastic gradient descent method in mini-batch mode of N sample frames.

After feature mapping, we use the dereverberated features for acoustic modeling. We employed a hybrid DNN framework to perform joint training of DNNs for both feature mapping and acoustic modeling. In the hybrid DNN, we directly stack the acoustic modeling layers on top of the feature mapping layers. The output layer of feature mapping becomes the input layer for acoustic modeling, which is also a linear hidden layer of the whole network. Using the same object function as the back-end DNN, namely the cross entropy (CE) criterion, all weights are retrained. After joint training, the hybrid DNN can yield a better recognition performance which can be explained as the feature mapping network is refined to enable a better phone classification instead of optimizing the original MMSE criterion.

In the recognition stage, a normal decoding process is conducted using original reverberant features and the hybrid DNN. We denote this joint training structure as DNN-JT1. To better understand this training procedure, we further illustrate it in Algorithm 2.

Algorithm 2 : Training procedure of conventional joint training structure (DNN-JT1)

Step1: Front-end DNN training

1. Extract reverberant and clean LMFB features from all training utterances.
2. Train feature mapping DNN with reverberant-clean feature pairs under MMSE criterion.

Step2: Back-end DNN training

1. Train a baseline DNN acoustic model using reverberant acoustic features based on randomly initialized weights.
2. Stack the baseline model layers on top of the mapping style dereverberation layers to get a hybrid DNN framework.
3. Fix the front-end layers to retrain the back-end acoustic model layers. We denote this intermediate model as DNN-FM1.

Step3: Joint training of front-end and back-end DNNs

1. Optimize the front-end and back-end layers as a whole network under CE criterion with reverberant features. This is the joint training step and we denote the hybrid model as DNN-JT1.

4.2 New joint training structure

Inspired by our recent work [43] for CHiME-3 challenge, we design a new joint training structure for distant speech recognition. In [43], original noisy and enhanced features can be concatenated as the input of back-end DNN. This early fusion strategy was verified to be effective for noisy speech recognition. Results from [44–46] also demonstrate that original feature, dereverberated feature and reverberation feature could supply compensatory information to acoustic modeling.

In this new structure, we splice the output of front-end dereverberation DNN with original reverberant features to feed the back-end DNN for acoustic modeling as shown in Fig. 4. It should be noted that the back-end weights in this structure are randomly initialized due to the unequal dimension between the spliced features and the input of pre-trained back-end DNN used in DNN-JT1. We denote this new joint training structure as DNN-JT2 with corresponding procedure in Algorithm 3.

Algorithm 3 : Training procedure of new joint training structure (DNN-JT2)

Step1: Front-end DNN training

1. Extract reverberant and clean LMFB features from all training utterances.
2. Train feature mapping DNN with reverberant-clean feature pairs under MMSE criterion.

Step2: Back-end DNN training

1. Stack randomly initialized acoustic model layers on top of the dual-output (reverberant and dereverberated) feature mapping layers to get a hybrid DNN framework.
2. Fix the front-end layers to retrain the back-end DNN layers. We denote this intermediate model as DNN-FM2.

Step3: Joint training of front-end and back-end DNNs

1. Optimize the front-end and back-end layers as a whole network under CE criterion with reverberant features. We denote the hybrid model as DNN-JT2.

4.3 New joint training structure with a learned connection layer

The randomly initialized back-end used in DNN-JT2 is different from the pre-trained back-end used in DNN-JT1. Alternatively, we use a linear connection layer to concatenate the dual output (original reverberant and dereverberated feature) front-end and the pre-trained back-end as shown in Fig. 5. The connection layer can provide a linear feature transformation and dimensionality reduction.

Intuitively, the connection layer can be randomly initialized and updated during the network training. We denote this new joint training structure with the learned connection layer as DNN-JT3 described in Algorithm 4.

4.4 New joint training structure with a fixed connection layer

One common problem of the learned connection layer in DNN-JT3 is overfitting which can be verified in the following experiments. When overfitting happened, the connection layer could not use the dereverberated and reverberant feature properly. As a solution, we use a fixed $2D \times D$ matrix $\mathbf{A}$ as shown in Eq. (3) to implement the linear connection layer, where D is the dimension of the input dereverberated features. This design simply averages the dereverberated and reverberant features and is similar to the average pooling operation in convolutional neural network (CNN) [47]. We denote this new joint training structure with the fixed connection layer as DNN-JT4 illustrated in Algorithm 5.

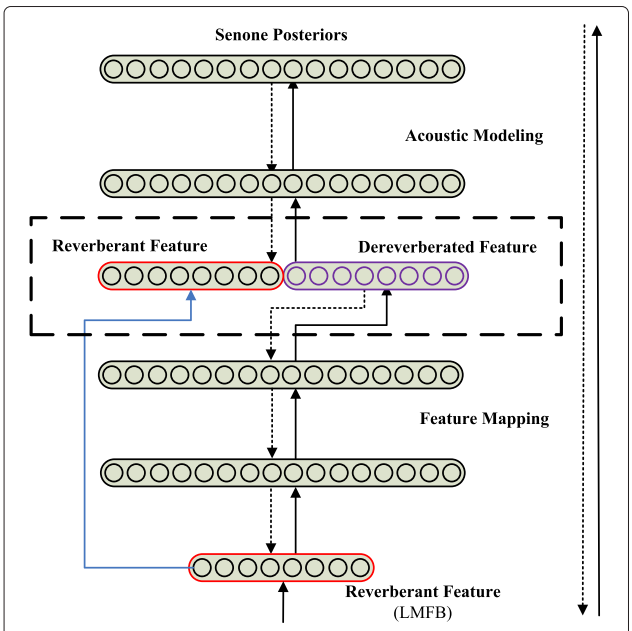

Fig. 4 New joint training structure (DNN-JT2)

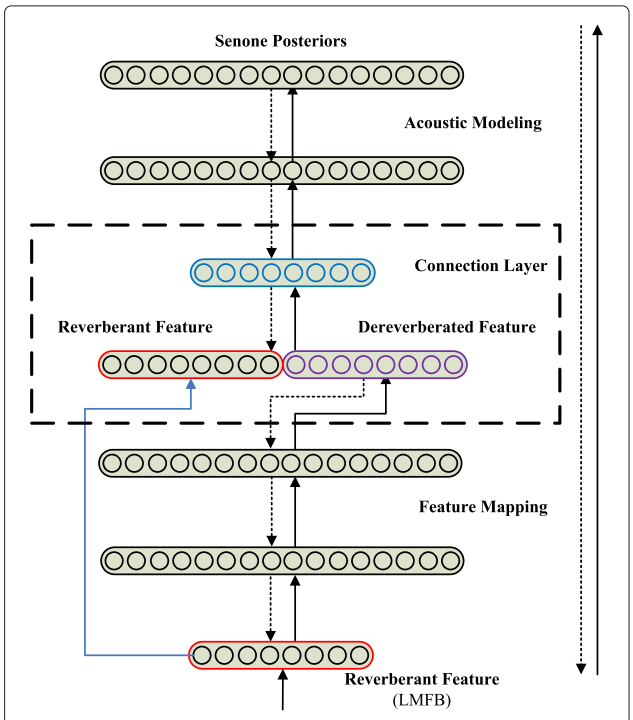

Fig. 5 New joint training structure with a connection layer (DNN-JT3, DNN-JT4)

Algorithm 4 : **Training procedure of new joint training structure with a learned connection layer (DNN-JT3)**

Step1: **Front-end DNN training**

1. Extract reverberant and clean LMFB features from all training utterances.
2. Train feature mapping DNN with reverberant-clean feature pairs under MMSE criterion.

Step2: **Back-end DNN training**

1. Train a baseline DNN acoustic model using reverberant acoustic features based on randomly initialized weights.
2. Use a randomly initialized linear connection layer to concatenate the dual output (reverberant and dereverberated) front-end and the baseline DNN acoustic model.
3. Fix the front-end layers to retrain the connection layer and back-end acoustic model layers. We denote this intermediate model as DNN-FM3.

Step3: **Joint training of front-end and back-end DNNs**

1. Optimize the front-end, connection layer and back-end as a whole network under CE criterion with reverberant features. We denote the hybrid model as DNN-JT3.

Algorithm 5 : **Training procedure of new joint training structure with a fixed connection layer (DNN-JT4)**

Step1: **Front-end DNN training**

1. Extract reverberant and clean LMFB features from all training utterances.
2. Train feature mapping DNN with reverberant-clean feature pairs under MMSE criterion.

Step2: **Back-end DNN training**

1. Train a baseline DNN acoustic model using reverberant acoustic features based on randomly initialized weights.
2. Use matrix **A** to concatenate the dual output (reverberant and dereverberated) front-end and the baseline DNN acoustic model.
3. Fix the front-end layers and the connection layer to retrain the back-end acoustic model layers. We denote this intermediate model as DNN-FM4.

Step3: **Joint training of front-end and back-end DNNs**

1. Optimize the front-end and back-end layers as a whole network under CE criterion with reverberant features. The connection layer is still fixed. We denote this hybrid model as DNN-JT4.

$$\mathbf{A} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1/2 \\ 1/2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \tag{3}$$

5 Experiments

Our experiments are conducted on a Mandarin speech recognition task with a vocabulary of more than 80,000 words. First, two kinds of RIRs were generated, namely real and synthetic RIRs. The real RIRs were measured from nine rooms with different volumes (small, medium, and large) and three types of distances between the speaker and the microphone array (1, 3, and 5 m). The reverberation times (T60) of the small-, medium-, and large-size rooms are about 0.29 s, 0.60∼0.86 s, 1.07∼1.40 s, respectively.

One hundred fifty synthetic RIRs were created according to the image method [48, 49] with random enclosure properties, microphones and source positions, and source radiation characteristics. Table 1 lists the parameters involved in the simulations with the ranges of the adopted uniform distributions. For a given enclosure, the target T60 is achieved by varying the wall absorption coefficient according to the Sabine's formula. The parameter ρ determines the source directivity from omnidirectional setting ($\rho = 0$) to highly directional setting ($\rho = 6$). The source and the microphone can be located anywhere within the room, while the source orientation is distributed between $-\pi$ and π. More details can be found in [50].

Then, two datasets were adopted for generating multi-style reverberant simulation data. One was 1000-h high-quality clean speech while the other one was 1000-h close-talking speech collected from mobile internet users.

As for the front-end, the frame length was set to 400 samples (or 25 ms) with a frame shift of 160 samples (or 10 ms) for the 16 kHz speech waveforms. The 257-dimensional LPS features were used to train DNN pre-processor. The architecture of DNN-PP was 2827-3072-3072-3072-257 with 11 frames of LPS features for the input layer, 3072 sigmoid nodes for each hidden layer, and one frame of LPS features for the output layer. Other parameter settings can refer to [16].

Table 1 The range of parameters for RIR generation

Parameter	Min	Max
T60	0.1 s	2 s
Room size	3 m	7 m
Room hight	3 m	5 m

The LMFB acoustic features for back-end consisted of 24-dimensional log Mel-filterbank feature plus their first- and second-order derivatives, and 3-dimensional pitch features. The final 75-dimensional LMFB features were adopted for both feature mapping and acoustic modeling DNNs. The architecture of feature mapping DNN was 825-2048-2048-2048-825, which denoted that the size was 825 (75*11, τ = 5) at the input layer, 2048 for three hidden layers with sigmoid nodes, and 825 for the output layer. Other parameter settings can refer to [31].

For acoustic modeling, each triphone was modeled by an HMM with three emitting states. There were totally 9004 tied states. For the Gaussian mixture model (GMM)-based HMM system, 40 Gaussian mixtures were used. For the DNN-HMM system, the input layer was a context window of 11 frames of LMFB features. The DNN for acoustic modeling had 6 hidden layers with 2048 sigmoid nodes in each layer, and the final softmax output layer had 9004 units, corresponding to the tied states of HMMs. The other parameters were set according to [31]. Three-gram language model was used for decoding.

For several joint training configurations, the architectures of DNN-JT1 and DNN-JT2 were 825-2048*3-825-2048*6-9004 and 825-2048*3-1650-2048*6-9004, respectively. DNN-JT3 and DNN-JT4 shared the same architecture 825-2048*3-1650-825-2048*6-9004 with a connection layer.

5.1 Proof-of-concept on simulated test data

For proof-of-concept, a medium-size training set (150 h) of reverberant speech was generated from 150-h high-quality clean speech convolving with real RIRs. Another 645 clean utterances covering 20 males and 17 females were used to construct the test set with real RIRs. The test reverberation conditions were as follows: three rooms (T60 = 0.25, 0.61, 1.10 s) and the 3-m distance between the speakers and the microphones. This test set is referred as SimData.

The WER on SimData for different models including clean-condition training with 150-h clean speech (Clean-Model), multi-condition training with 150-h reverberant speech (Reverb-Model), DNN trained on dereverberated speech (DNN-PP), and conventional joint training model (DNN-JT1) are shown in Table 2. We also reported the intermediate results (DNN-FM1) of DNN-JT1. DNN-FM1 was similar to the feature enhancement methods used in [25, 26, 29], in which the front-end dereverberation and the back-end acoustic modeling via DNNs or RNNs were trained separately. The results show that DNN-PP was effective in clean-condition training. However, the performances of both DNN-PP and DNN-FM1 were unsatisfactory in multi-condition training with high baseline performances, especially for the small T60. Only DNN-JT1 achieved consistent and significant performance gains over all other models on the test sets across all T60 conditions, which demonstrates the effectiveness and importance of joint training.

Table 2 WER (%) comparisons on SimData: Clean-Model and Reverb-Model stand for baseline systems of clean-condition training and multi-condition training, respectively

System	T60 = 0.25 s	T60 = 0.61 s	T60 = 1.10 s
Clean-condition training			
Clean-Model	21.49	50.21	91.69
DNN-PP	14.64	17.04	40.99
Multi-condition training			
Reverb-Model	6.75	7.96	21.10
DNN-PP	7.08	8.49	17.61
DNN-FM1	6.77	7.96	19.79
DNN-JT1	6.06	6.93	16.88

DNN-PP stands for pre-processing, DNN-JT1 for conventional joint training structure, and DNN-FM1 for intermediate result of DNN-JT1

Visually, the spectrograms of an example processed by DNN-PP are presented in Fig. 6, including clean speech, reverberant speech with T60 = 0.61 s and DNN-PP dereverberated speech. We can observe that the corrupted speech was processed effectively and neatly. Figure 7 displays the results of DNN-JT1, including the 24-dimensional static LMFB features of clean speech, reverberant speech with T60 = 0.61 s and DNN-PP dereverberated speech, and the output of front-end regression DNN before joint training (the estimated clean LMFB features), the output of front-end after joint training. We also used MSE (mean squared error) to measure the change of the output of front-end DNN before and after joint training. In Fig. 7, the average MSE between clean reference and feature mapping is 78.4 per frame. After joint training, the MSE is 264.1 per frame. Feature mapping could generate more similar results to the reference clean speech visually and yielded good performance of MSE while DNN-JT1 could achieve better recognition performances due to the further optimization of front-end DNN under the CE criterion designed for speech recognition.

5.2 Experiments on real test data

DNN-JT1 was verified to be effective on SimData compared with signal level dereverberation (DNN-PP) and feature level dereverberation (DNN-FM1). To test the effectiveness of our proposed approaches on real test data, we enlarged the training data. First, a DNN acoustic model (denoted as C-T) was trained using 1000-h close-talking speech. Then two multi-condition DNN models using 2000-h data were generated, denoted as Multi-1 and Multi-2. Multi-1 was trained using 1000-h close-talking plus 1000-h reverberant speech simulated from high-quality clean speech. The only difference of Multi-2

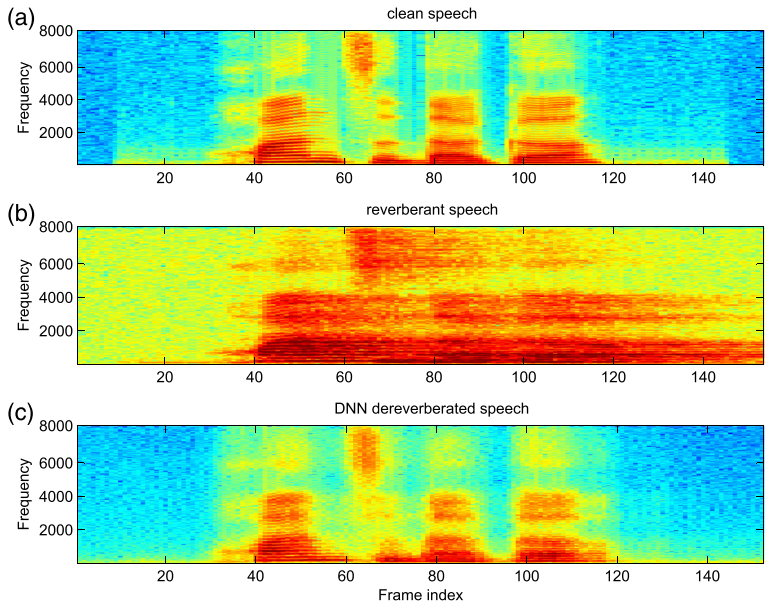

Fig. 6 An example of DNN dereverberation: **a** spectrogram of clean speech; **b** spectrogram of reverberant speech with T60 = 0.61 s, distance = 3 m; **c** spectrogram of DNN-PP dereverberated speech

from Multi-1 is that the 1000-h reverberant speech was simulated from the 1000-h close-talking speech. For simulating the reverberant data, all real and synthetic RIRs were used. We randomly chose 1000 sentences from the training set as our development data.

As for the test set, the real data (denoted as RealData) were collected for both close-talking and distant-talking conditions, which aimed at evaluating the robustness of our proposed approach against variations which cannot be reproducible by simulation data. For close-talking

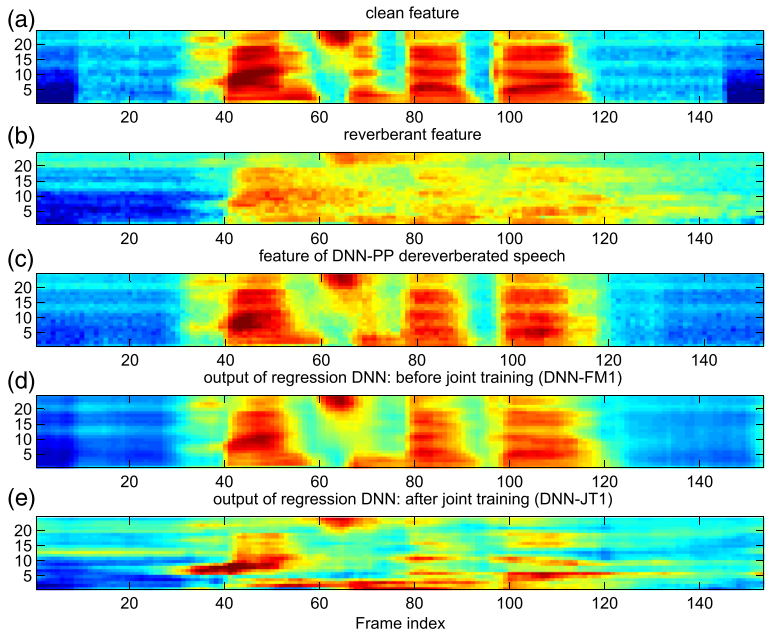

Fig. 7 DNN-JT1 results: **a** static LMFB features of clean speech; **b** static LMFB features of reverberant speech with T60 = 0.61 s, distance = 3 m; **c** static LMFB features of DNN-PP dereverberated speech. **d** The output of front-end regression DNN: before joint training (estimated clean LMFB features). **e** The output of front-end regression DNN: after joint training

conditions, RealData included three subsets, namely Hi-Fi conditions (Clean), common environments (C-E), and with background noises (N-E).

For distant-talking conditions, RealData contained 6 reverberation conditions: 3 rooms (Room1: a living room, Room2: a conference room and Room3: a classroom), 2 types of distances between the speaker and the microphone array (near ∼3 m and far ∼5 m). Fifty speakers (25 males and 25 females) were asked to read their own testing texts in each room, with 30 utterances for near fields and 30 utterances for far fields. All the data were recorded with an eight-channel circular array with diameter of 20 cm similar to REVERB challenge. It should be noted that all the speakers and reading texts in testing set are different from those in the training set.

Table 3 lists the results of different systems on RealData for close-talking conditions. First, both Multi-1 and Multi-2 models yielded performance degradations on Clean and C-E subsets compared with C-T model. But the multi-condition models were more robust to the background noises (N-E). Multi-2 consistently outperformed Multi-1, indicating that the close-talking speech from real users were more effective than the high-quality clean speech in the simulation of reverberant speech. All joint training approaches in the following experiments were implemented on top of Multi-2. DNN-JT4 achieved the best performance in average and consistently outperformed the corresponding multi-condition system (Multi-2) and other joint training approaches across all close-talking conditions. The performance of DNN-JT3 seemed abnormal compared with DNN-JT4. The only difference between these two models is that the connection layer was learned or manually designed. We analyzed the connection layer weights of DNN-JT3 and found large positive values concentrated on the diagonal of the original feature part due to the overfitting. In other words, DNN-JT3 mainly used the original feature, and the processed feature was not well used for acoustic modeling. In Table 3, the performance of DNN-JT3 was very close to that of Multi-2 which also confirmed our analysis. Based on the above analysis, a fixed averaging layer was used to constrain the back-end DNN by making use of the complementarity between original and enhanced features. The results demonstrate that the fixed connection layer was effective.

Table 4 gives a performance comparison of different single-channel systems on RealData for distant-talking conditions. For single-channel systems, one main channel data were selected from the microphone array. Several observations could be made. First, Multi-2 achieved a relative WER reduction of 7.3 % in average over Multi-1 in the distant-talking conditions, which was more significant than that in close-talking conditions. This is a good news as it is not necessary to collect a huge amount of high-quality clean speech in real practice. We can use easy-collected close-talking speech to construct massive training data from real users. Second, DNN-JT1 could bring consistent improvements over Multi-2 on all testing conditions. DNN-JT2 was better than DNN-JT1 for distant-talking conditions, which was opposite for close-talking conditions in Table 3. This might be due to that there was a strong complementarity between the original and processed features in distant-talking conditions. For example, in some cases, the severe speech distortions or lost speech information in dereverberated feature could be recovered by original feature. Similar to the close-talking conditions, DNN-JT3 did not work well and the performance was very close to Multi-2. However, DNN-JT4 with the manually designed fixed connection layer achieved the best performance in average, yielding overall relative WER reductions of 9.7 % over the baseline system and 3.3 % over DNN-JT1 for distant-talking conditions. The improvements indicate the manually designed fixed connection layer could well leverage on both original and dereverberated features effectively. Compared with DNN-JT2, DNN-JT4 yielded a comparable result with smaller parameters of DNN due to the existence of a connection layer.

Table 3 WER (%) comparisons on RealData for close-talking conditions

System	Clean	C-E	N-E	Avg
C-T	2.92	10.77	17.44	10.38
Multi-1	3.38	11.59	16.78	10.58
Multi-2	3.13	10.97	16.04	10.05
DNN-JT1	3.09	10.96	15.18	9.74
DNN-JT2	3.15	10.97	15.58	9.90
DNN-JT3	3.27	11.24	16.09	10.20
DNN-JT4	3.03	10.93	14.84	9.60

Clean for Hi-Fi environment, C-E for common environment and N-E for noisy environment

Table 4 WER (%) comparisons on RealData for distant-talking conditions with single-channel speech input

System	Room1	Room2	Room3	Avg
Single-channel systems				
Multi-1	18.30	27.07	36.46	27.28
Multi-2	16.56	25.59	33.71	25.29
DNN-JT1	15.43	24.30	31.10	23.61
DNN-JT2	14.74	23.77	30.20	22.90
DNN-JT3	16.50	25.69	33.30	25.16
DNN-JT4	15.04	23.64	29.84	22.84
Multi-2(10HL)	16.90	27.04	34.89	26.28

Room1 is a living room, Room2 is a conference room, and Room3 is a classroom

Table 5 WER (%) comparisons on RealData for distant-talking conditions with multi-channel beamforming front-end

System	Room1	Room2	Room3	Avg
Eight-channel + beamforming systems				
Multi-2	16.73	18.83	22.23	19.26
DNN-JT1	15.43	17.81	20.06	17.77
DNN-JT2	15.57	18.11	19.84	17.84
DNN-JT3	16.31	19.15	21.91	19.12
DNN-JT4	15.18	17.74	19.90	17.61

Room1 is a living room, Room2 is a conference room, and Room3 is a classroom

For a better and fair comparison to further demonstrate the effectiveness of the jointly trained model, a multi-condition model (Multi-2(10HL)) with the same network topology as DNN-JT1 was provided. We can come to a conclusion that the performance gains yielded by the design of front-end explicit dereverberation structure in the joint training framework could not be achieved by simply using more hidden layers in the back-end DNN.

Finally, we also tested our approach on the beamformed speech from the eight-channel WPE dereverberated signals. Table 5 lists the performance comparisons on RealData for distant-talking conditions with multi-channel front-end. The overall performances were consistent with single-channel systems. For multi-channel systems, the acoustic models used in Table 5 are the same as those in Table 4. The acoustic features of a test sample from microphone array processed by WPE and beamforming are presented in Fig. 8. First, we find this linear pre-processing introduced little artefact, which is important to back-end acoustic modeling. Next, the regression DNN trained on the real close-talking data as the learning targets could still enhance the features of beamformed speech effectively. The results show that Multi-2 with multi-channel front-end even significantly outperformed the best DNN-JT4 approach in the single-channel case. Based on this, DNN-JT4 could still yield an overall 8.6 % relative WER reduction which further verified the effectiveness of the new joint training structure when combining with conventional microphone array. It was a good example to take full advantage of both beamforming and deep learning-based dereverberation for distant speech recognition in real practice.

6 Conclusions

In this paper, we explore joint training strategies and propose new hybrid DNN architectures for distant speech recognition. The new DNNs yield significant performance gains over the conventional pre-processing and feature enhancement approaches via DNN-based dereverberation. Furthermore, the jointly trained DNNs are much more robust to real-world testing speech than multi-condition training DNNs for both close-talking and distant-talking conditions. Another interesting observation is that the close-talking speech data collected from real users can be used for both the simulation of

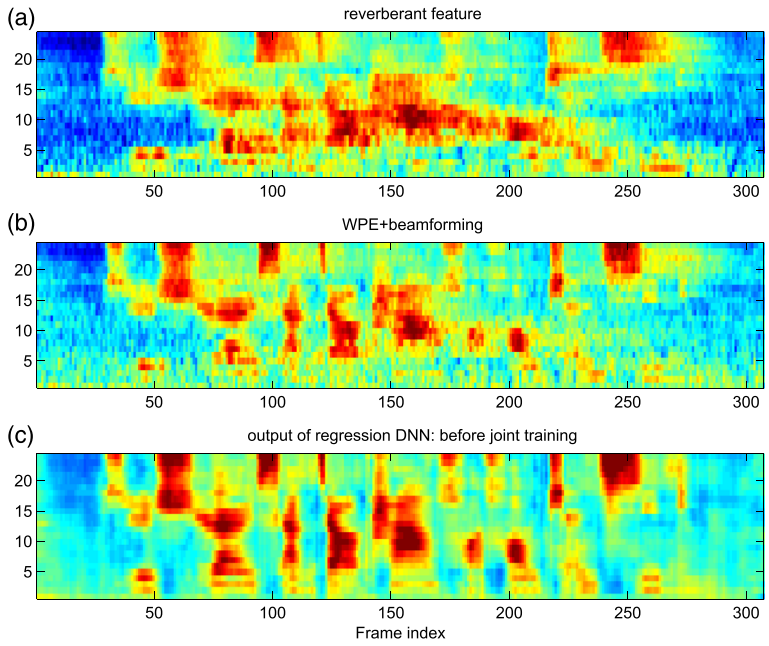

Fig. 8 An example of multi-channel front-end: **a** static LMFB features of one channel reverberant speech; **b** static LMFB features of WPE+Beamforming processed speech; **c** the output of front-end regression DNN: before joint training (estimated clean LMFB features)

reverberant speech and dereverberation as the learning targets in joint training framework, instead of the high-quality clean speech data. Our final experiments on a Mandarin speech recognition task with 2000-h training data show that the proposed framework achieves relative 9.7 and 8.6 % WER reductions over the multi-condition training systems for the cases of single-channel and multi-channel with beamforming, respectively.

In the future, other types of connection layer, fusion of different features in the joint training framework, LSTM architectures which are often more powerful in modeling long-term information will be explored.

Acknowledgements
This work was supported by the National Natural Science Foundation of China under Grants No. 61305002. The authors would also like to thank iFlytek Research for providing the training data, testing data, and computing platform.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, JinZhai Road, Hefei, China. [2]iFlytek Research, iFlytek Co., Ltd., Hefei, China. [3]Georgia Institute of Technology, Atlanta, USA.

Received: 13 March 2016 Accepted: 25 July 2016
Published online: 02 August 2016

References

1. X Huang, A Acero, H-W Hon, R Foreword By-Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. (Prentice Hall PTR, New Jersey, 2001)
2. M Wölfel, J McDonough, *Distant Speech Recognition*. (Wiley, New Jersey, 2009)
3. K Kinoshita, M Delcroix, T Yoshioka, T Nakatani, A Sehr, W Kellermann, R Maas, in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop On*. The reverb challenge: a common evaluation framework for dereverberation and recognition of reverberant speech, (IEEE, 2013), pp. 1–4
4. E Vincent, J Barker, S Watanabe, J Le Roux, F Nesta, M Matassoni, in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On*. The second 'CHiME' speech separation and recognition challenge: datasets, tasks and baselines, (IEEE, 2013), pp. 126–130
5. M Harper, in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop On*. The automatic speech recognition in reverberant environments (ASpIRE) challenge, (IEEE, 2015), pp. 547–554
6. M Brandstein, D Ward, *Microphone Arrays: Signal Processing Techniques and Applications*. (Springer, Berlin, 2001)
7. J McDonough, M Wölfel, in *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*. Distant speech recognition: bridging the gaps, (IEEE, 2008), pp. 108–114
8. ML Seltzer, in *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*. Bridging the gap: towards a unified framework for hands-free speech recognition using microphone arrays, (IEEE, 2008), pp. 104–107
9. M Wolf, C Nadeu, Channel selection measures for multi-microphone speech recognition. Speech Comm. **57**, 170–180 (2014). Elsevier, Amsterdam
10. I Himawan, P Motlicek, S Sridharan, D Dean, D Tjondronegoro, in *INTERSPEECH*. Channel selection in the short-time modulation domain for distant speech recognition, (2015), pp. 741–745
11. TN Sainath, RJ Weiss, KW Wilson, A Narayanan, M Bacchiani, in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference On*. Factored spatial and spectral multichannel raw waveform CLDNNs, (IEEE, 2016), pp. 5075–5079
12. X Xiao, S Watanabe, H Erdogan, L Lu, J Hershey, ML Seltzer, G Chen, Y Zhang, M Mandel, D Yu, in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference On*. Deep beamforming networks for multi-channel speech recognition, (IEEE, 2016), pp. 5745–5749
13. PA Naylor, ND Gaubitch, *Speech Dereverberation*. (Springer, Berlin, 2010)
14. GE Hinton, RR Salakhutdinov, Reducing the dimensionality of data with neural networks. Science. **313**(5786), 504–507 (2006)
15. G Hinton, S Osindero, Y-W Teh, A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
16. Y Xu, J Du, L-R Dai, C-H Lee, An experimental study on speech enhancement based on deep neural networks. IEEE Signal Process. Lett. **21**(1), 65–68 (2014)
17. A Narayanan, DL Wang, Investigation of speech separation as a front-end for noise robust speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(4), 826–835 (2014)
18. Y Xu, J Du, L-R Dai, C-H Lee, A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(1), 7–19 (2015)
19. K Han, Y Wang, DL Wang, WS Woods, I Merks, T Zhang, Learning spectral mapping for speech dereverberation and denoising. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(6), 982–992 (2015)
20. M Karafiát, F Grézl, L Burget, I Szöke, J Černockỳ, in *INTERSPEECH*. Three ways to adapt a CTS recognizer to unseen reverberated speech in BUT system for the ASpIRE challenge, (2015), pp. 2454–2458
21. K Kinoshita, M Delcroix, S Gannot, E Habets, R Haeb-Umbach, W Kellermann, V Leutnant, R Maas, T Nakatani, B Raj, A Sehr, T Yoshioka, A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research. EURASIP J. Adv. Signal Process. **2016**(1), 1–19 (2016)
22. L Couvreur, C Couvreur, C Ris, in *INTERSPEECH*. A corpus-based approach for robust ASR in reverberant environments, (2000), pp. 397–400
23. T Haderlein, E Nöth, W Herbordt, W Kellermann, H Niemann, in *Text, Speech and Dialogue*. Using artificially reverberated training data in distant-talking ASR, (Springer, 2005), pp. 226–233
24. M Ravanelli, M Omologo, in *INTERSPEECH*. Contaminated speech training methods for robust dnn-hmm distant speech recognition, (2015), pp. 756–760
25. X Feng, Y Zhang, J Glass, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition, (IEEE, 2014), pp. 1759–1763
26. M Mimura, S Sakai, T Kawahara, in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference On*. Deep autoencoders augmented with phone-class feature for reverberant speech recognition, (IEEE, 2015), pp. 4365–4369
27. F Weninger, S Watanabe, J Le Roux, JR Hershey, Y Tachioka, J Geiger, B Schuller, G Rigoll, in *REVERB Workshop, Florence, Italy*. The MERL/MELCO/TUM system for the REVERB challenge using deep recurrent neural network feature enhancement, (2014), pp. 1–8
28. F Weninger, S Watanabe, Y Tachioka, B Schuller, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition, (IEEE, 2014), pp. 4623–4627
29. M Mimura, S Sakai, T Kawahara, in *INTERSPEECH*. Speech dereverberation using long short-term memory, (2015), pp. 2435–2439
30. S Hochreiter, J Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
31. T Gao, J Du, L-R Dai, C-H Lee, in *Acoust Speech Signal Process (ICASSP), 2015 IEEE Int Conf*. Joint training of front-end and back-end deep neural networks for robust speech recognition, (2015), pp. 4375–4379
32. A Narayanan, D Wang, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Joint noise adaptive training for robust automatic speech recognition, (IEEE, 2014), pp. 2504–2508
33. Y Xu, J Du, Z Huang, L-R Dai, C-H Lee, in *INTERSPEECH*. Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement, (2015), pp. 1508–1512
34. R Giri, ML Seltzer, J Droppo, D Yu, in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference On*. Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning, (IEEE, 2015), pp. 5014–5018
35. H Kuttruff, *Room Acoustics*. (CRC Press, Florida, 2009)

36. V Tyagi, C Wellekens, in *Acoustics, Speech and Signal Processing (ICASSP), 2005 IEEE International Conference On*. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition, (IEEE, 2005), pp. 529–532
37. P Ghahremani, B BabaAli, D Povey, K Riedhammer, J Trmal, S Khudanpur, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. A pitch extraction algorithm tuned for automatic speech recognition, (IEEE, 2014), pp. 2494–2498
38. T Yoshioka, N Ito, M Delcroix, A Ogawa, K Kinoshita, M Fujimoto, C Yu, WJ Fabian, M Espi, T Higuchi, et al, in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop On*. The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices, (IEEE, 2015), pp. 436–443
39. M Delcroix, T Yoshioka, A Ogawa, Y Kubo, M Fujimoto, N Ito, K Kinoshita, M Espi, T Hori, T Nakatani, A Nakamura, in *Proc. REVERB Challenge Workshop*. Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge, (2014)
40. J Barker, R Marxer, E Vincent, S Watanabe, in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop On*. The third 'CHiME' speech separation and recognition challenge: dataset, task and baselines, (IEEE, 2015), pp. 504–511
41. T Yoshioka, T Nakatani, M Miyoshi, HG Okuno, Blind separation and dereverberation of speech mixtures by joint optimization. IEEE Trans. Audio Speech Lang. Process. **19**(1), 69–84 (2011)
42. J Du, Q Wang, T Gao, Y Xu, L-R Dai, C-H Lee, in *INTERSPEECH*. Robust speech recognition with speech enhanced deep neural networks, (2014), pp. 616–620
43. J Du, Q Wang, Y-H Tu, X Bao, L-R Dai, C-H Lee, in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop On*. An information fusion approach to recognizing microphone array speech in the CHiME-3 challenge based on a deep learning framework, (IEEE, 2015), pp. 430–435
44. Y Tachioka, S Watanabe, in *INTERSPEECH*. Uncertainty training and decoding methods of deep neural networks based on stochastic representation of enhanced features, (2015), pp. 3541–3545
45. Y Ueda, L Wang, A Kai, B Ren, Environment-dependent denoising autoencoder for distant-talking speech recognition. EURASIP J. Adv. Signal Process. **2015**(1), 1–11 (2015)
46. B Ren, L Wang, L Lu, Y Ueda, A Kai, Combination of bottleneck feature extraction and dereverberation for distant-talking speech recognition. Multimed. Tools Appl. **75**(9), 5093–5108 (2016)
47. Y LeCun, BE Boser, JS Denker, D Henderson, RE Howard, WE Hubbard, LD Jackel, in *Advances in Neural Information Processing Systems*. Handwritten digit recognition with a back-propagation network, (Citeseer, 1990), pp. 396–404
48. JB Allen, DA Berkley, Image method for efficiently simulating small-room acoustics. J. Acoust. Soc. Am. **65**(4), 943–950 (1979)
49. P Peterson, Simulating the response of multiple microphones to a single acoustic source in a reverberant room. J. Acoust. Soc. Am. **80**(5), 1527–1529 (1986)
50. M Matassoni, A Brutti, P Svaizer, in *Acoustic Signal Enhancement (IWAENC), 2014 14th International Workshop On*. Acoustic modeling based on early-to-late reverberation ratio for robust ASR, (IEEE, 2014), pp. 263–267