Progressive Joint Modeling in Unsupervised Single-channel Overlapped Speech Recognition

Zhehuai Chen, *Student Member, IEEE,*, Jasha Droppo, *Senior Member, IEEE,*, Jinyu Li, *Member, IEEE,*, Wayne Xiong, *Member, IEEE,*,

Abstract—Unsupervised single-channel overlapped speech recognition is one of the hardest problems in automatic speech recognition (ASR). Permutation invariant training (PIT) is a state of the art model-based approach, which applies a single neural network to solve this single-input, multiple-output modeling problem. We propose to advance the current state of the art by imposing a modular structure on the neural network, applying a progressive pretraining regimen, and improving the objective function with transfer learning and a discriminative training criterion. The modular structure splits the problem into three sub-tasks: frame-wise interpreting, utterance-level speaker tracing, and speech recognition. The pretraining regimen uses these modules to solve progressively harder tasks. Transfer learning leverages parallel clean speech to improve the training targets for the network. Our discriminative training formulation is a modification of standard formulations that also penalizes competing outputs of the system. Experiments are conducted on the artificial overlapped Switchboard and hub5e-swb dataset. The proposed framework achieves over 30% relative improvement of WER over both a strong jointly trained system, PIT for ASR, and a separately optimized system, PIT for speech separation with clean speech ASR model. The improvement comes from better model generalization, training efficiency and the sequence level linguistic knowledge integration.

Index Terms—unsupervised single channel overlapped speech recognition, permutation invariant training, progressive joint training, transfer learning, sequence discriminative training

I. INTRODUCTION

The cocktail party problem [1], [2], referring to multi-talker overlapped speech recognition, is critical to enable automatic speech recognition (ASR) scenarios such as automatic meeting transcription, automatic captioning for audio/video recordings, and multi-party human-machine interactions, where overlapping speech is commonly observed and all streams need to be transcribed. The problem is still one of the hardest problems in ASR, despite encouraging progresses [3], [4], [5], [6].

In this paper, we address the speech recognition problem when multiple people speak at the same time and only a single channel of overlapped speech is available. This is useful when only a single microphone is present, or when microphone

Zhehuai (Tom) Chen is with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, China. He conducted this work while working as an intern at Microsoft Research. (email: chenzhehuai@sjtu.edu.cn)

Jasha Droppo, Jinyu Li and Wayne Xiong are with Microsoft AI and Research, One Microsoft Way, Redmond, WA 98052, U.S.A. (e-mail: {jdroppo,jinyli,wayne.xiong}@microsoft.com)

Corresponding authors are Jasha Droppo and Zhehuai Chen (email: jdroppo@microsoft.com,chenzhehuai@sjtu.edu.cn phone: +1 425 703-7114).

array based algorithms fail to perfectly separate the speech. Specifically, the paper focuses on an unsupervised inference method, which does not need any prior knowledge of speakers. To obtain transcriptions of all speakers from the overlapped speech, joint inference is conducted based on multiple knowledge sources: frequency domain voice discrimination, temporal speaker tracing, linguistic information and speech recognition.

Prior work in unsupervised single-channel overlapped speech recognition generally separates the problem into speech separation and recognition stages. Before the deep learning era, the most popular speech separation technique is computational auditory scene analysis (CASA) [3]. There are two main stages in CASA approaches: segmentation and grouping. The segmentation stage decomposes mixed speech into time-frequency segments assumed to be derived from the corresponding speakers based on perceptual grouping cues [7]. The grouping stage simultaneously and sequentially concatenates the segments to generate independent streams for each speaker. Non-negative matrix factorization (NMF) [8] is another popular technique which aims to learn a set of non-negative bases that can be used to estimate mixing factors during evaluation. Recently, several deep learning based techniques have been proposed but seldom concentrate on the unsupervised case, which is more applicable. In [9], [10], [11], the authors propose deep clustering (DPCL), in which a deep network is trained to produce spectrogram embeddings that are discriminative for partition labels given in training data. The model is optimized so that in the neural network embedding space the time-frequency bins belonging to the same speaker are closer and those of different speakers are farther away. Speech segmentations are therefore implicitly encoded in the embeddings, and can be obtained by clustering algorithm. In [12], a DNN-based gender mixture detection system and three gender-dependent speech separation systems are constructed. The latter ones directly infer the feature streams of two speakers respectively. For all these methods, speech separation and recognition are two separate components and the latter is applied to the separated feature streams. The mismatched feature in the speech recognition stage is one of the limitation in these methods.

In light of permutation invariant training (PIT) proposed in speech separation [13] originally, the PIT-ASR model [14] is the first attempt in joint modeling of unsupervised single-channel mixed speech recognition. Whereas the original PIT technique jointly models the voice discrimination and speaker tracing, PIT-ASR further integrates speech recognition into the

neural network with a unified cross entropy (CE) criterion. Although PIT-ASR shows promising results, it suffers from several disadvantages, which are analyzed in Section II.

In this paper, progressive joint modeling is proposed to divide the single channel overlapped speech recognition problem into three sub-problems for initialization: frame-wise interpreting, speaker tracing and speech recognition (Figure 1). Each module is initialized by placing it into a series of networks that solve progressively more difficult problems. After the initialization, modules are jointly trained with two novel strategies, namely self-transfer learning and multi-output sequence discriminative training. Transfer learning is introduced in this problem, which leverages parallel clean speech to improve the training targets for the network. Our discriminative training formulation is a modification of standard formulations, that also penalizes competing outputs of the system. The proposed framework achieves 30% relative improvement over both a strong jointly trained system, PIT-ASR, and a separately optimized system, PIT for speech separation with clean speech ASR. The improvement comes from better model generalization, training efficiency and the sequence level linguistic knowledge integration.

The rest of the paper is organized as follows. In Section II, the unsupervised single-channel overlapped speech recognition problem is briefly reviewed. In Section III-A, the modular initialization and progressive joint training is proposed. In Section III-B, the self-transfer learning is proposed and in Section III-C, multi-output sequence discriminative training is proposed. In Section IV, the experimental results are reported in artificial overlapped Switchboard corpus and Eval2000 hub5e-swb test set, followed by the conclusion in Section V.

II. UNSUPERVISED SINGLE-CHANNEL OVERLAPPED SPEECH RECOGNITION

Unsupervised single-channel overlapped speech recognition refers to the speech recognition problem when multiple unseen talkers speak at the same time and only a single channel of overlapped speech is available. Different from supervised mode, there's not any prior knowledge of speakers in the evaluation stage.

In the problem, only the linearly overlapped single-channel signal is known, which is defined as,

$$\mathbf{O}_u^{(m)} = \sum_{n=1}^{N} \mathbf{O}_{un}^{(r)} \tag{1}$$

where $\mathbf{O}_{un}^{(r)}$ is the clean signal stream of speaker n at utterance u and $\mathbf{O}_u^{(m)}$ is the overlapped speech stream of utterance u. N is the number of streams. Single channel ASR is always formulated as a supervised sequence labeling problem given by $P(\mathbf{L}_u|\mathbf{O}_u)$, which is the posterior probability of transcription sequence $\mathbf{L}_u$ given the feature sequence $\mathbf{O}_u$ in the utterance u. Nevertheless, the multi-speaker problem is to model the joint distribution of N streams of transcriptions given the overlapped speech signal, $P(\mathbf{L}_{u1}, ..., \mathbf{L}_{uN}|\mathbf{O}_u^{(m)})$. Due to the symmetric labels given the mixture signals, it is no longer a supervised optimization problem. One branch of methods

assumes the inference streams are conditionally independent, and tries to assign the correct transcription stream $\mathbf{L}_{un}^{(r)}$ to the corresponding output stream n,

$$P(\mathbf{L}_{u1}, ..., \mathbf{L}_{uN}|\mathbf{O}_u^{(m)}) \approx \prod_{n=1}^{N} P(\mathbf{L}_{un}^{(r)}|\mathbf{O}_u^{(m)}) \tag{2}$$

Another branch of methods assume the overlapped signal can be separated to $\mathbf{O}_{un}^{(c)} : \mathbf{O}_{un}^{(c)} \approx \mathbf{O}_{un}^{(r)}$. Because the original streams $\mathbf{O}_{un}^{(r)}$ are conditionally independent with each other, the separated signal streams $\mathbf{O}_{un}^{(c)}$ are also assumed to be conditionally independent. Thus Equation (2) can be derived to Equation (3),

$$P(\mathbf{L}_{u1}, ..., \mathbf{L}_{uN}|\mathbf{O}_u^{(m)}) \approx \prod_{n=1}^{N} P(\mathbf{L}_{un}^{(r)}|\mathbf{O}_{un}^{(c)}) \tag{3}$$

However, neither assumption is precise. For the first assumption, there is no pre-determined method to obtain the ideal label arrangements, which is called the speaker tracing problem. The second assumption is that the speech separation and recognition are independent processes, which introduces an artificial information bottleneck.

In [13], the reference streams are treated as an unordered set. The PIT framework is proposed to address the speech separation problem by firstly determining the assignment of the reference stream and inference stream that minimizes the error at the utterance level based on the forward-pass result. This is followed by minimizing the error given the utterance level best assignment. [14] extends this by integrating speech recognition into the neural network with a unified cross-entropy (CE) training criterion.

$$\mathcal{J}_{\text{CE-PIT}} = \sum_{u} \min_{s' \in \mathbf{S}} \sum_{t} \frac{1}{N} \sum_{n \in [1,N]} CE(l_{utn}^{(s')}, l_{utn}^{(r)}) \tag{4}$$

Here, $\mathbf{S}$ is the permutation set of the reference representation and the inference representation. $l_{utn}^{(s')}$ is the n-th inference label of permutation s' at frame t in utterance u and $l_{utn}^{(r)}$ is the corresponding transcription label obtained by clean speech forced-alignment [15].

The PIT-ASR criterion [14] elegantly integrates speech separation, speaker tracing and speech recognition together as Figure 2(a). Its joint modeling approach eliminates the artificial bottleneck between the speech separation and speech recognition tasks. But the method suffers from several disadvantages which deteriorates the performance:

- Previous PIT-ASR work attempts to solve three of the most difficult problems in speech processing with one large model. This oversized model can lead to poor generalization and slow training.
- Speaker tracing and speech recognition are both sequence modeling problems. PIT-ASR concatenates frame level CE as the criteria of the sequence level problem, which limits the modeling effect of the neural network.
- PIT-ASR with a monolithic model is difficult to integrate with other speech separation and segmentation technologies. For instance, although linguistic information can be valuable for solving the speaker tracing problem [6], there is no natural way to incorporate it into PIT-ASR.

Fig. 1. *The Proposed System Framework. The single monolithic structure (the dashed line box) that predicts independent targets for each speaker, proposed in [14], is improved through modularization (three solid line boxes) and pretraining. Self-transfer learning and multi-output sequence discriminative training are conducted on modularly initialized layers.*

III. METHODS

In this work, we propose three separate enhancements to improve the performance of PIT-ASR.

First, the structure and accuracy of the model is improved through modularization and pretraining. Frame-wise interpreting, speaker tracing, and speech recognition modules replace the monolithic structures used in previous work. These modules are progressively pretrained and jointly fine-tuned.

Second, we demonstrate a natural way to incorporate a form of transfer learning. Clean speech features are used to generate soft label targets which are interpolated with the reference label alignments.

Third, multi-output discriminative training is applied to the system. As with single-stream speech recognition, multi-stream discriminative training can help with model generalization. Additionally, the objective function is augmented to reduce cross-speaker word assignment errors.

A. Modularization

In the original formulation, a PIT-ASR model consists of a single monolithic structure that predicts independent targets for each speaker. We improve this by replacing the main network structure with a modular structure, shown in Figure 1.

This modular structure consists of three tasks, namely interpreting mixed acoustic data, tracing speakers across time, and predicting acoustic label sequences. First, the frame-wise module is designed to extract the local time-frequency information necessary to separate the overlapped speech into individual acoustic representations. It is entirely local and does not depend on sequence-level information. Second, the speaker tracing module accepts frame-wise acoustic representations from the frame-wise module and traces the speaker information. This process concatenates adjacent acoustic representations of the same speaker together to infer the recovered speech features of each speaker. Third, the speech recognition modules accept the sequences of recovered acoustic features from each speaker, and produce sequences of label scores suitable for use in an automatic speech recognition system. Because each speech recognition module performs the same task, it is natural to share the parameters of this module across each instance in the final model.

Although it is possible to train the modularized network of Figure 2(e) from random initialization, it is better to use a progressive training strategy. This strategy is motivated by

the Curriculum learning theory in [16], and integrates both modular initialization and joint training. We train a simple model first, and then use it as a pre-trained building block for a more complicated model and task. Figures 2(b)-(e) illustrate how the model becomes progressively more complex while solving more difficult problems, from frame-wise mean squared error to whole utterance cross entropy.

Our simplest model, shown in Figure 2(b), is trained to solve a frame-wise speech separation task. For N speakers, given the mixed data $\mathbf{O}_u^{(m)}$, the model infers an acoustic representation o_{utn} for each speaker n at frame t of utterance u. The objective function of the frame-wise training, is given as

$$\mathcal{J}_{\text{F-PIT}} = \sum_u \sum_t \frac{1}{N} \min_{s' \in \mathbf{S}} \sum_{n \in [1,N]} MSE(o_{utn}^{(s')}, o_{utn}^{(r)}) \quad (5)$$

where, $\mathbf{S}$ is the permutation set of the reference representation and the inference representation. $o_{utn}^{(s')}$ and $o_{utn}^{(r)}$ is the frame level acoustic representation of permutation s' and the reference clean speech, respectively. In each frame t of the utterance u, the overall minimum square error, MSE, is obtained by comparing all the reference and inference representations of each permutation s'.

The architecture for pre-training the speaker tracing module is explained in Figure 2(c). The tracing module is combined with a pre-trained frame-wise module that has had its N output layers removed. As in [13], the PIT objective function is applied in utterance level.

$$\mathcal{J}_{\text{U-PIT}} = \sum_u \min_{s' \in \mathbf{S}} \sum_t \frac{1}{N} \sum_{n \in [1,N]} MSE(o_{utn}^{(s')}, o_{utn}^{(r)}) \quad (6)$$

The speech recognition module is separately pretrained in the same way as a conventional acoustic model, with clean speech and a cross-entropy objective function, maximizing $p(\mathbf{L}_u | \mathbf{O}_u)$. This is illustrated in Figure 2(d).

The final model, shown in Figure 2(e), is created by stacking the speech recognition modules onto the outputs of the pre-trained speaker tracing and frame-wise modules. It is jointly trained with an utterance level PIT-CE objective function given in Equation (4).

Notably, even though the proposed structure has several copies of the speech recognition module, the numbers of parameters between Figure 2(a) and Figure 2(e) are similar. Because the speech recognition modules are solving similar problems, their parameters can be shared [1]. In preliminary

---

[1]Namely the "shared" option in CNTK "CloneFunction".

(a) PIT-ASR Joint Training

(b) Frame-wise Interpreting

(d) Speech Recognition

(c) Speaker Tracing

(e) Final Joint Training
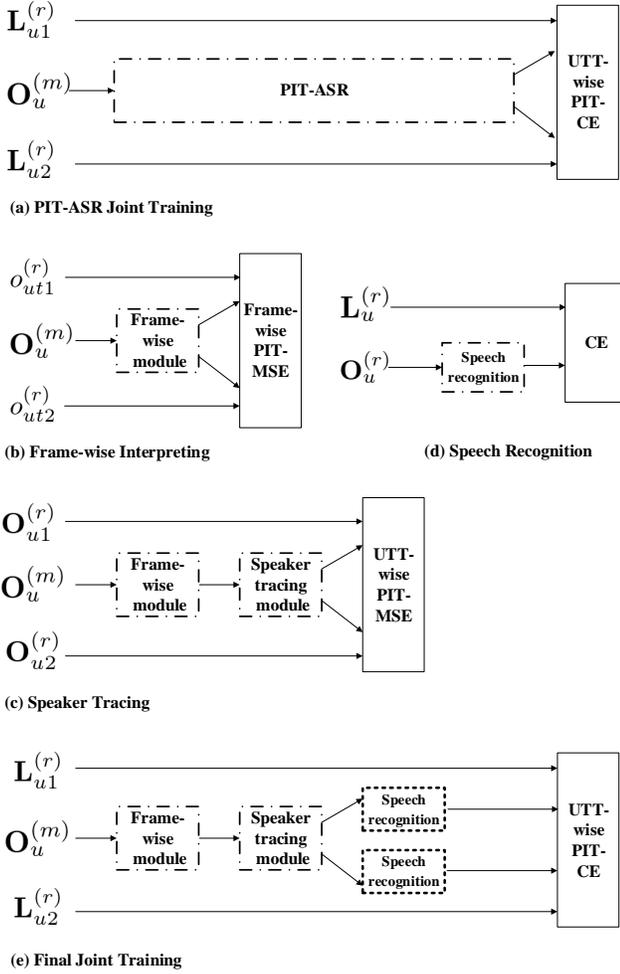
Fig. 2. *PIT-ASR Joint Training versus the Modular Initialization and Progressive Joint Training. The dash-dot blocks indicate the learnable model parameters. The dot-dot blocks indicate the learnable and shared model parameters.*

experiments, the performance gap between sharing and independent parameters is less than 3%. Thus to make the comparison fair, without specific explanation, the shared structure is used.

The advantage of the progressive joint training includes:

- Decreased model complexity leading to better system generalization and data efficiency. By separating system into proper modules, the model complexity is less than the all-in-one system in PIT-ASR. As unsupervised single-channel overlapped speech recognition contains several of the hardest components in speech processing, the model complexity reduction is critical to the empirical training performance.
- Faster convergence of the training process and better quality of the local minimum. Curriculum learning improves both the training speed and the performance of the model [16]. As shown in Figure 4, the progressive joint training needs fewer epochs to converge, and it converges to a better local minimum. An additional benefit is that the module initializations all take much

less time compared with the joint training [2].
- Potential to integrate with other technologies. State-of-the-art technologies in each field can be applied to the initialization of each module.

B. Transfer Learning Based Joint Training

Transfer learning, or teacher-student training, is a powerful technique to address domain adaptation problems in speech recognition. In this section, we show that multi-channel ASR is a type of domain adaptation, and that transfer learning can be used to improve model training.

1) Conventional Transfer Learning in Domain Adaptation: Transfer learning has been proposed to solve the distribution mismatch problem in feature space [17]. To use this method in single-output ASR domain adaptation, parallel data must be available from a source domain and a target domain. A fully trained model in the source domain (the teacher) processes data and generates posterior probabilities, which are sometimes referred to as "soft labels." These soft labels then replace or augment the usual "hard labels" when training the student model with parallel data in the target domain [18].

To train the student, the Kullback-Leibler divergence (KLD) between the output distributions of the teacher and student models is minimized as below.

$$
\begin{aligned}
KLD(y^{(T)}, y^{(S)}) &= \sum_i y_i^{(T)} \log \frac{y_i^{(T)}}{y_i^{(S)}} \\
&= \sum_i [\, y_i^{(T)} \log y_i^{(T)} - y_i^{(T)} \log y_i^{(S)} \,]
\end{aligned}
\tag{7}
$$

$$
= \sum_i -y_i^{(T)} \log y_i^{(S)}
\tag{8}
$$

where $y_i^{(T)}$ and $y_i^{(S)}$ is the teacher and student distributions respectively. Because the first term is not related to the student model optimization, only the second term is used for optimization. Comparing Equation (8) to CE criterion in ASR, the hard labeling is replaced by the soft distribution inferred from the source data by the teacher model.

2) Self-transfer Learning: In light of above discussion, self-transfer learning can be extended to the training of any multi-channel speech recognition system. The student is, of course the multi-channel speech recognition system. It operates in the target domain of mixed speech acoustic data, and must produce separate outputs for each speaker in the mixture. The teacher also must produce separate outputs for each speaker, but has access to the source domain: un-mixed clean speech. The teacher model is a set of clean speech acoustic models operating independently on the separate channels of clean speech.

The self-transfer learning method then minimizes the KLD between the output distribution of the mixed speech model and the set of clean speech models. The KL divergence defined for

---

[2]The training curve of PIT speech separation can be referred to [13]. Besides, initializations also can be done in parallel.

utterance level PIT training between the clean speech model distribution and the joint model distribution is as below,

$$\mathcal{J}_{\text{KLD-PIT}} = \sum_u \min_{s' \in \mathbf{S}} \sum_t \frac{1}{N} \sum_{n \in [1,N]}$$
$$KLD(P(l_{utn}^{(c)}|\mathbf{O}_{un}^{(r)}), P(l_{utn}^{(s')}|\mathbf{O}_u^{(m)})) \tag{9}$$

where the calculation of each $KLD(\cdot)$ pair is the same to the adaptation-purpose single-channel case in Equation (8). Namely, the joint-trained model distribution, $y^{(S)} = P(l_{utn}^{(s')}|\mathbf{O}_u^{(m)})$, is taken as the student model distribution, and the clean speech model distribution, $y^{(T)} = P(l_{utn}^{(c)}|\mathbf{O}_{un}^{(r)})$, is taken as the teacher model distribution. It is notable that when this method is applied to the modular structure proposed in this work, as in Figure 3, the speech recognition modules can be initialized with an exact copy of the teacher model.

The training framework for self-transfer learning is shown in Figure 3. The soft targets generated by the teacher models are interpolated with the hard labeling as in [19]. The training procedure is as below:

1) Clone the speaker tracing layers in the bottom. Clone $2N$ copies of clean ASR model initialized in Section III-A, half for stacking upon the speaker tracing layers, half for model inference given each clean speech stream.
2) Use simultaneous clean speech streams $\mathbf{O}_{un}^{(r)}$ and the overlapped speech stream $\mathbf{O}_u^{(m)}$ to do joint training.
   a) For each mini-batch, do forward propagation of the clean ASR model using each clean speech stream to calculate N streams of $P(l_{utn}^{(c)}|\mathbf{O}_{un}^{(r)})$ respectively. Do forward propagation of the joint model using overlapped speech stream to calculate N streams of inference distributions, $P(l_{utn}^{(s')}|\mathbf{O}_u^{(m)})$.
   b) For that mini-batch, calculate the error signal of Equation (9) and then do back propagation for the joint model.
   c) Update parameters of the joint model and repeat until convergence.

The proposed method elegantly solves the label mismatch problem and helps the model convergence. Namely, using hard labeling obtained from forced-alignment in the clean speech is not proper, because the feature has been distorted in the mixed speech. The proposed method replaces it with the soft distribution. In addition, the proposed method formulates the joint training of multi-channel ASR by domain adaptation between clean speech and overlapped speech. Thus the soft distribution also helps model convergence, because it's easier to recreate its performance, compared with training a speech recognition model from scratch.

The evidence can be observed from the training curve in Figure 4 that the initial CE of self-transfer learning based progressive joint training is much better than that of both joint modeling and progressive joint modeling. Notably, the different starting points between the progressive joint modeling and self-transfer learning based progressive joint modeling is because the CE in the former system is calculated versus hard labeling, while for the latter system it is versus

the soft distribution inferred from simultaneous clean speech [3]. Thus with a better starting point and less parameter updating requirement, finally the model also comes into better minimum in the figure.

The relationships of the proposed method and previous works are summarized as below.

- Model space adaptation. The formulation of the proposed method is similar to KLD-based adaptation [21] and teacher-student based domain adaptation [18]. In [21], to conservatively update model parameters using adaptation data, the target probability distribution is changed from the ground truth alignment to a linear interpolation with the distribution estimated from the unadapted model. In [18], the feature mismatch problem in the target domain is solved by minimizing the inference distribution divergence between the target and source domains using parallel-data. The reader is free to consider the proposed method as analogous to optimizing student network in the target domain, i.e. overlapped speech, to behave similarly to the well-trained teacher network in the source domain, i.e. clean speech, while bearing in mind the proposed method requires module stacking because the motivation is to do joint training.
- Stereo piecewise linear compensation for environment (SPLICE) [22]. The SPLICE algorithm uses stereo data to do noise reduction and channel distortion compensation. In [23], the clean feature is used for the teacher model to provide supervision on the stereo noisy data trained student model. In [19], the multi-channel enhanced feature is used for the teacher model. In this work, the teacher-student framework is also based on stereo data. The student model is initialized better to cope with the more difficult modeling problem, and the entire framework is expanded to handle multiple output streams.
- Progressive stacking transfer learning. [24] proposes to progressively conduct transfer learning to train speech enhancement layers. The motivation of the progressive stacking is only to gradually model a hard task by dividing into several same but smaller tasks. Thus the criteria of all tasks are the same. However, the proposed method is to do joint training of distinct tasks. And each task is fully trained with specific data and criterion.
- Self-supervised training. [25] proposes to use a teacher model based on more accurate sensor information as the supervision of the student model. The motivation of the proposed method is different, which is to transfer distribution between two models with feature mismatch. Besides, the empirical procedure of the proposed method is to fine-tune the original model in the target feature distribution from supervision of itself in parallel source distribution.

3) Learning from Ensemble: Following the transfer learning diagram, the joint model can also benefit from an ensemble of teachers [26]. Specifically, because the problem includes several sub-problems discussed in Section III-A, different

---

[3]We didn't use CE versus hard labeling for all systems, because CE between inference distribution and the hard labeling is not always a good indicator of the quality of the acoustic model [20], while the proposed curves better show the optimization process.
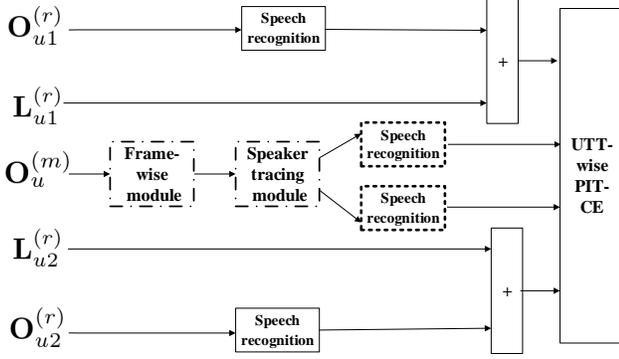
Fig. 3. *Transfer Learning Based Progressive Joint Training. The dash-dot blocks indicate the learnable model parameters. The dot-dot blocks indicate the learnable and shared model parameters.*

neural network structures can show different superiorities, e.g., with different numbers of stream-dependent layers and stream-independent layers. Learning from an ensemble of joint models with different structures is promising in both performance improvement and model compression.

C. Multi-output Sequence Discriminative Training

1) Motivation: Speech recognition is inherently a sequence prediction problem. In single-output ASR, sequence level criteria such as sequence discriminative training tend to improve performance. The unsupervised single-channel overlapped speech recognition problem further includes the speaker tracing problem, which is also a sequence level problem. Previous works concatenate frame level CE as the criteria of the sequence level problem, which limits the modeling effect of the neural network. In this paper, sequence discriminative training of multiple output streams is proposed for the first time.

2) Sequence Discriminative Criterion of Multiple outputs: In single-output ASR, to form a sequence discriminative training criterion, it is necessary to calculate the sequence posterior probability using Bayes' theorem as below,

$$P(\mathbf{L}_u|\mathbf{O}_u) = \frac{p(\mathbf{O}_u|\mathbf{L}_u)P(\mathbf{L}_u)}{p(\mathbf{O}_u)} \quad (10)$$

Here, $\mathbf{L}_u$ is the word sequence of utterance u. $P(\mathbf{L}_u)$ is the language model probability. $p(\mathbf{O}_u|\mathbf{L}_u)$ is the corresponding acoustic part. The marginal probability $p(\mathbf{O}_u)$ of the feature sequence $\mathbf{O}_u$, is modeled by summation of the probability over all possible hypothesis sequences.

$$p(\mathbf{O}_u) = \sum_{\mathbf{L}} p(\mathbf{O}_u, \mathbf{L}) = \sum_{\mathbf{L}} P(\mathbf{L})p(\mathbf{O}_u|\mathbf{L}) \quad (11)$$

Here, $\mathbf{L}$ denotes all competing hypotheses. As an example of the sequence discriminative training criteria, the maximum mutual information (MMI) [27] of inference distribution stream $\mathbf{L}_u$ in utterance u is defined as below,

$$\mathcal{J}_{\text{SEQ}}(\mathbf{L}_u, \mathbf{L}_u^{(r)}) = \log P(\mathbf{L}_u^{(r)}|\mathbf{O}_u) \quad (12)$$

where $\mathbf{L}_u^{(r)}$ is the corresponding reference.

For the overlapped speech recognition problem, the conditional independence assumption in the output label streams is still made as in Equation (2). Then the cross-entropy based PIT can be transformed to sequence discriminative criterion based PIT as below,

$$\mathcal{J}_{\text{SEQ-PIT}} = \sum_u \min_{s' \in \mathbf{S}} \frac{1}{N} \sum_{n \in [1,N]} -\mathcal{J}_{\text{SEQ}}(\mathbf{L}_{un}^{(s')}, \mathbf{L}_{un}^{(r)}) \quad (13)$$

Different from Equation (4), the best permutation is decided by $\mathcal{J}_{\text{SEQ}}(\mathbf{L}_{un}^{(s')}, \mathbf{L}_{un}^{(r)})$, which is the sequence discriminative criterion of taking the s'-th permutation in n-th output inference stream at utterance u. Similar to CE-PIT, $\mathcal{J}_{\text{SEQ}}$ of all the permutations are calculated and the minimum permutation is taken to do the optimization.

The definition of $\mathcal{J}_{\text{SEQ}}(\mathbf{L}_{un}^{(s')}, \mathbf{L}_{un}^{(r)})$ is similar to Equation (12) in single output ASR.

$$\begin{aligned} \mathcal{J}_{\text{MMI}} &= \sum_u \mathcal{J}_{\text{SEQ}}(\mathbf{L}_{un}^{(s')}, \mathbf{L}_{un}^{(r)}) \\ &= \sum_u \log \frac{\sum_{\mathbf{L}_u} p(\mathbf{O}_u|\mathbf{L}_u)^\kappa P(\mathbf{L}_u)}{\sum_{\mathbf{L}} p(\mathbf{O}_u|\mathbf{L})^\kappa P(\mathbf{L})} \end{aligned} \quad (14)$$

For simplicity, $\mathbf{L}_u = \mathbf{L}_{un}^{(r)}$ and $\mathbf{L}$ is all the hypothesis sequences generated by the output stream $\mathbf{L}_{un}^{(s')}$. $p(\mathbf{O}_u|\mathbf{L}_u)$ and $p(\mathbf{O}_u|\mathbf{L})$ is the conditional likelihood obtained from forward propagation of the joint model, $P(\mathbf{L}_u)$ and $P(\mathbf{L})$ is the prior probability obtained from language model.

3) Competing Hypothesis Modeling: The hypothesis sequence $\mathbf{L}$ refers to all the competing hypotheses in the speech recognition. Bad modeling of $\mathbf{L}$, namely ignoring some modeling errors, results in imprecise estimation of Equation (11), which hurts ASR performance. Thus competing hypotheses modeling is key to the discriminative training. Empirically, it is constrained by linguistic search space. In the single output ASR, the linguistic search space is further pruned by the online method, lattice-based discriminative training [28], or by the offline method, lattice-free (LF) discriminative training [29], [30], [31].

Compared with single output ASR, the additional error hypothesis types include: i) Assignment errors: someone said a word, but it is assigned to the wrong channel. ii) cross talk errors: one person said a word, but it appears in multiple channels. They both come from imperfect acoustic modeling and result in several challenges in discriminative training.

The first problem is linguistic search space modeling. As discussed in the first paragraph, there are mainly two branches of methods. When training the joint model, speaker tracing results can always change, which results in different permutations of the same utterance between different epochs. Thus if using lattice-based method, lattices should be updated after each epoch in case of bias in the search space modeling. Another choice is to use a pre-pruned senone level language model as the common search space for all utterances [31]. With this method, the lattice generation problem in multiple outputs can be solved and the discriminative training can be conducted efficiently in the shared search space.

The second problem is the swapped word modeling in multiple outputs. Swapped word results in both cross talk errors and assignment errors. Thus bad swapped word modeling hurts the ASR performance. Generally, the linguistic search space is estimated from the transcription of the training dataset. And then sequence criterion is calculated in this search space. Because there's no swapped word phenomenon in the transcription, thus the search space doesn't contain swapped word cases, which results in overestimating the sequence criterion. Especially for the multiple output streams, the swapped word errors are critical to the ASR performance. Three methods are proposed to cope with the problem.

- Artificial swapped words. A very simple method is to generate several copies of the transcription with artificially swapped words in each copy. And then the language model is estimated on the re-generated transcription. With this method, some of the swapped word cases can still exist in the search space. Thus the problem is alleviated. Empirically, the senone level language model is obtained from the senone level transcription, namely clustered tri-phone state alignment. In case of significant increasing in the search space because of the swapped word, we set a rule that in each frame, the probability of senone swapping is α. But if the senone is swapped, the senone sequence of the following β frames won't be swapped. And γ copies of the transcriptions are generated.
- De-correlated lattice free MMI (LF-DC-MMI). The motivation is that swapped words come from the other output streams. Thus adding these output streams into the search space and minimizing them in the denominator of discriminative training can alleviate the problem.

$$
\mathcal{J}_{\text{LF-DC-MMI}} = \sum_u \log\left[\frac{\sum_{\mathbf{L}_u} p(\mathbf{O}_u|\mathbf{L}_u)^\kappa P(\mathbf{L}_u)}{\left(\sum_{\mathbf{L}} p(\mathbf{O}_u|\mathbf{L})^\kappa P(\mathbf{L})\right)^{1-\lambda}} \cdot \frac{1}{\left(\sum_{\mathbf{L}_{\hat{u}}} p(\mathbf{O}_u|\mathbf{L}_{\hat{u}})^\kappa P(\mathbf{L}_{\hat{u}})\right)^\lambda}\right] \quad (15)
$$

In Equation (15), the other output streams are denoted as $\mathbf{L}_{\hat{u}}$. An interpolation weight λ is added with the augmented term in the denominator.

- De-correlated lattice free boosted MMI (LF-DC-bMMI). Analogous to boosted MMI [32] as Equation (16),

$$
\mathcal{J}_{\text{LF-bMMI}} = \sum_u \log \frac{\sum_{\mathbf{L}_u} p(\mathbf{O}_u|\mathbf{L}_u)^\kappa P(\mathbf{L}_u)}{\sum_{\mathbf{L}} p(\mathbf{O}_u|\mathbf{L})^\kappa P(\mathbf{L}) e^{-b \ \max_{\mathbf{L}_u} A(\mathbf{L},\mathbf{L}_u)}} \quad (16)
$$

we propose de-correlated lattice free boosted MMI (LF-DC-bMMI) as Equation (17). Here, b is the boosting factor. $A(\mathbf{L}, \mathbf{L}_u)$ is the state level accuracy between sequence $\mathbf{L}$ and $\mathbf{L}_u$. By this method, the ASR error hypotheses can be further minimized in the denominator. In the proposed method, both the ASR errors between the target inference sequence and the target reference, and the falsely-recognition of the

interfere streams, are boosted.

$$
\mathcal{J}_{\text{LF-DC-bMMI}} = \sum_u \log \Big[\sum_{\mathbf{L}_u} p(\mathbf{O}_u|\mathbf{L}_u)^\kappa P(\mathbf{L}_u) \cdot \frac{1}{\sum_{\mathbf{L}} p(\mathbf{O}_u|\mathbf{L})^\kappa P(\mathbf{L}) e^{-b \ \max_{\mathbf{L}_u} A(\mathbf{L},\mathbf{L}_u) - \hat{b} \ \max_{\mathbf{L}_{\hat{u}}} (1 - A(\mathbf{L},\mathbf{L}_{\hat{u}}))}} \Big] \quad (17)
$$

where $\hat{b}$ is the de-correlated boosting factor and $A(\mathbf{L}, \mathbf{L}_{\hat{u}})$ measures how many falsely recognitions of the interfere streams.

Experiments are conducted on all three methods in Section IV-D and the first method can be further combined with the other two.

IV. EXPERIMENT

The experimental results are reported in artificial overlapped Switchboard corpus and Eval2000 hub5e-swb test set. Although the methods presented here are valid for any number of overlapped speakers, we focus on the two-talker scenario.

A. Experimental Setup

For training, the Switchboard corpus [33] is used, which contains about 300 hours of speech. Evaluation is carried out on the Switchboard (SWB) subset of the NIST 2000 CTS (hub5e-swb) test set. The waveforms were segmented according to the NIST partitioned evaluation map (PEM) file.

Two-talker overlapped speech is artificially generated by mixing these waveform segments. To maximize the speech overlap, we developed a procedure to mix similarly sized segments at around 0dB. First, we sort the speech segments by length. Then, we take segments in pairs, zero-padding the shorter segment so both have the same length. These pairs are then mixed together to create the overlapped speech data. The overlapping procedure is similar to [14] except that we make no modification to the signal levels before mixing [4]. After overlapping, there's 150 hours data in the training, called *150 hours dataset*, and 915 utterances in the test set. After decoding, there are 1830 utterances for evaluation, and the shortest utterance in the hub5e-swb dataset is discarded. Additionally, we define a small training set, the *50 hours dataset*, as a random 50 hour subset of the *150 hours dataset*. Results are reported using both datasets.

In the training stage, 80-dimensional log-filterbank features were extracted every 10 milliseconds, using a 25-millisecond analysis window. The convolution neural network (CNN) models use 41 context frames (20 in both left and right) and the long short term memory networks (LSTM) processed one frame of input at a time. All neural networks were trained with the Microsoft Cognitive Toolkit (CNTK) [34]. The detailed setup of CNN is listed in Section IV-B. The acoustic model is based on three state left-to-right triphone models with 9000 tied states (senones). The individual senone alignments for the two-talkers in each mixed speech utterance are from the single-speaker ASR alignment [31]. For compatibility, the alignment of the shorter utterance within the mixed speech is padded

---

[4]We will try to release scripts that create the training data in the future.

with the silence state at the front and the end. The clean speech recognition performance in the corpus can be referred to [31], [30]. Using clean speech model to do decoding in the overlapped speech isn't reported as it's as bad as in [14]. The baseline model of joint training is a PIT-ASR model with a setup similar to [14]. The PIT-ASR model is composed of 10 bidirectional LSTM layers with 768 memory cells in each layer [5], and 80-dimensional feature. The baseline model of separately optimized system is a PIT for speech separation (PIT-SS) model combined with a clean speech ASR model. As PIT-SS model has shown competitive performance compared with other speech separation systems, only PIT-SS model is taken as the baseline. The PIT-SS model has a setup similar to [13] but with 6 bidirectional LSTM layers with 768 memory cells in each layer, it directly outputs multiple channels of the 80 dimensional log-filterbank features the speech recognition module expects. The speech recognition module, pretrained as a clean speech model, is composed of 4 bidirectional LSTM layers with 768 memory cells in each layer. It is trained from the corresponding source speech segments used to create the overlapped corpus. After initialization, the WER performance of the ASR model in the clean speech test set is 17.0%. Although it would be easy to incorporate a stronger acoustic model [31] in conjunction with the proposed method, we chose a structure that allows for a fair comparison, in terms of the number of model parameters, among the baselines and proposed methods. However, as discussed in Section III-A, the modular system needs fewer parameters and training iterations to achieve good performance. Notably, the use of enhanced signals after speech separation as training data of the speech recognition module tends to degrade the ASR performance and isn't included. The reason can be from the sporadic distortions that signal processing inevitably adds, similar to what has been observed in [19].

In the evaluation stage, a 30k-vocabulary language model derived from the most common words in the Switchboard and Fisher corpora is used. The decoder uses a statically compiled unigram graph, and dynamically applies the language model score. The unigram graph has about 300k states and 500k arcs [31]. Two outputs of the PIT-ASR model are both used in decoding to obtain the hypotheses for two talkers. For scoring, we evaluated the hypotheses on the pairwise score mode against the two references, and used the assignment with better word error rate (WER) for each utterance [14]. Only the average WER of two output streams is reported, as the task is to correctly recognize all words from both speakers.

B. Separate Optimization v.s. Joint Modeling

Table I shows the performance of the naive joint modeling, PIT-ASR, compared with that of the separately optimized system.

TABLE I
Separate Optimization v.s. Joint Modeling. Fine-tune ST denotes to fine-tune the speaker tracing module and Fine-tune ASR denotes to fine-tune the speech recognition module defined in Figure 2.

Layers	Modular	Fine-tune ST	Fine-tune ASR	WER	Rel. (%)
10 · 0	×	n/a	n/a	57.5	0
6 · 4	×	n/a	n/a	52.8	-8.2
	√	×	×	93.4	+62.4
	√	√	×	51.3	-10.7
	√	√	√	50.2	-12.7

The first row shows the performance of the joint training baseline model in this corpus, PIT-ASR [14]. Compared with the 0dB WER result listed in [14], 55.80%, the performance is reasonable [6]. As discussed in Section III-A, the separately optimized system has a similar number of parameters but different model architecture. Thus to make a fair comparison, the model with 6 stream-independent layers in the bottom and 4 parameter-shared stream-dependent layers in the top, denoted as 6·4, is listed in the second row [7]. The learnable structure is the same to the dash-dot blocks shown in Figure 2(e), but trained from scratch as 10·0. The performance of the 6·4 structure is significantly better than that of the 10·0 structure. The reason is that unlike in the pure speech separation task, the speech recognition stage in this task is also very hard and needs more nonlinear layers. It also shows that this task is much harder than the speech separation, so better joint training method to fulfill the performance of each module is critical to the success.

Without fine-tuning parameters, the performance of the separately optimized system is shown in the third row. The significantly worse performance comes from the feature mismatch in Equation (3). With fine-tuning parameters, the performance is restored in the fourth and fifth rows. The system in the fifth row can be viewed as a strong baseline with separate optimization and fine-tuning.

The better performance of the progressive joint training is from better model generalization and training efficiency [8]. Figure 4 show the effect. Training curves of both joint modeling, i.e. the second row in the table, and progressive joint modeling, i.e. the fifth row in the table, are plotted. From the figure, both better starting point and better converged minimum can be observed in the joint progressive training. With better joint training strategy shown in Section IV-C, such modeling effect can be further fulfilled.

Table II shows the performance of the joint modeling from different modular initialization setups. All modular systems are fine-tuned after joint training. The first and second rows show the naive joint training models with BLSTM and BLSTM combined with CNN, respectively. 6·4 BLSTM refers to 6 layers BLSTM in the bottom and 4 parameter-shared layers

---

[5]In the updated arxiv version of [14], the number of layers changed from 10 to 4 and a moderate WER improvement is obtained. Since the authors didn't provide further information, this work mainly compares with the result of 10 layers. In this work, more layers always bring about better performance. And we believe that improved performance with fewer layers in [14] is a result of bad model generalization in the PIT-ASR model framework.

[6]The differences include: the dataset is 50 hours versus 400 hours, while the clean speech WER is 17.0% versus 26.6% . In Section IV-E, the experiments are extended to 150 hours corpus.

[7]As a comparison, the baseline 10·0 has 10 stream-independent layers in the bottom and no stream-dependent layers except the output layers.

[8]After submitting the paper, we notice that similar progressive joint training idea is proposed in [35]. Experiments using the AMI corpus [36] shows similar improvement.
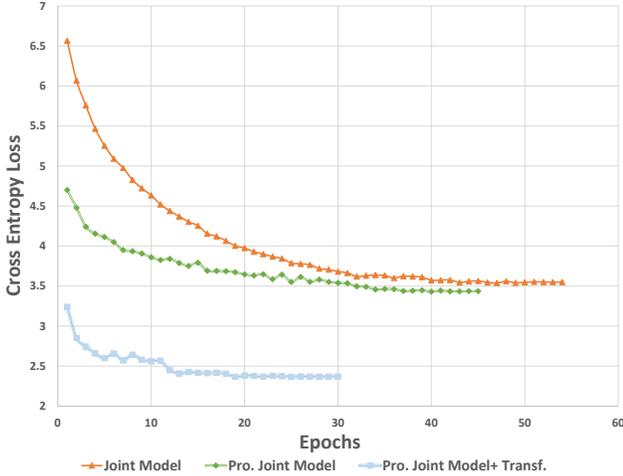
Fig. 4. *Validation Curves of Naive Joint Modeling and the Proposed Methods. Joint modeling, progressive joint modeling and transfer learning based progressive joint modeling are denoted as Joint Model, Pro. Joint Model and Pro. Joint Model + Transf. in the figure. Each epoch contains 24 hours of data.*

for each output in the top, i.e. $6 \cdot 4$ in Table I. Layer-wise context expansion with attention (LACE) model is used for CNN [37], which is a TDNN [38] variant where each higher layer is a weighted sum of nonlinear transformations of a window of lower layer frame. Each LACE block starts with a convolution layer with stride 2 which sub-samples the input and increases the number of channels. This layer is followed by three RELU-convolution layers with jump links. The channel size is 48. The slightly different setup compared with [31] is to make parameter number of one LACE block comparable with one layer of bidirectional LSTM with 768 memory cells, i.e. 5M parameters. Only one block of LACE is used to replace one layer of BLSTM as the frame-wise interpreting module, because the speaker tracing and speech recognition modules are the hardest parts in the problem. The other parts of the structure are the same, and the proposed structure is denoted as 1 LACE + 5·4 BLSTM. From the table, it can be observed that there's no improvement by merely stacking these kinds of neural networks together and jointly training them.

In the third and fourth rows, the model is firstly initialized with frame-wise interpreting, Figure 2(b), speaker tracing, Figure 2(c), and speech recognition, Figure 2(d), tasks respectively and then jointly trained. Comparing the fourth row to the third row, 1 LACE + 5·4 BLSTM shows larger improvement than 6·4 BLSTM. Two conclusions can be derived from the results: i) CNN structure is more suitable for the frame-wise interpreting module because it focuses on the local context and has better modeling power of frequency variations [39]. Meanwhile, LSTM is good at temporal modeling, which is more suitable for the speaker tracing and speech recognition module. The architecture with 1 LACE + 5·4 BLSTM layers combines their strength. Notably, [39] proposes a similar structure, called CLDNN, to form

the acoustic model with modules focusing on different scales and take advantage of the complementarity of CNN, LSTM and DNN. The difference is that, to fulfill the respective advantages in modeling, the proposed method further pretrains each module with different criteria. ii) As the performance improvement from modularization and initialization is much larger in 1 LACE + 5·4 BLSTM, it shows that module-wise initialization is important to fulfill the modeling power of neural networks especially with different structures and scales.

To further analyze the frame-wise interpreting ability of CNN and BLSTM, experiments without frame-wise interpreting initialization are conducted in the fifth and sixth rows. This time, the performances are similar both in 6·4 BLSTM and 1 LACE + 5·4 BLSTM. It shows that in initialization of the speaker tracing module, both BLSTM and CNN can spontaneously learn the frame-wise interpreting ability. We notice that in [13], frame-wise PIT training doesn't show good performance, which is similar to our observation.

TABLE II
Progressive Joint Modeling Based on Modular Initialization. Initializations of frame-wise interpreting, speaker tracing and speech recognition are denoted as FI, ST and ASR respectively.

Modular Init.	Neural network	WER	Rel. (%)
×	6·4 BLSTM	52.8	0
	1 LACE + 5·4 BLSTM	52.9	+0.2
FI+ST+ASR	6·4 BLSTM	50.3	-4.9
	1 LACE + 5·4 BLSTM	47.4	-10.2
ST+ASR	6·4 BLSTM	50.2	-5.0
	1 LACE + 5·4 BLSTM	47.4	-10.2

C. Self-transfer Learning Based Joint Modeling

Table III shows the performance improvement of the transfer learning applied to joint modeling. For transfer learning, the interpolation weight between hard and soft labels is 0.5 .

TABLE III
Transfer Learning Based Joint Modeling

Layers	Modular	teacher	WER	Rel. (%)
10·0	×	×	57.5	0
	×	9·1 ⊕ 6·4 ⊕ 3·7	55.0	-4.4
	×	clean	52.5	-8.7
6·4	×	×	52.8	-8.2
	×	clean	47.1	-18.0
	√	clean	38.9	-32.4
	√	MMI clean	35.8	-37.7

The original PIT-ASR system is in the first row and a better PIT-ASR baseline with 6·4 structure in Table I is also included in the fourth row.

The ensemble-based transfer learning proposed in Section III-B3 is tested in the second row. The ensemble contains 3 types of structure, 9·1, 6·4 and 3·7, where the left number denotes the bottom stream-independent layers and the right number denotes the top stream-dependent layers. The student network learns from each teacher one-by-one. Although it's not a large gain, it shows improvement after learning from each teacher.

The result of replacing hard labeling with simultaneous clean speech based transfer learning is listed in the third and the fifth rows. In both model architectures, transfer learning brings about a relative 10% improvement over the respective baseline, which is comparable with the result in [18]. It shows that soft distribution inferred by the model with similar architecture is superior to the hard labeling.

The self-transfer learning based progressive joint modeling is finally listed in the sixth and the seventh row by using CE-trained and MMI-trained clean speech teacher respectively. Notably, as the model framework discussed in III-B, the initializations of speech recognition modules are the respective clean speech teachers. The result shows over 30% relative improvement. Comparing the third and the fifth rows with Table II, it can be further observed that, combining progressive joint training and self-transfer learning brings about even larger improvement compared with the summation of the relative improvement from each of the two technologies. The learning curve of the proposed method is also shown in Figure 4.

From these results, we conclude: i) The proposed method brings about faster convergence and better converged minimum. The reason is discussed in Section III-B. The better convergence result also comes from the removal of the inappropriate hard alignment in the joint training. ii) Easier convergence helps the model fulfill the best performance in each module. That's the explanation of the even better synergy result compared with the summation of the relative improvements from transfer learning and progressive joint training. iii) Better teacher generates better student. And the MMI-trained distribution can also be transferred to the student model, similar to what has been observed in [40].

Figure 5 further shows the student performance versus quality of the teacher in transfer learning based joint modeling. It can be observed that better student can usually be obtained with better teacher. An important reason is that self-transfer learning is conducted by minimizing the divergence of its own distributions in mixed speech and clean speech. Thus better original distribution, including MMI-trained distribution, can intrinsically be part of the joint model and brings about better results. The only inflection point is in epoch=3 of the MMI teacher, where the student performance is similar to epoch=1 although the teacher model has better WER performance. We believe the reason is that the distribution of the teacher model of epoch=3 is hard to transfer to the student model because of the transition process from CE-trained distribution to MMI-trained distribution.

D. Sequence Discriminative Training

Table IV shows the performance improvement of sequence discriminative training based joint modeling. In this experiment, the baseline is PIT-ASR model without progressive joint training. All the structures are kept the same as 10·0 and only criteria are changed. $\kappa = 0.1$, which is in accordance with the decoding setup. The senone level language model for competing hypothesis modeling is obtained from the clustered tri-phone state alignment. Tri-gram is used, similar to [31].
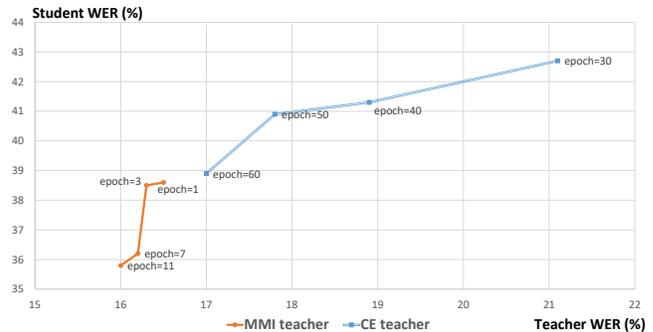
Fig. 5. *Student Performance versus Teacher Quality in Transfer Learning Based Progressive Joint Modeling. All the student models are fully converged based on the respective teacher model. The epoch number denotes how many epochs the teacher model is trained in the clean speech. Each epoch contains 24 hours of data. The MMI teachers are initialized from the best CE teacher, i.e. epoch=60 .*

For the artificial swapped word method, the hyper-parameters in Section III-C3 is decided by the preliminary experiment. $\alpha = 0.4$, $\beta = 10$ and $\gamma = 2$. The swapped word senone level search graph is 4 times larger than the original tri-gram senone level language model graph. $\lambda = 0.1$, $b = 0.1$ and $\hat{b} = 0.2$.

TABLE IV
Sequence Discriminative Training Based Joint Modeling

corpus	criterion	Den. graph	WER	Rel. (%)
	PIT-CE	n/a	57.5	0
	LF-MMI		54.6	-4.9
	LF-DC-MMI	senone transcription	53.2	-7.3
50 hours	LF-bMMI		53.5	-6.9
	LF-DC-bMMI		53.1	-7.4
	LF-MMI	+ art. swapped word	53.5	-6.9
	LF-DC-bMMI		52.7	-8.2
	PIT-CE	n/a	42.2	0
150 hours	LF-MMI	senone transcription	40.1	-4.9
	LF-DC-bMMI		39.1	-7.3

The baseline PIT-ASR system is shown in the first row, denoted as PIT-CE to show the criterion of the system. Applying the naive sequence discriminative training method for multiple outputs in the second row, only brings about 4.9% relative improvement. For the recognition result, word precision is improved but insertion error increases. The reason is from imperfect search space modeling of swapped words discussed in III-C.

By applying the proposed LF-DC-MMI method in the third row, the recognition result is significantly improved compared with both baseline and the naive LF-MMI. The proposed method minimizes the swapped words from parallel output streams in the denominator modeling. Thus the problem can be alleviated.

The fourth and fifth rows show the effect of using bMMI instead of MMI in the formulations. The bMMI criterion boosts the ASR errors in the denominator modeling, implicitly including possibly swapped words. Although significant improvement can be observed between LF-MMI and LF-

bMMI, the LF-DC-bMMI explicitly includes the swapped words and achieves an even better result.

The proposed artificial swapped word method is shown in the sixth and seventh rows. By comparing the sixth row with the second row, and comparing the seventh row with the fifth row, it shows slight but consistent improvement in solving the swapped word problem. And the method can also be combined with LF-DC-bMMI to achieve 8.2% relative improvement versus the CE-trained PIT-ASR baseline.

In the eighth to tenth rows, experiments are conducted on 150 hours corpus. The results are similar, and LF-DC-bMMI criterion shows consistent improvement versus PIT-CE and naive sequence discriminative training criterion. In Section IV-E, it is shown that sequence discriminative training can be combined with other technologies and achieves further consistent and significant improvement.

The discriminative training criterion helps the system training in two ways. Firstly, sequence level criterion helps the sequence level speaker tracing problem in PIT modeling. Specifically, linguistic information is encoded in the senone level language modeling in discriminative training. Thus the procedure implicitly integrates linguistic information in the speaker tracing problem. Secondly, sequence discriminative training improves the speech recognition module. Notably, all the sequence discriminative training procedures are applied after CE initialization as in [31]. With initialization, it also helps Equation (13) to reach a better minimum.

E. Combination and Extension to Larger Corpus

Table V summarizes the performance improvement of integrating all the proposed methods.

TABLE V
Performance Summary in SWBD 50 Hours Dataset

Neural network	Model	WER	Rel. (%)
10·0 BLSTM	PIT-CE	57.5	0
6·4 BLSTM	progressive joint training	50.2	-13
	+ clean teacher	38.9	-32.4
	+ MMI clean teacher	35.8	-37.7
	+ LF-DC-bMMI	35.2	-38.8
1 LACE + 5·4 BLSTM	progressive joint training	47.4	-17.5
	+ clean teacher	36.0	-37.4
	+ MMI clean teacher	34.6	-39.8
	+ LF-DC-bMMI	34.0	-40.9

The PIT-ASR model [14], denoted as PIT-CE, is taken as the baseline of naive joint modeling in the first row. The separately optimized system, namely PIT-SS+ASR, is not included here. As shown in Table II, the performance deteriorates because of feature mismatch. Instead, the proposed progressive joint training model in the second row can be taken as a stronger separately optimized system with fine-tuning. The proposed self-transfer learning based joint training model shows further significant improvement in the third and fourth rows. Finally, the multi-output sequence discriminative training is applied and achieves moderate improvement, although the teacher model is already MMI-trained, similar to what has been observed in [41]. Figure 6 shows decoding examples of

the proposed methods versus the PIT baseline. The baseline contains many errors due to bad model generalization with limited size of dataset. With the proposed methods, errors are significantly reduced. Notably, in this example, the self-transfer learning based progressive joint training mainly reduces errors from similar pronunciations, while sequence discriminative training mainly reduces explicit syntax or linguistic errors, which is in line with the expectation.

With similar number of parameters but different neural networks, namely 1 LACE + 5·4 BLSTM, the system can be consistently improved in the sixth to ninth rows. We believe the further improvement comes from the proper modularization of the problem, which is discussed in Section IV-B.

Table VI expands the dataset to 150 hours to show the effect of more training data.

TABLE VI
Performance Summary in SWBD 150 Hours Dataset

Neural network	Model	WER	Rel. (%)
10·0 BLSTM	PIT-CE	42.2	0
6·4 BLSTM	progressive joint training	41.0	-2.9
	+ clean teacher	32.8	-22.3
	+ LF-DC-bMMI	30.8	-27.0
1 LACE + 5·4 BLSTM	progressive joint training	39.4	-6.6
	+ clean teacher	30.4	-27.9
	+ LF-DC-bMMI	28.0	-33.6

The naive joint training baseline in the first row significantly benefits from more data and shrinks the gap to the proposed progressive joint training model in the second row. However, it still even significantly worse than the self-transfer learning and sequence discriminative training based joint model trained in 50 hours data in Table V. It again shows the disadvantages of large model complexity and insufficient model generalization discussed in Section II. i.e. compared with merely increasing data, the better method to solve the problem is to improve the model generalization. Besides, the convergence speed of naive joint training model in the larger dataset is even slower, namely 4 times more epochs versus the proposed method.

Comparing Table VI with Table V, the proposed self-transfer learning based joint training and multi-output sequence discriminative training show consistent relative improvement versus the progressive joint training. Compared with Table V, sequence discriminative training achieves larger relative improvement on the CE-trained teacher based system.

In both 50 hours and 150 hours corpus, the proposed method achieves over 30% relative improvement respectively, versus the PIT-ASR system and the PIT-SS+ASR system. The improvement comes from better model generalization, training efficiency and the sequence level linguistic knowledge integration.

Although this paper addresses the case of simultaneous speech of two people talking at a relative level of 0dB, we believe it will be straightforward to extend the system to handle more realistic conditions. The case where one speaker is louder than the other has already been observed to be easier for PIT-style models than the 0dB data explored in this paper [14]. For more than two speakers, extension of

```
1  id: (sw_4776_a-006)
2  Labels: <>
3  File: sw_4776
4  Channel: a
5
                                    0 reference
6  REF:  i just ** **** ** **** GO   WALKING AND  I  USUALLY TRY TO go for about an hour (e-) and i HAVE TRIED DOING IT     EVERY DAY BUT i mean some times I do NOT EVEN DO     IT  ONCE DURING THE WEEK  but you know MOST OF THE TIME I TH
7
                                    1 PIT-CE baseline
8  Scores: (#C #S #D #I) 18 22 11 4
9  HYP:  i just SO LONG AS THEY HAVE COME    HERE AS MUCH    AS  WE go for about an hour      and i **** ***** WAS  READING IN   ANY WAY i mean some times * do *** IT   LONGER THAN A    WOMAN  AND STUFF but you know **** ** **** * **
10 Eval:        I  I  I  I  S              S  S  S       S   S  S                           S  S  S                                    D  D  S   S  S       S  S   S  S                  D    D  D  D  D
11
                                    2 + self-transf. progressive joint training
12 Scores: (#C #S #D #I) 34 7 10 1
13 HYP:  i just go OUT YOU    and i usually *** HAVE go for about an hour    *** i **** tried ***** IN every day but i mean some times i do not even do it once during the week but you know **** ** *** **** * *** ONCE WE  TURN DEFINITE
14 Eval:          I  S                   D  S                                     D        D    D  S                                                                                    D  D  D  D   D D  S  S  S  S
15
                                    3 + MMI teacher
16 Scores: (#C #S #D #I) 36 9 6 2
17 HYP:  i just go OUT HERE   and i HAD    LIKE to HAVE go for about an hour   *** i **** ***** AM   TRYING every day but i mean some times i do not even do it once during the week but you know **** ** *** WHAT REALLY TRYING to get
18 Eval:          I  S          S   I                      S                     D        D    S                                                                                       D  D  D  S  S      S
19
                                    4 + seq. disc. tr.
20 Scores: (#C #S #D #I) 37 9 5 1
21 HYP:  i just go OUT HERE   and i usually HAVE to go for about an hour    and i **** tried ***** IN ANOTHER day but i mean some times i do not even do it once during the week but you know **** ** WE  ARE  REALLY TRYING to get *** IT
22 Eval:          I  S                      S                                     D        D    S  S                                                                                   D  D  S  S  S     S          D  S
23 |
24 id: (sw_4854_b-009)
25 Labels: <>
26 File: sw_4854
27 Channel: b
                                    0 reference
28 REF:  well I (don-) i DO NOT CAMP NEAR AS MUCH AS WE USED TO WE    USED TO    go *** TO   THE LAKE all the time (%hesitation) WE (USE-) WE   ALL SKI and everything *** WE USED   TO ALL GO TO   THE LAKE ALL  the TIME AND MY PARENTS H
29
                                    1 PIT-CE baseline
30 Scores: (#C #S #D #I) 20 22 20 2
31 HYP:  well *       i ** *** **** **** ** **** ** ** **** ** THINK AND  HERE'S go FOR ABOUT AN  HOUR all the time              I  WOULD  TRADE OFF NOW and everything BUT I  REALLY DO NOT I  MEAN I  AM   SURE the **** *** ** ****** *
32 Eval:        D        D  D  D   D  D  D D  D D  D D  D  D  D  S  S         I  S  S                                            S  S   S   S                                I  S  S     S  S  S S  S  S  S  S      D  D  D     D
33
                                    2 + self-transf. progressive joint training
34 Scores: (#C #S #D #I) 42 9 11 0
35 HYP:  well i     i ** *** CAMPED near as much as we used to we **** ** ** ** FEEL LIKE all the time            we    we all *** SCAN everything ** **** ** *** THESE DOGS OVER LIKE all the time and my parents had a cabin IN th
36 Eval:        D  S            D    D  D D S  S                      D  D  D D  S  S                                          D  S            D  D  D D  S  S  S                                                        S
37
                                    3 + MMI teacher
38 Scores: (#C #S #D #I) 42 10 10 0
39 HYP:  well *       i ** THINK CAMPED near as much as we used to we **** ** ** *** FEEL LIKE all the time            we    we all SCREAM and everything ** **** ** *** YOU STILL HAVE LIKE all the time and my parents had a cabin IN
40 Eval:        D        D  S  S           D    D  D D S  S                     D  D  D S  S                                     S            D  D  D D  S  S  S                                                        S
41
                                    4 + seq. disc. tr.
42 Scores: (#C #S #D #I) 47 8 7 0
43 HYP:  well *       i ** THINK CAMPED near as much as we used to we used to go to *** lake all the time            we    we all SCREAM and everything ** **** ** *** THESE DOGS BUT LIKE all the time and my parents had a cabin IN
44 Eval:        D        D  S  S                                                     D                                          S            D  D  D D  S  S  S                                                        S
45
```

Fig. 6. *50 hours dataset decoding examples of the proposed methods versus the PIT baseline. The upper part is from stream A and the lower part is from stream B. C, S, D, I refer to correct, substitution, deletion and insertion.*

the proposed system should follow the same construction described in [35]. Finally, we expect robustness to background noise and reverberation to come from standard techniques such as multi-condition training [43].

V. CONCLUSION

In this work, we proposed to divide the single channel overlapped speech recognition problem into three sub-problems: frame-wise interpreting, speaker tracing and speech recognition. Each module is firstly optimized separately with specific designed criteria, which significantly improves the system generalization and training efficiency. After the initialization, modules are jointly trained with two novel strategies: self-transfer learning and multi-output sequence discriminative training. Specifically, in the joint training stage, the clean speech model fine-tunes its parameters with other modules in overlapped speech to fit its own distribution in the simultaneous clean speech. And then sequence discriminative training designed for multiple outputs is applied to integrate linguistic and sequence information. The proposed framework achieves 30% relative improvement over both a strong jointly trained system, PIT-ASR, and a separately optimized system, PIT-SS+ASR.

The proposed framework shows promising perspectives of future improvements, which are: i) Integrating state-of-the-art technologies in each module in the initialization stage, e.g., DPCL [9]. ii) Applying other sequence level criteria to improve the speaker tracing and speech recognition modules, i.e. connectionist temporal classification (CTC) [45]. iii) Explicit integration of language model in the joint modeling, e.g. joint decoding [6] and end-to-end modeling [46].

REFERENCES

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[2] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound.* MIT press, 1994.

[3] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications.* Wiley-IEEE press, 2006.

[4] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Computer Speech & Language*, vol. 24, no. 1, pp. 1–15, 2010.

[5] J. Du, Y. Tu, Y. Xu, L. Dai, and C.-H. Lee, "Speech separation of a target speaker based on deep neural networks," in *Signal Processing (ICSP), 2014 12th International Conference on.* IEEE, 2014, pp. 473–477.

[6] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo, "Deep neural networks for single-channel multi-talker speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 10, pp. 1670–1679, 2015.

[7] M. Wertheimer, "Laws of organization in perceptual forms." 1938.

[8] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Interspeech 2006*, 2006.

[9] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on.* IEEE, 2016, pp. 31–35.

[10] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on.* IEEE, 2017, pp. 246–250.

[11] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[12] Y. Wang, J. Du, L.-R. Dai, and C.-H. Lee, "A gender mixture detection approach to unsupervised single-channel speech separation based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1535–1546, 2017.

[13] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 241–245.

[14] D. Yu, X. Chang, and Y. Qian, "Recognizing multi-talker speech with permutation invariant training," *CoRR*, vol. abs/1704.01985, 2017. [Online]. Available: http://arxiv.org/abs/1704.01985v3

[15] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large vocabulary continuous speech recognition using htk," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. 2. Ieee, 1994, pp. II–125.

[16] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[18] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," in *Interspeech 2017*, 2017.

[19] S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "Student-teacher network learning with enhanced features," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5275–5279.

[20] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[21] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7893–7897.

[22] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large-vocabulary speech recognition under adverse acoustic environments." in *INTERSPEECH*, 2000, pp. 806–809.

[23] K. Markov and T. Matsui, "Robust speech recognition using generalized distillation framework." in *INTERSPEECH*, 2016, pp. 2364–2368.

[24] S. Wang, K. Li, Z. Huang, S. M. Siniscalchi, and C.-H. Lee, "A transfer learning and progressive stacking approach to reducing deep model sizes with an application to speech enhancement," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5575–5579.

[25] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *Journal of Field Robotics*, vol. 26, no. 2, pp. 120–144, 2009.

[26] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in neural information processing systems*, 2014, pp. 2654–2662.

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

[28] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2005.

[29] S. F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at ibm under the darpa ears program," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1596–1608, 2006.

[30] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi." in *INTERSPEECH*, 2016, pp. 2751–2755.

[31] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[32] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4057–4060.

[33] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech,*

[34] F. Seide and A. Agarwal, "Cntk: Microsoft's open-source deep-learning toolkit," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2135–2135.

[35] Y. Qian, X. Chang, and D. Yu, "Single-channel multi-talker speech recognition with permutation invariant training," *arXiv preprint arXiv:1707.06527v1*, 2017.

[36] T. Hain, L. Burget, J. Dines, P. N. Garner, F. Grézl, A. El Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the amida systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

[37] D. Yu, W. Xiong, J. Droppo, A. Stolcke, G. Ye, J. Li, and G. Zweig, "Deep convolutional neural networks with layer-wise context expansion and attention." in *INTERSPEECH*, 2016, pp. 17–21.

[38] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[39] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[40] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[41] L. Lu and S. Renals, "Small-footprint highway deep neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1502–1511, July 2017.

[42] X. Zhang and D. Wang, "Binaural reverberant speech separation based on deep neural networks," *Proc. Interspeech 2017*, pp. 2018–2022, 2017.

[43] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[44] L. Drude and R. Haeb-Umbach, "Tight integration of spatial and spectral features for bss with deep clustering embeddings," *Proc. Interspeech 2017*, pp. 2650–2654, 2017.

[45] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[46] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," *arXiv preprint arXiv:1706.02737*, 2017.

and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on, vol. 1. IEEE, 1992, pp. 517–520.

Zhehuai Chen received his B.S. degree in the Department of Electronic and Information Engineering from Huazhong University of Science and Technology, China, in 2014. He is currently a Ph.D. candidate in Shanghai Jiao Tong University working on speech recognition. His current research interests include speech recognition, speech synthesis and deep learning.

Jasha Droppo (M'03-SM'07) received the B.S. degree in electrical engineering (with honors) from Gonzaga University, Spokane, WA, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1996 and 2000, respectively. At the University of Washington, he helped to develop and promote a discrete theory for time-frequency representations of audio signals, with a focus on speech recognition. He is best known for his research in robust speech recognition, including algorithms for speech signal enhancement, model-based speech feature enhancement, robust speech features, model-based adaptation, and noise tracking. His current interests include the use of neural networks in acoustic modeling and the application of large data and general machine learning algorithms to previously hand-authored speech recognition components.

Jinyu Li (M'08) received the bachelor and master degree from University of Science and Technology of China, in 1997 and 2000, with the highest honor, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, in 2008. From 2000 to 2003, he was a researcher in the Intel China Research Center and Research Manager in iFlytek Speech, China. Currently, he is a principal applied scientist and technical lead in Microsoft Corporation, Redmond, WA. He leads a team to design and improve speech modeling algorithms and technologies that ensure industry state-of-the-art speech recognition accuracy for Microsoft products such as Cortana and xBox Kinect. His major research interests cover several topics in speech recognition, including deep learning, noise robustness, discriminative training, feature extraction, and machine learning methods. He authored more than 70 refereed publications and around 20 patents. He is the leading author of the book Robust Automatic Speech Recognition A Bridge to Practical Applications-Academic Press, Oct, 2015. Currently, he serves as the associate editor of IEEE/ACM Transactions on Audio, Speech and Language Processing.
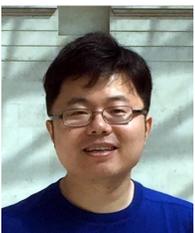
Wayne Xiong (M'07) received the Bachelor degree in electrical engineering from Tsinghua University, Beijing, P. R. China, in 2005, and the Master degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2007. He joined Microsoft Corporation in 2007. He is currently working as a Senior Researcher in the Speech and Dialog Research Group. His research interests include speech recognition and next generation of dialog systems.