A Multiobjective Learning and Ensembling Approach to High-Performance Speech Enhancement With Compact Neural Network Architectures

Qing Wang, Jun Du, Li-Rong Dai, and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—In this study, we propose a novel deepneural network (DNN) architecture for speech enhancement (SE) via a multiobjective learning and ensembling (MOLE) frameworkto achieve a compact and lowlatency design, while maintaining good performance in quality evaluations. MOLE follows the boosting concept when combining weak models into a strong classifier and consists of two compact DNNs. The first, called the multiobjective learning DNN (MOL-DNN), takes multiple features, such as log-power spectra (LPS), mel-frequency cepstral coefficients (MFCCs) and Gammatone frequency cepstral coefficients (GFCCs) to predict a multiobjective set that includes clean speech feature, dynamic noise feature, and ideal ratio mask (IRM). The second, called the multiobjective ensembling DNN (MOE-DNN), takes the learned features from MOL-DNN as inputs and separately predicts clean LPS and IRM, clean MFCC and IRM, and clean GFCC and IRM using three sets of weak regression functions. Finally, a postprocessing operation can be applied to the estimated clean features by leveraging the multiple targets learned from both the MOL-DNN and the MOE-DNN. On speech corrupted by 15 noise types not seen in model training the SE results show that the MOLE approach, which features a small model size and low run-time latency, can achieve consistent improvements over both DNN- and long short-term memory (LSTM)-based techniques in terms of all the objective metrics evaluated in this study for all three cases (the input contexts contain 1-frame, 4-frame and 7-frame instances). The 1-frame MOLE-based SE system outperforms the DNN-based SE system with a 7-frame input expansion at a 3-frame delay and also achieves better performance than the LSTM-based SE system with 4-frame, no delay expansion by including only 3 previous frames, and with 170 times less processing latency.**

***Index Terms*—Speech enhancement (SE), deep neural network (DNN), multiobjective learning, multiobjective ensembling, compact and low-latency design.**

I. Introduction

SINGLE-CHANNEL speech enhancement is a challenging problem in many applications such as automatic speech recognition (ASR), mobile speech communication, and hearing aids [1]. Speech enhancement (SE) [1], [2] aims to improve the quality and intelligibility of a speech signal degraded by noisy adverse conditions. However, the performance of current speech enhancement systems in real acoustic environments is not always satisfactory due to a large variety of unanticipated noise types that make it difficult to characterize noisy speech mathematically. Conventional speech enhancement approaches, such as spectral subtraction [3], Wiener filtering [4], minimum mean squared error (MMSE) estimation [5], [6] and optimally-modified log-spectral amplitude (OM-LSA) speech estimator [7], [8], are considered to be unsupervised techniques and have been studied extensively in the past several decades.

Recently, supervised machine learning has shown some advantages over conventional SE techniques. Xu *et al.* [9], [10] adopted deep neural network (DNN) as a regression model to map the log-power spectra (LPS) features of noisy speech [11] to those of clean speech. A variety of noise types were included in the training stage to achieve a good generalization capability to unseen noise environments. A separate deep auto encoder (SDAE) to estimate the clean speech and noise spectra by minimizing the total reconstruction error of noisy speech spectrum was proposed in [12]. Deep recurrent neural networks (DRNNs) were introduced as a technique for feature enhancement in robust ASR [13], [14]. Long short-term memory (LSTM) recurrent neural networks have also been proven to outperform conventional neural network architectures in speech recognition and speech enhancement tasks due to their ability in modelling long-term acoustic context [15]–[17]. In [18], a DNN was employed to perform binary classification for speech separation. A single DNN to jointly predict the real and imaginary components of the complex ideal ratio mask (cIRM), was adopted in [19], which demonstrated that cIRM outperformed the conventional magnitude-only ideal ratio mask (IRM) [20]. DRNNs were employed in [21] to estimate the spectra of two target sources by integrating the time-frequency (T-F) mask into a loss

Manuscript received August 6, 2017; revised March 4, 2018 and December 23, 2018; accepted March 13, 2018. Date of publication March 23, 2018; date of current version April 24, 2018. This work was supported in part by the National Key R&D Program of China under Grant 2017YFB1002202, in part by the National Natural Science Foundation of China under Grants 61671422 and U1613211, in part by the Key Science and Technology Project of Anhui Province under Grant 17030901005, and in part by the MOE-Microsoft Key Laboratory of University of Science and Technology of China. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong Cui. *(Corresponding author: Jun Du.)*

Q. Wang, J. Du, and L.-R. Dai are with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China (e-mail: xiaosong@mail.ustc.edu.cn; jundu@ustc.edu.cn; lrdai@ustc.edu.cn).

C.-H. Lee is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: chl@ece.gatech.edu).

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

Digital Object Identifier 10.1109/TASLP.2018.2817798

function, yielding performance gains when compared to non-negative matrix factorization (NMF) [22] and the conventional DNN approaches.

Thus far, most deep learning based SE approaches have focused on achieving performance improvements by designing complicated neural network architectures that involve high computational costs. For example, the approaches based on convolutional neural networks (CNNs) [23] and LSTMs [24] require more computations and higher latency than the conventional feedforward and fully connected DNNs due to the use of convolutional/recurrent layers and longer acoustic context information. In real applications, especially in speech communication, a strict restriction of run-time latency exists. Clearly, designing a DNN architecture that simultaneously achieves good performance while maintaining a compact and low-latency real-time performance is a key research topic for deep learning based speech enhancement, and this aspect is the practical focus of our study.

In an earlier study [25], we found that a primary set of clean speech LPS features can be predicted better by incorporating clean speech mel-frequency cepstral coefficients (MFCCs) [26] and clean speech IRMs into the objective function through multi-objective learning. When exploring other auxiliary features, such as Gammatone frequency cepstral coefficients (GFCCs) [27], noise LPS and its corresponding MFCCs and GFCCs, and their IRMs, a preliminary set of experiments found that when these multiple sets of features are known and utilized, the SE performance can be greatly improved. For example, the perceptual evaluation of speech quality (PESQ) score [28] increased from 2.89 for our baseline DNN to 3.81 when the known auxiliary features were used to train a new DNN for SE. We will refer to this set of unavailable, yet critical, features as *oracle* information later in our experiment section.

Motivated by the usefulness of auxiliary features, we propose a compact neural network architecture, called the *multi-objective learning and ensembling* (MOLE), that is a stack of two DNNs. The first, called the *multi-objective learning DNN* (MOL-DNN), takes LPS, MFCC and GFCC features of noisy speech and static noise [9], [10] as input to predict an ensemble of three complementary feature subsets for LPS, MFCC and GFCC, respectively, in which each subset consists of three components, namely, a feature for clean speech and its corresponding IRM, plus that same feature for dynamic noise. Static noise is calculated using the first several frames of an utterance and thus fixed within that utterance. In contrast, the dynamic noise varies in each frame and is estimated from the output of MOL-DNN. The second DNN, called the *multi-objective ensembling DNN* (MOE-DNN), takes the three subsets of predicted multi-objective ensemble from MOL-DNN as input to jointly predict an ensemble of three feature sets for clean speech LPS, MFCC and GFCC, respectively, and their corresponding IRMs. The primary LPS features and the clean speech IRM can be utilized for post-processing, and the auxiliary information from MFCC and GFCC features are used for multi-task learning to improve the predictions of the primary LPS features and their IRMs [25].

Although each feature set learned from the compact multi-tasking MOL-DNN might not be as accurate as the oracle counterpart, the MOE-DNN can be a strong predictor of clean speech because similar to the boosting concept [29], it combines an ensemble of weak regression functions into a strong one. Moreover, MOE-DNN can also be compact because the prediction is relatively easy when given prior awareness of the auxiliary information. To further improve the performance, a post-processing operation is applied that leverages the complementary targets learned from both MOL-DNN and MOE-DNN. The experimental results, on speech corrupted by 15 noise types not used in model training, show that the MOLE approach, at a small model size and low run-time latency, can achieve consistent improvements over both DNN and LSTM based techniques in terms of all the objective metrics evaluated in this study for all three cases: 1-frame, 4-frame and 7-frame input context expansion. The 1-frame MOLE-based SE system outperforms the DNN-based SE system with a 7-frame input expansion at a 3-frame delay. It also achieves better performance than the LSTM-based SE system with 4-frame input expansion by including only 3 previous frames, but 170 times less processing latency.

Compared with other studies, the proposed MOLE approach combines several findings from our previous works [17], [20], [25], [30] to achieve a compact, low-latency, high-performance design for deep learning based speech enhancement. This study is also related to some previous representative works [31]–[35]. Zhang and Wang [31] proposed deep ensemble learning approaches to monaural speech separation, where two multi-context networks were adopted for averaging and stacking to leverage the contextual information in noisy speech. However, this approach requires a longer time delay, a larger model size and more computational complexity compared with the single DNN architecture. In [32], a stacked DNN architecture was used to exploit speech activity information for speech enhancement using left and right acoustic contexts. Nie *et al.* [33] presented a variant of deep stacking networks to model the time correlations for classification-based speech separation. Multi-task learning architectures were adopted by Qian *et al.* for far-field speech recognition in [34], [35]. The concepts underlying the proposed multi-objective learning are similar to those in the multi-factor joint learning proposed in [35], but these two methods have three main differences: 1) the tasks and the corresponding evaluation measures are completely different. This study focused on speech enhancement, while [35] was aimed at far-field speech recognition; 2) the auxiliary information is completely different. In [35], speaker, phone and environmental factor representations were used, while in this paper, clean speech, noise, and IRMs of three feature types were used; 3) to design a compact model, we applied a single DNN to learn multi-objective simultaneously, while [35] used one DNN to extract each factor. Additional information was used in these approaches [31]–[35] for regression or classification tasks, yielding performance gains over a single network. Nevertheless, the practical issues in real applications were not considered in these approaches.

Therefore, in contrast to previous work, the contributions of this study are summarized as follows: (i) practical issues are primarily considered in the design of the proposed MOLE architecture which uses a learning procedure similar to boosting that not only improves the model's speech enhancement

performance but also leads to a more compact design and better convergence (as shown in learning curves); (ii) key information for speech enhancement, including clean speech, noise and IRM, is fully utilized in the proposed framework via multi-task learning, ensemble learning and post-processing; (iii) the complementarity of different feature types, namely, LPS, MFCC and GFCC, is used to further enhance the performance and make the DNN models more compact; and (iv) the long latency problem that results from the frame expansion of the acoustic context, which is crucial in the conventional DNN and LSTM approaches, is partly alleviated in our proposed MOLE approach. This is because the role the multi-objective plays is somewhat similar to the role of frame expansion in achieving predictions of clean speech features.

The rest of the paper is organized as follows. The proposed MOLE framework is described in detail in Section II. We then illustrate the multi-objective concept in Section III. The results of comprehensive speech enhancement experiments are presented in Section IV. Finally, we summarize our finding in Section V.

II. The MOLE Framework

The proposed MOLE framework illustrated in Fig. 1 consists of two stages: MOL and MOE. The MOL stage, shown in the bottom dashed box, uses one compact DNN to learn multi-objective in different feature domains, while the MOE stage, shown in the top dashed box, uses another compact DNN for the ensemble of learned features. Different from the conventional DNN [10], [30] which is used like a black box to predict clean speech, the proposed MOLE framework includes two explicitly designed progressive learning stages, that can potentially improve both the effectiveness and compactness of this new model. The details of the multi-objective concept will be introduced in Section III. In the following subsections, we elaborate on the three main modules of MOLE: multi-objective learning, multi-objective ensembling and post-processing.

A. Multi-Objective Learning (MOL)

As shown in Fig. 1, the MOL stage is designed to learn the multi-objective, including clean speech, dynamic noise and IRM, defined in K ($K = 3$ in this study) acoustic feature domains, namely, LPS, MFCC and GFCC. To learn the multi-objective, one common way is that each target is predicted by a single DNN. Therefore, multiple DNNs are required leading to more parameters. In this study, we use a single multi-tasking DNN to learn all auxiliary targets, which can reduce the parameter redundancy and improve the generalization capability of DNN. Furthermore, this MOL-DNN could be compactly designed for predicting multiple types of weak targets, which are later combined to generate a strong predictor in the MOE stage similar to boosting [29].

As for the input layer of MOL-DNN, both the noisy speech and static noise estimation of all acoustic feature types are concatenated as:

$$\mathbf{v}_t^{\text{MOL}} = [\mathbf{y}_{t-\tau_1,t+\tau_2}^1, \hat{\mathbf{z}}_t^1, ..., \mathbf{y}_{t-\tau_1,t+\tau_2}^K, \hat{\mathbf{z}}_t^K] \tag{1}$$

where $\mathbf{y}_{t-\tau_1,t+\tau_2}^k$ and $\hat{\mathbf{z}}_t^k$ $(k = 1, ..., K)$ denote the noisy speech vector and static noise vector of kth acoustic feature type at tth frame, respectively. Multiple feature streams [36], [37] are concatenated in the MOL-DNN for merging different properties of the input signal. Please note that for the noisy speech vector, frame expansion includes τ_1 left frames and τ_2 right frames, where τ_1 and τ_2 are two important parameters related to the latency in real applications, which will be discussed in the experiments.

The output layer consists of multiple targets to be predicted, including the reference clean speech, dynamic noise, and IRM of all acoustic feature types. Using multi-objective MMSE criterion as the loss function, we adopted stochastic gradient descent (SGD) algorithm to optimize the model parameters with random initialization:

$$\begin{aligned} E^{\text{MOL}} = & \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K} \|\hat{\mathbf{x}}_{t,k}^{\text{MOL}}\left(\mathbf{v}_t^{\text{MOL}}, \mathbf{W}^{\text{MOL}}\right) - \mathbf{x}_{t,k}^{\text{MOL}}\|_2^2 \\ & + \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K} \|\hat{\mathbf{n}}_{t,k}^{\text{MOL}}\left(\mathbf{v}_t^{\text{MOL}}, \mathbf{W}^{\text{MOL}}\right) - \mathbf{n}_{t,k}^{\text{MOL}}\|_2^2 \\ & + \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K} \|\hat{\mathbf{m}}_{t,k}^{\text{MOL}}\left(\mathbf{v}_t^{\text{MOL}}, \mathbf{W}^{\text{MOL}}\right) - \mathbf{m}_{t,k}^{\text{MOL}}\|_2^2 \end{aligned} \tag{2}$$

where E^{MOL} is an equally weighted mean squared error. $\hat{\mathbf{x}}_{t,k}^{\text{MOL}}$, $\hat{\mathbf{n}}_{t,k}^{\text{MOL}}$ and $\hat{\mathbf{m}}_{t,k}^{\text{MOL}}$ are the vectors of clean speech estimation, dynamic noise estimation and IRM estimation for the kth acoustic feature type at the tth frame, respectively. Correspondingly, $\mathbf{x}_{t,k}^{\text{MOL}}$, $\mathbf{n}_{t,k}^{\text{MOL}}$ and $\mathbf{m}_{t,k}^{\text{MOL}}$ are the reference or oracle versions. T is the mini-batch size for the SGD algorithm. $\mathbf{W}^{\text{MOL}}$ denotes the parameters to be optimized in the MOL stage. Similar to [10], the dropout strategy is also applied as a standard configuration to address the overfitting problem.

It should be emphasized that the multi-objective function defined in (2) is quite different from those in our previous work [25] and recent work [20]. First, we predict more auxiliary targets in MFCC and GFCC feature domains. Second, in this study the mean square errors of each target are equally weighted as all the targets learned in the MOL stage are supposed to be equally important which are then fed to the subsequent MOE-DNN. However, in [20], [25], the main purpose of the multi-objective function is to improve the generalization capability of DNN for predicting the clean LPS features, where small scaling factors should be applied for other learning targets to achieve a good performance. The multi-objective function defined in (2) is a robust design by avoiding parameter tuning of the scaling factors for so many learning targets.

B. Multi-Objective Ensembling (MOE)

The design of MOE-DNN is partially motivated by our previous work on dynamic noise aware training [30]. The MOE-DNN can be considered as multi-stream aware training because it uses not only the dynamic noise information in the LPS feature domain but also the IRM information and other feature

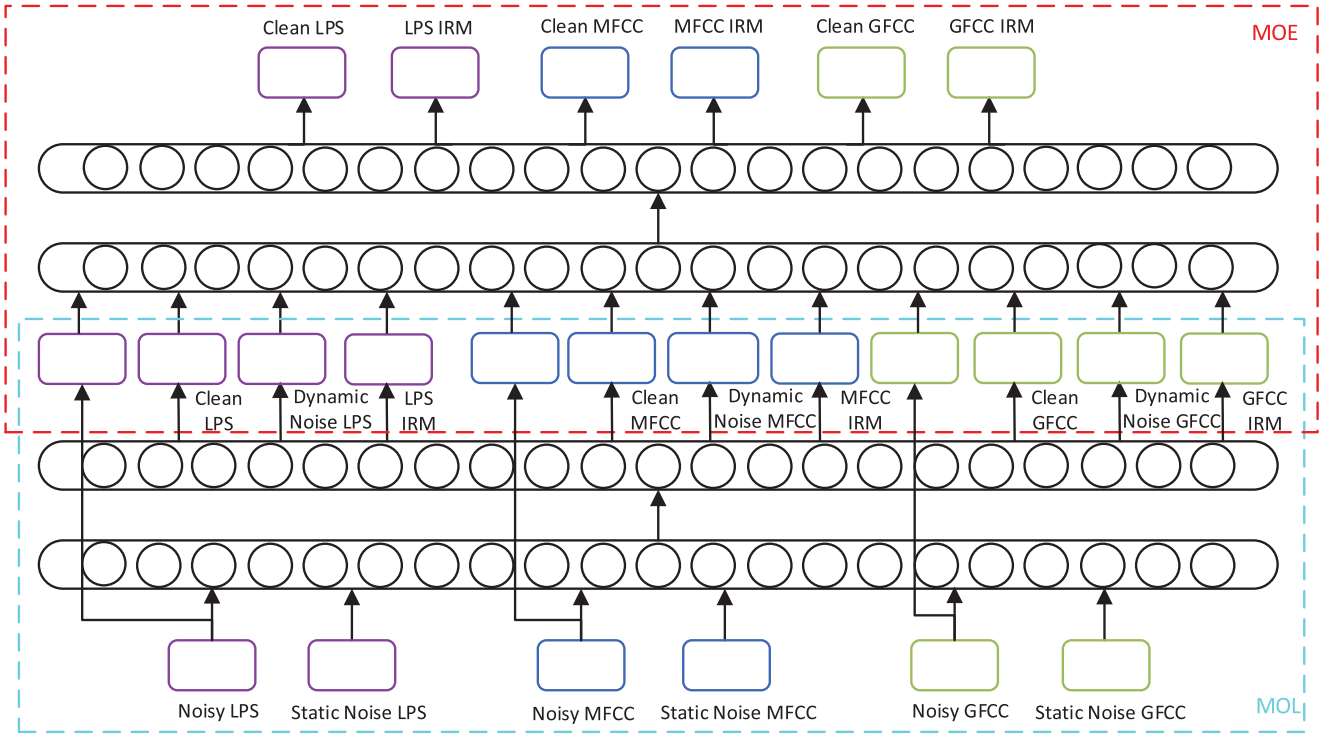

Fig. 1. Illustration of the proposed multi-objective learning and ensemble framework.

types. Based on the multi-stream predicted by the MOL-DNN, the second-stage MOE-DNN forms a strong predictor of clean speech by adopting a concept similar to boosting. Moreover, the use of clean speech estimation, dynamic noise estimation, and the IRM estimation in MOE-DNN for each frame was inspired by traditional speech enhancement methods [7], [8], in which an online tracking procedure is commonly employed by using estimations of clean speech, noise signal, and speech presence probability (similar to IRM) in the current frame to predict those statistics in the next frame. From this perspective, the optimization of MOE-DNN seems like adaptive training with the ensemble of multi-stream.

In Fig. 1, multiple input streams of MOE-DNN $\mathbf{v}_t^{\text{MOE}}$ at the tth frame are formed by concatenating the noisy speech of all feature types and the output of MOL-DNN:

$$\mathbf{v}_t^{\text{MOE}} = \left[\mathbf{y}_t^1, \hat{\mathbf{x}}_{t,1}^{\text{MOL}}, \hat{\mathbf{n}}_{t,1}^{\text{MOL}}, \hat{\mathbf{m}}_{t,1}^{\text{MOL}}, \ldots, \mathbf{y}_t^K, \hat{\mathbf{x}}_{t,K}^{\text{MOL}}, \hat{\mathbf{n}}_{t,K}^{\text{MOL}}, \hat{\mathbf{m}}_{t,K}^{\text{MOL}}\right] \tag{3}$$

where each input stream is adopted only at tth frame, implying that MOE-DNN has no hard latency. The MOE-DNN parameters are randomly initialized, and then optimized by the SGD algorithm using a multi-objective MMSE criterion:

$$E^{\text{MOE}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}\|\hat{\mathbf{x}}_{t,k}^{\text{MOE}}\left(\mathbf{v}_t^{\text{MOE}}, \mathbf{W}^{\text{MOE}}\right) - \mathbf{x}_{t,k}^{\text{MOE}}\|_2^2 + \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}\|\hat{\mathbf{m}}_{t,k}^{\text{MOE}}\left(\mathbf{v}_t^{\text{MOE}}, \mathbf{W}^{\text{MOE}}\right) - \mathbf{m}_{t,k}^{\text{MOE}}\|_2^2 \tag{4}$$

where E^{MOE} is an equally weighted mean squared error, and $\hat{\mathbf{x}}_{t,k}^{\text{MOE}}$ and $\hat{\mathbf{m}}_{t,k}^{\text{MOE}}$ are the clean speech estimation and IRM estimation for the kth acoustic feature type in the tth frame, respectively. Correspondingly, $\mathbf{x}_{t,k}^{\text{MOE}}$ and $\mathbf{m}_{t,k}^{\text{MOE}}$ are the reference versions, and $\mathbf{W}^{\text{MOE}}$ denotes the parameters to be optimized in the MOE stage. Other training details are similar to those of the MOL-DNN.

Although both the MOL-DNN and the MOE-DNN conduct multi-task learning, their motivations are quite different. The MOL-DNN is mainly intended to provide the learned targets to the MOE-DNN. However, the role of the multi-task learning in the MOE-DNN is to improve the framework's generalization capability for predicting the clean LPS features, a task that was inspired by our previous work [17], [25]. In [25], we demonstrated that multi-objective learning using MFCC features as additional constraints can improve model generalization. Here, we extend that concept and use both MFCC and GFCC features in the MOE-DNN. Moreover, our recent study [17] verified that learning the targets of clean speech and IRM could be complementary, an idea that is also incorporated into our MOE-DNN training process.

C. Post-Processing

To alleviate the over-estimation or under-estimation problem of the enhanced spectrum [25] and to fully utilize the complementary targets learned from both the MOL-DNN and MOE-DNN [17], we perform post-processing via a simple averaging operation in the LPS domain:

$$\hat{\mathbf{x}}_{t,1}^{\text{PP}} = \frac{1}{3}\left[\hat{\mathbf{x}}_{t,1}^{\text{MOL}} + \hat{\mathbf{x}}_{t,1}^{\text{MOE}} + \mathbf{y}_t^1 + \log\left(\hat{\mathbf{m}}_{t,1}^{\text{MOE}}\right)\right] \tag{5}$$

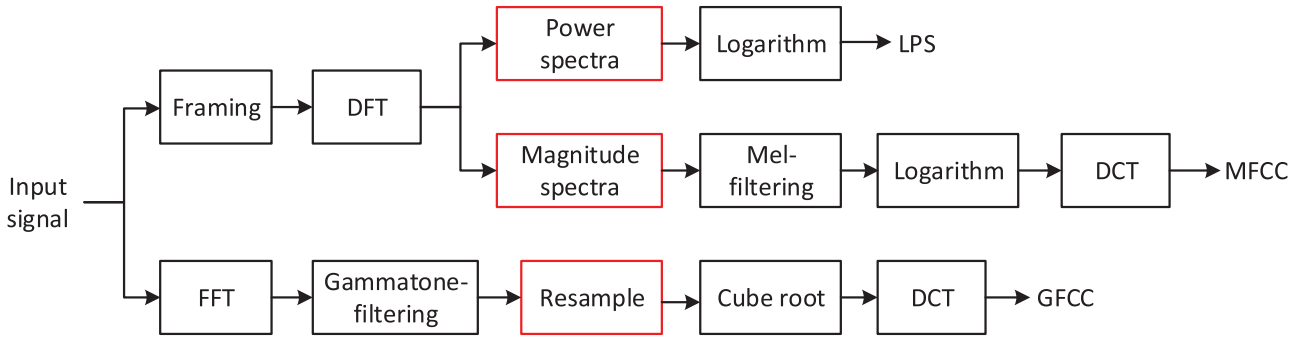

Fig. 2. Illustration and comparison of the feature extraction procedures for LPS, MFCC and GFCC.

where $\hat{\mathbf{x}}_{t,1}^{\mathrm{MOL}}$ and $\hat{\mathbf{x}}_{t,1}^{\mathrm{MOE}}$ are both directly predicted clean LPS features that might be complementary, because MOL-DNN and MOE-DNN are designed with different input features and output targets. Given the noisy LPS input $\mathbf{y}_t^1$ and the estimated IRM $\hat{\mathbf{m}}_{t,1}^{\mathrm{MOE}}$ (which will be defined in (7)), we can see that $\mathbf{y}_t^1 + \log(\hat{\mathbf{m}}_{t,1}^{\mathrm{MOE}})$ are the predicted clean log power spectra features from the IRM estimation in the MOE stage, which has been verified to be complementary with the direct mapping results $\hat{\mathbf{x}}_{t,1}^{\mathrm{MOE}}$ for different SNRs according to [17]. Finally, the post-processing result $\hat{\mathbf{x}}_{t,1}^{\mathrm{PP}}$ is adopted for waveform reconstruction [10].

III. Multi-Objective Features

The previous section provided our motivation for investigating the multi-objective design. In Section IV-A, we will further verify this approach by performing a set of experiments on the oracle information. In this section, we elaborate on the details of the noise and IRM information in different feature domains. Here, the clean speech information refers to clean speech features in different feature domains.

A. Feature Extraction

Fig. 2 illustrates the feature extraction procedures for LPS, MFCC and GFCC. In our previous work [25], we incorporated the MFCC features and demonstrated their complementarity to LPS features in reducing speech distortions. Accordingly, we include MFCC features in the MOLE framework. In contrast to the MFCC extraction procedure using Mel-filters, the GFCC features are extracted using Gammatone filters. Because of the complementarity among these three feature types, we combine them in the MOLE framework. These acoustic features are all known to be related to human auditory perception but have different emphases [38]–[40]. One shared operation for extracting the LPS and MFCC features is discrete Fourier transform (DFT) for T-F analysis. LPS features which have been widely adopted for speech enhancement [9], [10], are related to the human perceptual domain [38], [41] and can be transformed back to the waveform domain without any information loss. MFCC features are popular in both ASR [42] and speaker recognition [43]. High similarity exists between the spacing of Mel-filtering and the human perception scale [39]. When compared to LPS features, MFCC features are concentrated in a low-dimensional space via Mel-filtering and the correlation among different dimensions are removed using a discrete cosine transform (DCT) [44]. We used MFCC features to improve the prediction of clean LPS features in our previous work [25]. GFCC features are obtained using Gammatone filters to extract auditory features based on computational auditory scene analysis (CASA) [45], and they have previously been used in the speech and speaker recognition community [27], [46], [47].

For an input signal with a sampling rate of 16 kHz, LPS and MFCC use a framing operation with a frame length of 512 samples (or 32 ms) and a frame shift of 256 samples. After the DFT operation, a 257-dimensional LPS feature vector is extracted via the logarithm of the power spectra. For the MFCC features, 40 Mel-filters are applied to magnitude spectra, followed by the logarithm operation and DCT. Then, a 41-dimensional MFCC feature vector is formed with 40 static feature dimensions and one energy dimension as in [25]. For the GFCC features, we follow a procedure similar to that proposed in [27], [46], [47]. First, a bank of 64-channel Gammatone filters [45] are applied by decomposing the input signal into the T-F domain. Then, a resampling strategy is adopted to achieve a frame shift of 256 samples, followed by a cubic root operation and DCT. Finally, a 30-dimensional GFCC feature vector is generated as in [46], [47].

B. Noise Information

Noise information has been shown to be effective in improving DNN training for robust ASR [48] and speech enhancement [10], [20], [30]. It also plays an important role in this study. In the MOL stage, input noisy speech features are concatenated with a static noise feature calculated as follows:

$$\hat{\mathbf{z}}_t^k = \frac{1}{T'} \sum_{\tau=1}^{T'} \mathbf{y}_\tau^k \tag{6}$$

where the first T' frames of each utterance are used to estimate the static noise information for each feature type. Obviously, the static noise $\hat{\mathbf{z}}_t^k$ is then a fixed vector within each utterance. For the dynamic noise of each feature type defined in the output layer of the MOL-DNN or at the input layer of the MOE-DNN, reference or oracle noise information is provided as the learning target of the MOL-DNN. In [20], [30], dynamic noise aware training was shown to perform better than static noise aware training in non-stationary environments.

For the static/dynamic noise LPS features, the high dimensionality of the original feature vector is reduced by mapping the 257 linear frequency bins to the 64 frequency bins of Gammatone filter banks. This dimensionality reduction not only improves the enhancement performance but, according to our previous work [20], it also reduces the model size and the computational complexity. For the MFCC and GFCC features of static/dynamic noise, the same dimensions are adopted as in the clean/noisy speech feature vectors.

C. Ideal Ratio Mask Information

IRM is a measure to estimate the speech presence in a local T-F unit. IRM is extended from the ideal binary mask (IBM) [49] widely used in CASA [50]. We adopt the IRM information in the proposed framework because continuous targets yield better performance than do binary targets in speech enhancement [51], [52]. If we assume that clean speech and noise are statistically independent, the IRM of each T-F bin for the kth acoustic feature type is defined following [53]:

$$m_{t,k}(d) = \frac{x_{t,k}(d)}{x_{t,k}(d) + n_{t,k}(d)} \tag{7}$$

where $m_{t,k}(d)$ is the dth element of $\mathbf{m}_{t,k}$, and $x_{t,k}(d)$ and $n_{t,k}(d)$ denote clean speech and noise energy, respectively, in the kth feature domain at time frame t and frequency bin d. As shown in Fig. 2, the clean speech and noise energy in different feature domains are calculated after the operation in the red rectangle boxes. For MFCC and GFCC domains, they are computed before the non-linear transformation, namely, logarithm and cube root operations. Therefore, the dimensions of MFCC IRM and GFCC IRM are 40 and 64, respectively. In the LPS domain, $x_{t,k}(d)$ and $n_{t,k}(d)$ reflect the exact power spectra of clean speech and noise in the linear frequency domain. Then, for the LPS IRM in the output layer of the MOE-DNN, we use a full-band 257-dimension, which is the same as that of the clean LPS feature vector used in post-processing. However, for the MOL-DNN, based on [20], a sub-band 64-dimensional LPS IRM vector is adopted.

Clearly, the IRM information is used in several parts of the MOLE framework, including multi-tasking, adaptive training and post-processing. Because IRM is partially inspired by the human auditory attention mechanism [52], the MOLE architecture with IRM information can be considered as an implicit attention-based neural network design where the IRM functions as an indicator of speech presence or absence. Finally, the dimensionality of multi-objective set in the MOLE architecture for the different acoustic feature types is summarized in Table I.

TABLE I
DIMENSIONALITY OF MULTI-OBJECTIVE SET IN THE MOLE ARCHITECTURE FOR THE DIFFERENT ACOUSTIC FEATURE TYPES

	LPS	MFCC	GFCC
Noisy/Clean Speech	257	41	30
Static/Dynamic Noise	64	41	30
IRM (MOL-DNN)	64	40	64
IRM (MOE-DNN)	257	40	64

TABLE II
THE FIFTEEN UNSEEN NOISE TYPES FROM NOISEX-92 CORPUS

N1: Jet Cockpit 1	N2: Jet Cockpit 2	N3: Destroyer Engine
N4: Destroyer Operations	N5: F-16 Cockpit	N6: Factory 1
N7: Factory 2	N8: HF Channel	N9: Military Vehicle
N10: M109 Tank	N11: Machine Gun	N12: Pink
N13: Volvo	N14: Speech Babble	N15: White

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Here, we extend the speech sampling rate from the 8 kHz used in our previous work [9], [10] to 16 kHz as one commonly used setting. We adopted 115 noise types, including 100 noise types recorded by Hu [54] and some other in-house musical noises to improve the generalization capacity to unseen noise scenarios. The clean speech data was acquired from the TIMIT corpus [55]. All 4620 utterances from the TIMIT training set were corrupted by the abovementioned 115 noise types at six SNR levels (i.e., 20 dB, 15 dB, 10 dB, 5 dB, 0 dB and -5 dB) to build an 80-hour multi-condition training set. The 192 utterances from the core TIMIT test set were used to construct the test set. Because we only conduct an evaluation of mismatched noise types in this study, 15 other unseen noise types from the NOISEX-92 corpus [56] (shown in Table II) were adopted for testing.

The Microsoft Cognitive Toolkit (CNTK) was used for neural network training [57]. In the MOL-DNN, different context window sizes were used for acoustic expansion of the input layer. In the MOE-DNN, we used only the central frame in the input layer. For both the MOL-DNN and MOE-DNN, sigmoid activation function were employed for all the hidden layers. Because the IRM values are in the range [0,1], we also adopted a sigmoid activation function for the IRM targets in the output layer. For the other targets in the output layer, linear units were used. Both MOL-DNN and MOE-DNN use two hidden layers with 1024 units. All the networks were initialized with random weights. The mini-batch size T was set to 256. For fine-tuning, the learning rate of a mini-batch was initially 0.01 and decreased by 10% when the squared loss on the development set increased. The momentum rate was 0.9 and the weight decay coefficient was 0.00001. The dropout rate was set to 0.1 for both the input layer and hidden layers. The total number of epochs was 40. Mean and variance normalization was applied to the input and target feature vectors. To estimate the static noise information, the first $T' = 6$ frames of each utterance were used. The dimensionality of each feature vector is described in Table I.

For comparison purposes, we trained a DNN baseline system according to [10] and used the LSTM-RNN architecture adopted in [17]. For both architectures, we used only the LPS features of noisy speech and clean speech signals to serve as the inputs and the target outputs. The MMSE-based objective function was adopted to train the DNN and LSTM-RNN with random initialization. A dropout strategy [58] was used in the DNN training. In addition, the DNN architecture was fixed at 3 hidden layers, with 2048 units in each hidden layer and a static noise

TABLE III
AVERAGE PERFORMANCE COMPARISONS ON TEST SETS ACROSS 15 UNSEEN NOISE TYPES AT ALL SNR LEVELS WITH VARYING TYPES OF ORACLE INFORMATION

	PESQ	STOI	SSNR	LSD
DNN Baseline	2.89	86.0	4.68	3.21
+ Oracle Noise LPS	3.47	93.2	6.50	2.35
+ Oracle LPS IRM	3.67	94.4	6.70	2.20
+ Oracle (MFCC + GFCC)	3.81	95.5	7.25	1.78

estimation concatenated with the noisy speech input. For the LSTM-RNN, we employed 2 LSTM layers with 1024 cells in each layer. The truncated back propagation through time (BPTT) algorithm [59] was adopted to update the LSTM parameters with 16 frames and 16 utterances processed simultaneously.

We adopted six objective metrics to evaluate the performance of our proposed framework. A perceptual evaluation of speech quality (PESQ) [28] and the short-time objective intelligibility (STOI, in %) [60] were used to assess the quality and intelligibility of enhanced speech. Segmental SNR (SSNR) measures the degree of noise reduction while log-spectral distortion (LSD) is designed as an indicator of the speech distortion [11]. The instrument-measured noise reduction (NR) and segmental speech SNR (SSSNR) proposed in [61] were alternative measures of noise reduction and speech distortion, respectively. For most of the experiments, we provide performance comparisons with the PESQ, STOI, SSNR and LSD measures. In addition, for the overall comparisons among DNN, LSTM and MOLE systems in Section IV-C, we list all six evaluation measures.

In the following, we conduct experiments to evaluate the effectiveness of the proposed framework. First, oracle experiments were conducted to demonstrate the main motivation by assuming that the underlying noise signals and the reference IRM information were known. Then, we show the complementarity of the learnable multi-objective in the MOLE framework and the compactness of the MOLE architecture. Finally, we performed comparisons with the DNN- and LSTM-based approaches to show the advantage of MOLE regarding both performance and practical issues such as model sizes, latency requirements and computational complexity.

A. Oracle Experiments

We designed a set of oracle experiments to demonstrate the main motivation of this study. Suppose that all the targets that MOL-DNN aims to learn are already known (i.e., oracle information) and the corresponding feature vectors are concatenated with noisy LPS features as the input of the DNN in [10]. For example, as Table III shows, the baseline DNN achieves an average PESQ score of 2.89. By adding the oracle noise LPS information, the predictions become better and all the objective metrics improve substantially. For example, the PESQ level rises to 3.47. When the additional oracle LPS IRM information was added, all four evaluation metrics improved consistently, which implies that the noise information and the IRM information are complementary. Moreover, adding oracle information from the MFCC and GFCC feature domains yielded further performance gains, especially for SSNR and LSD, improving the PESQ to 3.81, almost a full point better than that of the baseline DNN. Based on these results, we would expect that the proposed MOLE architecture, using the learned multi-objective set, should boost the performance along a trend similar to that of using the oracle information.

TABLE IV
COMPARISON OF AVERAGE PERFORMANCES ON TEST SETS ACROSS 15 UNSEEN NOISE TYPES WITH ALL SNR LEVELS BETWEEN THE DNN BASELINE AND ORACLE SYSTEMS USING DIFFERENT SETTINGS OF (τ, N_{L}, N_{U}), WHERE τ IS THE NUMBER OF FRAMES IN THE INPUT LAYER AND N_{L} AND N_{U} ARE THE NUMBERS OF HIDDEN LAYERS AND UNITS, RESPECTIVELY

	DNN Baseline				Oracle			
(τ, N_{L}, N_{U})	PESQ	STOI	SSNR	LSD	PESQ	STOI	SSNR	LSD
(1, 3, 2048)	2.76	84.6	4.57	3.59	3.81	95.5	7.18	1.77
(4, 3, 2048)	2.82	85.4	4.67	3.34	3.81	95.5	7.15	1.80
(4, 2, 1024)	2.77	84.2	3.84	3.64	3.79	95.2	6.86	1.90
(7, 3, 2048)	2.89	86.0	4.68	3.21	3.81	95.5	7.25	1.78

TABLE V
SYSTEMS TRAINED WITH DIFFERENT ADDITIONAL INFORMATION

Systems	Additional Information		
DNN [10]	Static noise LPS		
MOLE1	Clean LPS	+ Dynamic noise LPS	+ LPS IRM
MOLE2	+ Clean MFCC	+ Dynamic noise MFCC	+ MFCC IRM
MOLE3	+ Clean GFCC	+ Dynamic noise GFCC	+ GFCC IRM

Table IV show results based on another motivation for this study: the compactness of the MOLE architecture. The DNN baseline system and the best oracle system in Table III were compared with different settings of triplets, namely, acoustic context size, number of hidden layers and number of units in each hidden layer. For the baseline system, longer acoustic contexts could ensure much better performances by making comparisons among the triplet settings (7, 3, 2048), (4, 3, 2048) and (1, 3, 2048). Meanwhile, the larger model size could also yield performance gains by making comparisons between the settings (4, 3, 2048) and (4, 2, 1024). These observations imply that the large latency and model size are necessary for the DNN baseline model to guarantee a relatively good performance, which is a primary concern in real-world applications. Therefore, 3 hidden layers and 2048 units for each layer were the default setting for the DNN baseline in all subsequent experiments. However, with the oracle information, all the evaluation metrics seemed to be insensitive to different settings, which motivated us to design the compact MOLE architecture with multi-objective to make clean speech prediction easier.

B. Experiments on the MOLE Framework

To demonstrate the complementarity of multiple streams, we first define three different MOLE variants in Table V. MOLE1 uses only learned targets in the LPS domain while MOLE2

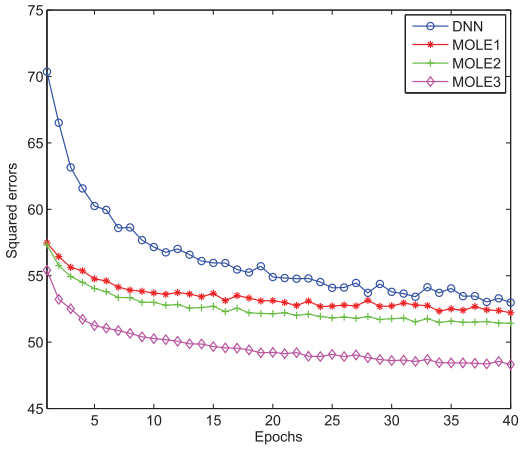

Fig. 3. Learning curves of DNN and MOLE on the development set.

adopts the additional learned targets in the MFCC domain and MOLE3 employs all the targets, including that in the GFCC domain, as shown in Fig. 1. In this subsection, the acoustic context for both the DNN baseline and MOL-DNN was set to 4 frames (1 current frame plus 3 historical frames), which would generate no hard latency (no use of future frames) in real applications. For MOLE systems without post-processing, the reconstructed clean speech waveform was obtained from the clean LPS estimate of the MOE stage. For MOLE2 and MOLE3, the MFCC and GFCC domain estimates were not directly used for the final clean speech reconstruction but were adopted to improve the generalization capability of the network, as inspired by our previous work [25].

Fig. 3 illustrates the comparison of learning curves between the DNN and the three MOLE variants using their averaged squared errors on the development set. Clearly, the learning curves of the MOLE systems with their multi-objective ensembles achieved more stable and better convergence than did that of the DNN approach. When MFCCs and GFCCs were included in the MOLE framework, the squared errors decreased at each step. In particular, the MOLE3 system with GFCC feature had a much smaller average squared error, which is consistent with the LSD values listed in Table VII. This result demonstrates the strong complementarity among the three feature types. More interestingly, the initial point of the MOLE3 learning curve was already close to the convergence point of the DNN learning curve, which demonstrates the importance of using multiple feature streams.

Tables VI and VII show comparisons of the average performance on the test set by the DNN baseline and several MOLE systems (PESQ and STOI) and (SSNR and LSD), respectively, at different SNR levels across 15 unseen noise types. To simplify Tables VI and VII, we list only four SNR levels because the performances across different SNR levels were quite stable. Several observations can be made. First, MOLE1 outperformed DNN in most cases, especially on the STOI metric (e.g., from 66.3 to 70.8 at −5 dB input SNR). This shows the effectiveness of the dynamic noise and IRM information in the LPS domain, whose complementarity was verified in our previous work [20]. There

TABLE VI
COMPARISON OF AVERAGE PESQ AND STOI RESULTS BY THE DNN BASELINE AND SEVERAL MOLE SYSTEMS ON TEST SETS AT FOUR SNR LEVELS ACROSS 15 UNSEEN NOISE TYPES

SNR		10 dB	5 dB	0 dB	−5 dB	Ave
DNN	PESQ	3.07	2.75	2.35	1.88	2.51
	STOI	91.4	86.4	78.2	66.3	80.6
MOLE1	PESQ	3.09	2.79	2.44	2.03	2.59
	STOI	92.0	87.6	80.7	70.8	82.8
MOLE2	PESQ	3.16	2.86	2.51	2.10	2.66
	STOI	92.8	88.5	81.7	72.0	83.8
MOLE3	PESQ	3.18	2.88	2.53	2.13	2.68
	STOI	92.8	88.6	81.8	72.1	83.9
MOLE3 + PP	PESQ	3.25	2.94	2.58	2.18	2.74
	STOI	93.3	88.9	81.8	72.0	84.0

TABLE VII
COMPARISON OF AVERAGE SSNR AND LSD RESULTS BY THE DNN BASELINE AND SEVERAL MOLE SYSTEMS ON TEST SETS AT FOUR SNR LEVELS ACROSS 15 UNSEEN NOISE TYPES

SNR		10 dB	5 dB	0 dB	−5 dB	Ave
DNN	SSNR	5.61	4.05	2.41	0.94	3.25
	LSD	2.79	3.22	3.92	5.27	3.80
MOLE1	SSNR	5.75	4.09	2.33	0.66	3.21
	LSD	2.57	2.98	3.56	4.34	3.36
MOLE2	SSNR	6.12	4.37	2.52	0.78	3.45
	LSD	2.49	2.97	3.61	4.36	3.35
MOLE3	SSNR	6.57	4.75	2.82	0.97	3.78
	LSD	2.38	2.86	3.49	4.25	3.25
MOLE3 + PP	SSNR	7.28	5.35	3.33	1.45	4.35
	LSD	2.28	2.79	3.45	4.29	3.20

are two exceptions for the SSNR measure at the 0 dB and −5 dB input SNRs. These exceptions may be due to the aggressive noise reduction of the DNN model at low SNRs (better SSNR) which often degraded the listening quality and intelligibility (worse PESQ and STOI). Second, by adding the learned targets in the MFCC domain, PESQ, STOI and SSNR all consistently improved from MOLE1 to MOLE2. In addition, the PESQ and STOI gains across the different SNR levels were quite stable. Third, the learned targets in the GFCC domain (MOLE3) yielded consistent improvements over MOLE2 on the SSNR and LSD measures. In addition, the reduction in LSD from MOLE2 to MOLE3 (0.1 dB on average) was more effective than that from MOLE1 to MOLE2 (0.01 dB on average). Finally, the post-processing (MOLE3 + PP) by the multi-objective ensemble of MOL-DNN and MOE-DNN generated consistent gains over MOLE3, especially in the PESQ and SSNR metrics. Overall, from Tables VI and VII, we can conclude that all the multi-objective sets are quite complementary (similar to the oracle experiments shown in Table III) and it contributed to the rise in the four evaluation metrics that measure speech quality, speech intelligibility, noise reduction and speech distortion. From DNN to MOLE3 + PP, we adopted the boosting concept for the ensemble of multiple types of learned weak stream to create a strong predictor. Accordingly, consistently large improvements were achieved for the four evaluation metrics at four SNR levels, with average gains of 0.23, 3.4, 1.1 dB, and 0.6 dB for PESQ, STOI, SSNR and LSD, respectively.

TABLE VIII
SYSTEMS WITH DIFFERENT INPUT AND OUTPUT FEATURE SETTINGS

System	Input Features	Output Features
A/B	Noisy + Static Noise	Clean + Dynamic Noise + IRM
C	Output of A	Clean LPS
D	Output of A + Noisy	Clean LPS
E	Output of A + Noisy	+ LPS IRM
F	Output of A + Noisy	+ MFCCs + GFCCs
G	Input of F without IRM	Output of F

TABLE IX
AVERAGE PERFORMANCE COMPARISON ON TEST SETS ACROSS 15 UNSEEN NOISE TYPES AT ALL SNR LEVELS AMONG THE SYSTEMS SHOWN IN TABLE VIII

	A	B	C	D	E	F	G
PESQ	2.89	2.91	2.92	2.96	3.03	3.04	3.03
STOI	86.0	86.4	87.0	87.7	88.1	88.3	88.1
SSNR	4.26	4.38	4.24	5.13	6.00	6.14	5.98
LSD	3.49	3.47	3.21	2.90	2.80	2.73	2.79

Table VIII provides descriptions of several DNN systems with different input and output feature settings. Systems A and B are MOL-DNN models with two hidden layers and different numbers of hidden nodes in each layer (1024 for A and 1600 for B). Systems C-G are all MOE-DNNs with the same hidden layer and node settings as System A. System C was a special MOE-DNN model in which no noisy features were used as the input and only clean LPS features were predicted in the output layer. System D, E and F used the same input; however, D estimated only clean LPS while E estimated clean LPS and LPS IRM simultaneously, and F is the proposed MOLE3 + PP from Tables VI and VII. The difference between G and F was that the IRM information was not included in the input to G.

Table IX shows a performance comparison of the systems listed in Table VIII. These results provide a strong justification for the choice of our proposed MOLE setup. First, large performance gaps existed between the purely multi-tasking systems A/B and the proposed MOLE system F on all the evaluation metrics even though the model sizes of B and F were almost the same. This result implies that multi-objective ensembling is quite important and the multi-tasking DNN (MOL-DNN) in the MOLE is just a way to estimate the multi-stream fed to the MOE-DNN as the ensemble. Second, comparing C and D, we can see that the original noisy features are complementary with the estimated multiple streams. Third, D, E and F were designed to provide justification for the choices made in the MOE-DNN setup. When IRM information was incorporated in the training targets (D versus E), all four metrics improved. Because the dimensions of MFCC and GFCC were quite low, we adopted them in our final MOLE framework because they result in slight but consistent improvements across all measures (E versus F). Even the MFCC and GFCC outputs of MOE-DNN were not directly used for the final clean speech estimate, instead, they served as a multi-tasking approach to improve the generalization capability of the network. Fourth, F consistently outperformed G on all metrics although the gains were not large. Accordingly, IRMs were still included as input to the MOE-DNN.

TABLE X
AVERAGE PERFORMANCE COMPARISON ON TEST SETS ACROSS 15 UNSEEN NOISE TYPES OF ALL SNR LEVELS AMONG MOLE ARCHITECTURES TRAINED WITH DIFFERENT SETTINGS OF (N_{L}, N_{U}), WHERE N_{L} AND N_{U} ARE THE NUMBERS OF HIDDEN LAYERS AND UNITS USED IN THE MOL-DNN OR MOE-DNN, RESPECTIVELY

MOL-DNN	MOE-DNN	PESQ	STOI	SSNR	LSD
(2, 1024)	(2, 1024)	3.04	88.3	6.14	2.73
(2, 1024)	(3, 2048)	3.04	88.3	6.13	2.73
(3, 2048)	(2, 1024)	3.02	88.6	6.27	2.67
(3, 2048)	(3, 2048)	3.02	88.7	6.40	2.63

Table X shows a comparison of the average performance on the test sets across 15 unseen noise types with all SNR levels among four MOLE3 + PP architectures trained with different MOL-DNN and MOE-DNN parameter settings. With the same MOL-DNN settings, using a quite compact setting (2, 1024) for the MOE-DNN achieved a comparable performances on all four metrics compared to the default setting of (3, 2048) used for the DNN baseline. This indicates that MOE-DNN can be compact given its awareness of the multiple streams and provided solid confirmation of the findings from the oracle experiments in Table IV. Similar observations can be made for the MOL-DNN model. This implies that although each learned target might be not the most accurate when using a compact setting (2, 1024) for MOL-DNN, the performance after the multi-objective ensemble using MOE-DNN via the boosting concept is comparable to that of the default setting (3, 2048). Overall, these results serve to demonstrate the compactness of the MOLE architecture design well. Note that in the following experiments the notation "MOLE" denotes MOLE3 plus post-processing (MOLE3 + PP in Tables VI and VII).

C. Comparison With DNN and LSTM

In addition to the performance, the time latency, model size and computational complexity are also crucial for deep learning based methods in real-time speech applications. In this subsection, we conduct an overall comparison of both the performance and practical issues among different speech enhancement approaches using the DNN, LSTM and MOLE architectures.

First, a comparison of different deep learning approaches on the test sets for all the evaluation metrics across 15 unseen noise types at all SNR levels is listed in Table XI. The setting of the input frame number τ as the acoustic context determined the hard latency of the deep models: $\tau = 1$ denoted that only the central frame was adopted with no hard latency; $\tau = 4$ used 3 history frames; and $\tau = 7$ employed both 3 history and future frames. For the DNN models, the acoustic context was quite important for all evaluation metrics, which was also demonstrated in [9], [10]. For the LSTM model, the concatenated future frames improved the performance while the concatenated history frames caused the results to be slightly worse. This was reasonable because the history information is already embedded in the recursive structure of the LSTM; thus, including previous frames

TABLE XI
PERFORMANCE AND PRACTICAL ISSUE COMPARISON ON TEST SETS ACROSS 15 UNSEEN NOISE TYPES WITH ALL SNR LEVELS AMONG DNN, LSTM AND MOLE TRAINED WITH DIFFERENT INPUT FRAME NUMBER (τ) SETTINGS

τ	System	Performance Gain						Practical Issue	
		PESQ	STOI	SSNR	LSD	NR	SSSNR	N_{M}	N_{T}
–	Noisy	2.32	82.1	1.20	7.12	–	–	–	–
1	DNN	0.44	2.5	3.37	3.53	5.10	2.83	1	1
	LSTM	0.59	4.5	3.99	3.87	5.67	3.83	1.4	102
	MOLE	0.68	5.8	4.64	4.28	6.45	4.39	0.5	0.6
4	DNN	0.50	3.3	3.47	3.78	5.17	2.94	1.2	1.1
	LSTM	0.59	4.3	3.62	3.73	5.26	3.32	1.7	103
	MOLE	0.72	6.2	4.94	4.39	6.66	4.64	0.6	0.9
7	DNN	0.57	3.9	3.48	3.91	5.11	2.87	1.3	1.3
	LSTM	0.68	5.6	3.98	4.12	5.62	3.68	2.0	104
	MOLE	0.77	6.6	4.86	4.42	6.57	4.54	0.7	1.1

For noisy signals, the absolute values of evaluation metrics are presented, while for other signals, performance gains are presented. N_{M} and N_{T} represent the model size and run-time latency, respectively, which are normalized by the DNN system with $\tau = 1$.

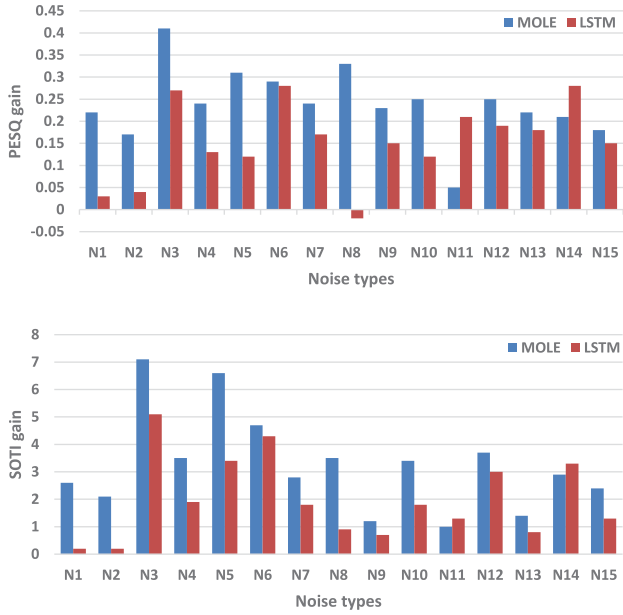

Fig. 4. Average PESQ/STOI gains of the MOLE and LSTM approaches compared with the DNN approach among 15 unseen noise types across all SNR levels.

in the LSTM might cause information redundancy. The observation for the MOLE model was similar to that of the DNN model; however, the problem of hard latency, which is required for DNNs to achieve better performance, was partly alleviated in MOLE because the performance gains in the MOLE system from $\tau = 1$ to $\tau = 7$ (e.g., 0.09 in PESQ and 0.8 in STOI) were less important than those in the DNN system (e.g., 0.13 in PESQ and 1.4 in STOI). This implies that the MOLE model with its multi-objective set relaxed the constraint of long frame expansion compared to the DNN model, and it possesses a capacity similar to the LSTM model for longer acoustic context modelling. More interestingly, even the MOLE system with no hard latency ($\tau = 1$) achieved consistent improvements on all metrics (e.g., a PESQ gain of 0.11 and a STOI gain of 1.9) over the best DNN system with $\tau = 7$, and it yielded comparable PESQ/STOI performance and better SSNR/LSD and NR/SSSNR performance compared to with the best LSTM system with $\tau = 7$.

Table XI also provides comparisons of practical issues such as model size and computational complexity. The size of each model (N_{M}), measured by the amount of neural network parameters, and the computational complexity of each model (N_{T}), measured by the time latency of neural network processing were normalized by those of the DNN model with $\tau = 1$. We observed that MOLE model size is one-half the DNN model size and one-third the LSTM model size for the same acoustic context τ, which confirms the compact design of both MOL-DNN and MOE-DNN in the proposed MOLE framework based on the motivation of the oracle experiments in Section IV-A. Regarding computational complexity, at a smaller τ values, the MOLE model was much more efficient than were the DNN and LSTM models. When $\tau = 1$, the MOLE model achieved a 40% reduction in run-time latency over the DNN model and it was 170 times faster than the LSTM model. In summary, the MOLE approach achieved consistent improvements in all the objective evaluation metrics over both DNN and LSTM approaches while using a much smaller model size and exhibiting less run-time latency.

Fig. 4 shows the average PESQ and STOI gains of the MOLE and LSTM approaches compared with the DNN approach among 15 unseen noise types across all SNR levels. The input frame number τ was set to 1 for all the deep models because this setting was more practical and induced no hard latency. For both the PESQ and STOI measures, MOLE consistently and substantially outperformed the DNN for all noise types, demonstrating its effectiveness and robustness for both stationary and non-stationary environments. The gains by LSTM were not stable across different noise types (e.g., it achieved slight improvements on N1/N2, but performed worse on N8). Comparing MOLE with LSTM, MOLE performs better in most cases, with two exceptions: N11 and N14. N11 is machine-gun noise and N14 is speech babble noise. These two noise types are relatively uncomplicated cases in the NOISEX-92 corpus. The LSTM approach adopts recurrent layers to preserve information from a long historical context, while the MOLE approach uses multi-stream for adaptive training. This is the main difference between these two approaches. Taking N14 (babble noise) as shown in Fig. 5 for example, it is relatively easy for both MOLE and LSTM approaches to remove background noise. However, speech enhanced by the MOLE approach introduced slightly more speech distortions than the LSTM approach shown in the rectangular boxes, thus leading to a slight reduction in PESQ and STOI.

To give intuitive interpretations of the observations made from Fig. 4, spectrograms of four representative examples with the noise types of N3 (Destroyer Engine), N8 (HF Channel), N11 (Machine Gun) and N14 (Speech Babble) are shown in Fig. 5. For N3, MOLE generated the most effective gains over the DNN in both PESQ and STOI. Without using a longer acoustic

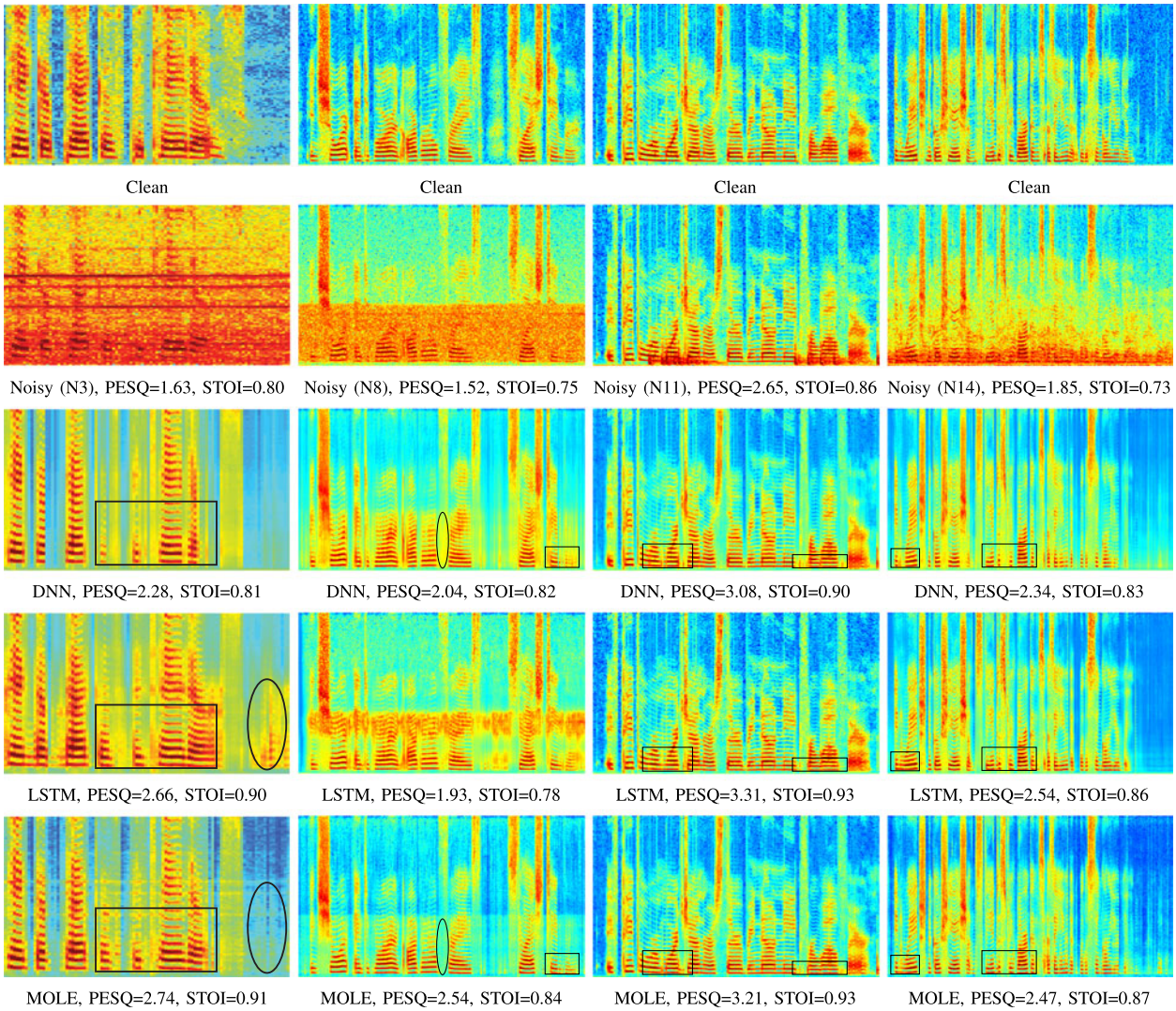

Fig. 5. Spectrograms of four representative examples (from Fig. 4) with four noise types: N3 (Destroyer Engine), N8 (HF Channel), N11 (Machine Gun) and N14 (Speech Babble), respectively. The input SNRs of noisy speech for all the noise types are 5 dB. Each column corresponds to one example set, showing clean speech, noisy speech, DNN-enhanced speech, LSTM-enhanced speech and MOLE-enhanced speech. The input frame number τ for all models was set to 1.

context, the DNN often cut one speech segment into small fragments due to its aggressive noise removal (e.g., the black boxes of the N3 and N14 cases) while MOLE preserve the speech segmentation well based on its awareness of the multiple streams, as shown in the rectangle boxes. Meanwhile, LSTM seemed to misjudge some noise segmentations as speech, as shown in the circled area. For N8, it was quite clear that one certain sub-band noise segmentation remained in the LSTM-processed speech signal, which led to the poor listening quality with worse PESQ than the DNN in Fig. 4. When N11 (Machine Gun) was the burst noise, LSTM, using its historical information, could better track the noise than MOLE, which used only the central frame, thus yielding better PESQ results. Similarly, LSTM performed slightly better than MOLE for the non-stationary N14 (Speech Babble). For more examples, readers can refer to the link.[1]

Generally, DNN did well at noise reduction but had a high risk of yielding more speech distortions or even removing speech segments. LSTM used its long-term historical information to better preserve the speech segments than did the DNN. However, LSTM failed to completely remove the background noise in several of the unseen environments. The behaviours of the DNN and LSTM can be explained as resulting from the different trade-offs made between noise reduction and speech preservation. In contrast, MOLE achieved better noise reduction and speech preservation in most cases, which is why MOLE yielded the best performances across all the evaluation metrics.

[1]http://home.ustc.edu.cn/~xiaosong/demo/MOLE.html

V. Conclusion

In this study, we focus on the practical issues related to deep learning based speech enhancement and propose a compact two-stage multi-objective learning and ensembling DNN framework. This goal is accomplished by effectively utilizing of multi-stream features. The first-stage MOL-DNN predicts three useful sets of features: LPS, MFCC and GFCC. Each feature set consists of that feature for clean speech and its

corresponding IRM and that for dynamic noise. These three feature sets are used together with the noisy speech features as input to the second-stage MOE-DNN, which in turn predicts three sets of output features for clean speech: LPS and MFCC and GFCC and their corresponding IRMs. By estimating the auxiliary information of the related MFCC and GFCC, the primary LPS and its IRM features are better estimated and used for post-processing. The experimental results on 15 unseen noise types show that the proposed MOLE framework yields consistent improvements over both DNN- and LSTM-based techniques on the metrics PESQ, STOI, SSNR, LSD, NR and SSSNR. It delivers this improved performance using a smaller model size while requiring lower run-time latency based on all three test cases used for input context expansion, namely, 1 frame with no expansion, 4 frames—including 3 previous frames but with no latency, and 7 frames with a delay of 3 frames. The 1-frame MOLE-based SE system is of particular interest because it outperforms even the 7-frame DNN-based SE system. It also yields better performances but with 170 times less latency compared to the 4-frame LSTM-based SE system. Finally, although the proposed MOLE framework makes strides in addressing the practical issues in real-time applications, there are still huge gaps between the performances achievable in oracle experiments and the results attained thus far on real datasets. In future research, we will attempt to bridge these gaps.

REFERENCES

[1] J. Benesty, S. Makino, and J. D. Chen, *Speech Enhancement*. New York, NY, USA: Springer, 2005.

[2] P. C. Loizou, *Speech Enhancement: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2013.

[3] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[4] J. S. Lim and A. V. Oppenheim, "All-pole modeling of degraded speech," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 3, pp. 197–210, Jun. 1978.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.

[6] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust. Speech and Signal Process.*, vol. 33, no. 2, pp. 443–445, Apr. 1985.

[7] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Process.*, vol. 81, no. 11, pp. 2403–2418, 2001.

[8] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 466–475, Sep. 2003.

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

[10] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[11] J. Du and Q. Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, 2008, pp. 569–572.

[12] M. Sun, X. Zhang, H. V. Hamme, and T. F. Zheng, "Unseen noise estimation using separable deep auto encoder for speech enhancement," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 24, no. 1, pp. 93–104, Jan. 2016.

[13] A. L. Maas, T. M. ONeil, A. Y. Hannun, and A. Y. Ng, "Recurrent neural network feature enhancement: The 2nd CHiME challenge," in *Proc. 2nd CHiME Workshop Mach. Listening Multisource Environ. Held Conjunction ICASSP*, 2013, pp. 79–80.

[14] A. L. Maas, Q. V. Le, T. M. ONeil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Proc. Interspeech*, 2012, pp. 22–25.

[15] Z. Chen, S. Watanabe, H. Erdogan, and J. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Proc. Interspeech*, 2015, pp. 3274–3278.

[16] M. Wollmer, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, "Feature enhancement by bidirectional LSTM networks for conversational speech recognition in highly non-stationary noise," in *Proc. ICASSP*, 2013, pp. 6822–6826.

[17] L. Sun, J. Du, L. R. Dai, and C. H. Lee, "Multiple-target deep learning for LSTM-RNN based speech enhancement," in *Proc. Hands-free Speech Commun. Microphone Arrays*, 2017, pp. 136–140.

[18] Y. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, Jul. 2013.

[19] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for joint enhancement of magnitude and phase," in *Proc. ICASSP*, 2016, pp. 5220–5224.

[20] Q. Wang, J. Du, L. R. Dai, and C. H. Lee, "Joint noise and mask aware training for DNN-based speech enhancement with SUB-band features," in *Proc. Hands-free Speech Commun. Microphone Arrays*, 2017, pp. 101–105.

[21] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[22] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 21, no. 10, pp. 2140–2151, Oct. 2013.

[23] S.-W. Fu, Y. Tsao, and X. Lu, "SNR-aware convolutional neural network modeling for speech enhancement," in *Proc. Interspeech*, 2016, pp. 3758–3772.

[24] Z.-Q. Wang and D. L. Wang, "Recurrent deep stacking networks for supervised speech separation," in *Proc. ICASSP*, 2017, pp. 71–75.

[25] Y. Xu, J. Du, Z. Huang, L. R. Dai, and C. H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," in *Proc. Interspeech*, 2015, pp. 1508–1512.

[26] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[27] Y. Shao, S. Srinivasan, and D. Wang, "Incorporating auditory feature uncertainties in robust speaker identification," in *Proc. ICASSP*, 2007, pp. 277–280.

[28] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. ICASSP*, 2001, pp. 749–752.

[29] R. E. Schapire, "The strength of weak learnability," *Mach. Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[30] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *Proc. Interspeech*, 2014, pp. 2670–2674.

[31] X.-L. Zhang and D. L. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 24, no. 5, pp. 967–977, May 2016.

[32] P. G. Shivakumar and P. Georgiou, "Perception optimized deep denoising autoencoders for speech enhancement," in *Proc. Interspeech*, 2016, pp. 3743–3747.

[33] S. Nie, H. Zhang, X. L. Zhang, and W. J. Liu, "Deep stacking networks with time series for speech separation," in *Proc. ICASSP*, 2014, pp. 6667–6671.

[34] Y. Qian, T. Tan, and D. Yu, "An investigation into using parallel data for far-field speech recognition," in *Proc. ICASSP*, 2016, pp. 5725–5729.

[35] Y. Qian, T. Tan, D. Yu, and Y. Zhang, "Integrated adaptation with multi-factor joint-learning for far-field speech recognition," in *Proc. ICASSP*, 2016, pp. 5770–5774.

[36] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, pp. 1635–1638.

[37] H. Bourlard, S. Dupont, and C. Ris, "Multi-stream speech recognition," Tech. Report IDIAR-RR 96-07, 1996.

[38] F. Xie and D. V. Compernolle, "A family of MLP based nonlinear spectral estimators for noise reduction," in *Proc. ICASSP*, 1994, pp. 53–56.
[39] D. O'Shaughnessy, *Speech Communication: Human and Machine*. India: Universities Press, 1987.
[40] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the gammatone function," *Applied Psychology Unit Report 2341*, 1988.
[41] E. A. Wan and A. T. Nelson, "Networks for speech enhancement," *Handbook Neural Netw. Speech Process.*, Artech House, Boston, MA, USA, 1999.
[42] R. Vergin, D. OShaughnessy, and A. Farhat, "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 5, pp. 525–532, Sep. 1999.
[43] K. S. R. Murty and B. Yegnanarayana, "Combining evidence from residual phase and MFCC features for speaker recognition," *IEEE Signal Process. Lett.*, vol. 13, no. 1, pp. 52–55, Jan. 2006.
[44] A. V. Oppenheim, *Discrete-Time Signal Processing*. India: Pearson Education India, 1999.
[45] D. L. Wang and G. J. Broun, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2006.
[46] Y. Shao, Z. Jin, D. Wang, and S. Srinivasan,"An auditory-based feature for robust speech recognition," in *Proc. ICASSP*, 2009, pp. 4625–4628.
[47] Y. Shao and D. Wang, "Robust speaker identification using auditory features and computational auditory scene analysis," in *Proc. ICASSP*, 2008, pp. 1589–1592.
[48] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398–7402.
[49] N. Roman, D. Wang, and G. J. Brown, "Speech segregation based on sound localization," *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 2236–2252, 2003.
[50] D. L. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech Separation by Humans and Machines* New York, NY, USA: Springer, 2005, pp. 181–197.
[51] N. Madhu, A. Spriet, S. Jansen, R. Koning, and J. Wouters, "The potential for speech intelligibility improvement using the ideal binary mask and the ideal wiener filter in single channel noise reduction systems: Application to auditory prostheses," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 21, no. 1, pp. 63–72, Jan. 2013.
[52] C. Hummersone, T. Stokes, and T. Brookes, "On the ideal ratio mask as the goal of computational auditory scene analysis," in *Blind Source Separation*. New York, NY, USA: Springer, 2014, pp. 349–368.
[53] A. Narayanan and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 22, no. 4, pp. 826–835, Apr. 2014.
[54] G. Hu, "100 nonspeech environmental sounds," http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html, 2014.
[55] J. S. Garofolo *et al.*, "TIMIT acoustic-phonetic continuous speech corpus," Philadelphia, PA, USA: Linguistic Data Consortium, 1993.
[56] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, no. 3, pp. 247–251, 1993.
[57] D. Yu *et al.*, "An introduction to computational networks and the computational network toolkit," Microsoft Technical Report MSR-TR-2014C112, 2014.
[58] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv:1207.0580, 2012.
[59] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, vol. 3, no. 28, pp. 1310–1318, 2013.
[60] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.
[61] T. Lotter and P. Vary, "Speech enhancement by MAP spectral amplitude estimation using a super-gaussian speech model," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 1110–1126, 2005.

Qing Wang received the B.S. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2012, where she is currently working toward the Ph.D. degree. Her research interests include speech enhancement and robust speech recognition.

Jun Du received the B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively. From 2004 to 2009, he was with iFlytek Speech Lab, USTC. During the above period, he worked as an Intern twice for 9 months at Microsoft Research Asia (MSRA), Beijing, China. In 2007, he was a Research Assistant for 6 months with the Department of Computer Science, The University of Hong Kong. From July 2009 to June 2010, he was with iFlytek Research on speech recognition. From July 2010 to January 2013, he joined MSRA as an Associate Researcher, working on handwriting recognition, OCR, and speech recognition. Since February 2013, he has been with the National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP), USTC.

Li-Rong Dai was born in China in 1962. He received the B.S. degree in electrical engineering from Xidian University, Xian, China, in 1983, and the M.S. degree from the Hefei University of Technology, Hefei, China, in 1986, and the Ph.D. degree in signal and information processing from the University of Science and Technology of China (USTC), Hefei, China, in 1997. He joined USTC in 1993. He is currently a Professor at the School of Information Science and Technology, USTC. His research interests include speech synthesis, speaker and language recognition, speech recognition, digital signal processing, voice search technology, machine learning, and pattern recognition. He has published more than 50 papers in these areas.

Chin-Hui Lee is a Professor in the School of Electrical, and Computer Engineering, Georgia Institute of Technology. Before joining academia in 2001, he had 20 years of industrial experience ending in Bell Laboratories, Murray Hill, NJ, USA, as a Distinguished Member of Technical Staff, and the Director of the Dialogue Systems Research Department. He is a Fellow of ISCA. He has published more than 450 papers, and 30 patents, and was highly cited over 30 000 times for his original contributions with an h-index of 66 on Google Scholar. He received numerous awards, including the Bell Labs President's Gold Award in 1998. He also received the SPS's 2006 Technical Achievement Award for "Exceptional Contributions to the Field of Automatic Speech Recognition." In 2012, he was invited by ICASSP to give a plenary talk on the future of speech recognition. In the same year, he received the ISCA Medal in scientific achievement for pioneering and seminal contributions to the principles and practice of automatic speech and speaker recognition.