A MAXIMUM LIKELIHOOD APPROACH TO MULTI-OBJECTIVE LEARNING USING GENERALIZED GAUSSIAN DISTRIBUTIONS FOR DNN-BASED SPEECH ENHANCEMENT

Shu-Tong Niu[⋆]*, Jun Du*[⋆]*, Li Chai*[⋆]*, Chin-Hui Lee*[†]

[⋆]University of Science and Technology of China, Hefei, Anhui, P.R.China
[†] Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT

The multi-objective learning using minimum mean squared error criterion for DNN-based speech enhancement (MMSE-MOL-DNN) has been demonstrated to achieve better performance than single output DNN. However, one problem of MMSE-MOL-DNN is that the prediction error values on different targets have a very broad dynamic range, causing difficulty in DNN training. In this paper, we extend the maximum likelihood approach proposed in our previous work [1] to the multi-objective learning for DNN-based speech enhancement (ML-MOL-DNN) to achieve the automatic adjustment of the dynamic range of prediction error values on different targets. The conditional likelihood function to be maximized is derived under the generalized Gaussian distribution (GGD) error model. Moreover, the control of the dynamic range of the prediction error values on different targets is achieved by the scale factors in GGD. Furthermore, we propose a method to update the shape factors automatically utilizing the one-to-one mapping between the kurtosis and shape factor in GGD instead of manual adjustment. The experimental results show that our ML-MOL-DNN can achieve better performance than MMSE-MOL-DNN in terms of different objective measures.

Index Terms— multi-objective learning, maximum likelihood, deep neural network, shape factors update, generalized Gaussian distribution

1. INTRODUCTION

Speech enhancement is an important problem in signal processing which is widely used in practice [2]. In the past several decades, numerous speech enhancement methods were developed which can be divided into unsupervised methods and supervised methods. Many classic unsupervised methods, such as spectral subtraction [3], Wiener filtering [4], a MMSE estimator [5] and so on, can achieve good performance in stationary noise conditions but often fail to track non-stationary noise.

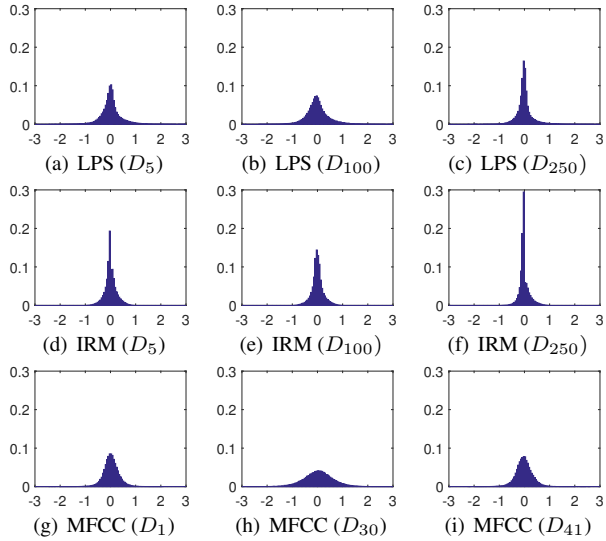
Supervised methods have developed rapidly in recent years with the great progress of people's research on deep learning technologies. In supervised methods, machine learning plays an important role, and deep neural networks (DNNs) have shown great advantages in many supervised learning tasks. Many types of DNNs were used in supervised speech enhancement such as feed-forward DNNs [6, 7, 8], recurrent neural networks (RNNs) [9, 10], convolutional neural networks (CNNs) [11, 12], generative adversarial networks (GANs) [13] and so on. According to the training targets, these supervised speech enhancement methods can be categorized into two main groups: mapping-based methods and masking-based methods. Mapping-based methods directly learn the mapping between clean speech and noisy speech [8], masking-based methods learn a time-frequency (T-F) mask like ideal ratio mask (IRM) [6] from a noisy signal firstly, and then obtain the enhanced features from the estimated mask. Moreover, the features also play an important role in supervised speech enhancement, and there are many types of features used in supervised speech enhancement such as log-power spectra (LPS) features [14], mel-frequency cepstral coefficient (MFCC) [15], gammatone frequency cepstral coefficient (GFCC) [15] and so on.

Recently, advanced objective functions have been explored [12, 16, 17, 18]. A method is to construct a multi-objective learning (MOL) framework which learns multiple types of features by using joint objective functions [12, 16, 17]. These MOL frameworks based on MMSE criterion can achieve better performance than single output DNN. However, a disadvantage of these methods is that the values of the prediction errors vary greatly among different target feature types, which makes it difficult for DNN to fully learn the targets with small prediction error values during the training process. In order to solve this problem, we propose ML-MOL-DNN approach. Inspired by the multi-stream method [19, 20], we apply the ML criterion [1] within the probabilistic learning framework to multi-objective learning through the concept of multi-stream. We treat different types of target features as different streams and assume they are independent of each other. We also use the generalized Gaussian distribution as the approximate distribution of each target feature. In this case, we can get the conditional likelihood function of all types of target features and optimize the DNN parameters by maximizing conditional likelihood function. The purpose of this method is to reduce the difference in prediction error values among different targets by scale factors, so that the information in all different types of target features can be used fully during the training process. We also propose a new parameter update strategy based on the previous ML method using GGD for DNN-based speech enhancement (ML-GGD-DNN) [1] to fit the shape factors which change during the training process. As a special case of our method, we use three types of target features (LPS, IRM, MFCC) in this study. The experiment results show that the proposed ML-MOL-DNN can effectively reduce the difference of prediction error values among different targets compared with the MMSE-MOL-DNN. The evaluation on the WSJ0 corpus [21] also shows that the proposed ML-MOL-DNN can achieve a significantly improvement compared with the MMSE-MOL-DNN. Moreover, the proposed shape factors update strategy can achieve better performance compared with our previous ML-GGD-DNN approach.

2. THE PROPOSED ML-MOL-DNN

2.1. Motivation

Fig. 1 shows the distributions of selected dimensions of prediction error vectors from three target feature types (LPS, IRM, MFCC) on



(a) LPS (D_5) (b) LPS (D_{100}) (c) LPS (D_{250})
(d) IRM (D_5) (e) IRM (D_{100}) (f) IRM (D_{250})
(g) MFCC (D_1) (h) MFCC (D_{30}) (i) MFCC (D_{41})

Fig. 1. The distributions for selected dimensions of the prediction error vectors from the well-trained MMSE-MOL-DNN on the cross-validation set: (a)-(c) refer to LPS, (d)-(e) refer to IRM and (g)-(i) correspond to MFCC.

the cross-validation set for well-trained MMSE-MOL-DNN configured as described in Section 3.1. It is observed that the prediction error values are quite different among different types of target features, which makes it difficult for DNN to learn the targets with small prediction error values. This is our motivation to design ML-MOL-DNN to adjust the prediction error values from different target feature types automatically. Furthermore, all distributions shown in Fig. 1 are generally Gaussian-like, which illustrates the reasonability of using GGD to approximate the distribution of prediction error vectors in each dimension in Section 2.2.

2.2. Derivation for ML-MOL-DNN

In conventional MMSE-MOL-DNN, one target usually corresponds to one type of feature, and a mini-batch stochastic gradient descent algorithm is performed in mini-batches with multiple epochs to improve the following error function:

$$E = \frac{1}{N}\sum_{n=1}^{N}\sum_{s=1}^{S}\lambda_s \times \|\hat{\boldsymbol{x}}_{n,s}(\boldsymbol{y}_{n,s}, \boldsymbol{W}) - \boldsymbol{x}_{n,s}\|_2^2 \tag{1}$$

where N represents the mini-batch size, S is the number of feature types in DNN output, λ_s denotes the weighting factor of the s-th feature, $\boldsymbol{x}_{n,s}$ and $\hat{\boldsymbol{x}}_{n,s}(\boldsymbol{y}_{n,s}, \boldsymbol{W})$ represent the reference and estimated s-th feature at sample index n respectively, $\boldsymbol{y}_{n,s}$ is the input feature vector, $\boldsymbol{W}$ is the DNN parameter set to be learned.

In ML-MOL-DNN, we treat different types of target features as different streams and introduce the multi-stream method [19, 20] under the probabilistic framework. According to the multi-stream method, if we relax the assumption of dependence among the different types of features, the conditional likelihood function of all types of target features can be seen as a product of the likelihood functions of single target feature, raised to the appropriate stream exponents that capture the reliability of each type of target feature:

$$p(\boldsymbol{x}_n|\boldsymbol{y}_n, \boldsymbol{W}, \boldsymbol{\Theta}) = \prod_{s=1}^{S}[p(\boldsymbol{x}_{n,s}|\boldsymbol{y}_{n,s}, \boldsymbol{W}, \boldsymbol{\Theta}_s)]^{\gamma_s} \tag{2}$$

where $\boldsymbol{\Theta}_s$ represents the parameter set of conditional likelihood function in s-th feature, $\boldsymbol{x}_n = \{\boldsymbol{x}_{n,s}|s = 1, 2, ..., S\}$, $\boldsymbol{y}_n = \{\boldsymbol{y}_{n,s}|s = 1, 2, ..., S\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_s|s = 1, 2, ..., S\}$, γ_s are stream exponents which depend on feature types in general.

According to the ML-GGD-DNN approach [1], each dimension of the s-th feature prediction error vector follows a univariate GGD with zero mean, an unrestricted scale factor α_{s,d_s} and shape factor β_{s,d_s} at sample index n:

$$p(e_{n,s,d_s}|\alpha_{s,d_s}, \beta_{s,d_s}) = \frac{\beta_{s,d_s}}{2\alpha_{s,d_s}\Gamma(\frac{1}{\beta_{s,d_s}})}\exp\left(-\left(\frac{|e_{n,s,d_s}|}{\alpha_{s,d_s}}\right)^{\beta_{s,d_s}}\right) \tag{3}$$

If the reference vector $\boldsymbol{x}_{n,s}$ is a random variable and the prediction errors in all dimensions are independent, we can get the joint distribution for all dimensions at sample index n of s-th feature:

$$\begin{aligned} &p(\boldsymbol{x}_{n,s}|\boldsymbol{y}_{n,s}, \boldsymbol{W}, \boldsymbol{\alpha_s}, \boldsymbol{\beta_s}) \\ &= \prod_{d_s=1}^{D_s}\frac{\beta_{s,d_s}}{2\alpha_{s,d_s}\Gamma(\frac{1}{\beta_{s,d_s}})}\exp\left(-\left(\frac{|x_{n,s,d_s} - \hat{x}_{n,s,d_s}|}{\alpha_{s,d_s}}\right)^{\beta_{s,d_s}}\right) \end{aligned} \tag{4}$$

where $\boldsymbol{\alpha_s} = \{\alpha_{s,d_s}|d_s = 1, 2, ..., D_s\}$, $\boldsymbol{\beta_s} = \{\beta_{s,d_s}|d_s = 1, 2, ..., D_s\}$, $\boldsymbol{x}_{n,s} = \{x_{n,s,d_s}|d_s = 1, 2, ..., D_s\}$ and $\boldsymbol{y}_{n,s} = \{y_{n,s,d_s}|d_s = 1, 2, ..., D_s\}$. Therefore we can get a joint distribution of all types of target features under multi-stream framework shown in Eq. (2) as follows:

$$\begin{aligned} &p(\boldsymbol{x}_n|\boldsymbol{y}_n, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \prod_{s=1}^{S}\left[\prod_{d_s=1}^{D_s}\frac{\beta_{s,d_s}}{2\alpha_{s,d_s}\Gamma(\frac{1}{\beta_{s,d_s}})}\exp\left(-\left(\frac{|x_{n,s,d_s} - \hat{x}_{n,s,d_s}|}{\alpha_{s,d_s}}\right)^{\beta_{s,d_s}}\right)\right]^{\gamma_s} \end{aligned} \tag{5}$$

where $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_s|s = 1, 2, ..., S\}$, $\boldsymbol{\beta} = \{\boldsymbol{\beta}_s|s = 1, 2, ..., S\}$. Given a mini-batch training set with N data pairs $(\boldsymbol{Y}, \boldsymbol{X}) = \{(\boldsymbol{y}_n, \boldsymbol{x}_n)|n = 1, 2, ..., N\}$ and assuming that they are drawn independently from the distribution in Eq. (5), the corresponding likelihood function is:

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \prod_{n=1}^{N}p(\boldsymbol{x}_n|\boldsymbol{y}_n, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \prod_{n=1}^{N}\prod_{s=1}^{S}\left[\prod_{d_s=1}^{D_s}\frac{\beta_{s,d_s}}{2\alpha_{s,d_s}\Gamma(\frac{1}{\beta_{s,d_s}})}\exp\left(-\left(\frac{|x_{n,s,d_s} - \hat{x}_{n,s,d_s}|}{\alpha_{s,d_s}}\right)^{\beta_{s,d_s}}\right)\right]^{\gamma_s} \end{aligned} \tag{6}$$

where the parameter set $(\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is to be optimized. Accordingly, the log-likelihood function can be written as:

$$\begin{aligned} \ln p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \sum_{n=1}^{N}\ln p(\boldsymbol{x}_n|\boldsymbol{y}_n, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \sum_{n=1}^{N}\sum_{s=1}^{S}\sum_{d_s=1}^{D_s}\gamma_s \times \ln\left(\frac{\beta_{s,d_s}}{2\alpha_{s,d_s}\Gamma(\frac{1}{\beta_{s,d_s}})}\right) \\ &\quad - \sum_{n=1}^{N}\sum_{s=1}^{S}\sum_{d_s=1}^{D_s}\gamma_s \times \left(\frac{|x_{n,s,d_s} - \hat{x}_{n,s,d_s}|}{\alpha_{s,d_s}}\right)^{\beta_{s,d_s}} \end{aligned} \tag{7}$$

We can maximize Eq. (7) with respect to $\boldsymbol{\alpha}$. Then the update formula can be derived as:

$$\alpha_{s,d_s} = \left(\frac{\beta_{s,d_s}}{N} \sum_{n=1}^{N} |x_{n,s,d_s} - \hat{x}_{n,s,d_s}|^{\beta_{s,d_s}} \right)^{\frac{1}{\beta_{s,d_s}}} \quad (8)$$

Maximizing Eq. (7) with respect $\boldsymbol{W}$ is equivalent to minimize the following loss function:

$$E(\boldsymbol{W}) = \sum_{n=1}^{N} \sum_{s=1}^{S} \sum_{d_s=1}^{D_s} \gamma_s \times \left(\frac{|x_{n,s,d_s} - \hat{x}_{n,s,d_s}|}{\alpha_{s,d_s}} \right)^{\beta_{s,d_s}} \quad (9)$$

By comparing the Eq. (1) and Eq. (9), we can find that the ML-MOL-DNN can naturally reduce the difference in prediction error values among different types of target features by α_{s,d_s}, which can allow DNN to make full use of the information in different types of target features during the multi-objective training process. Furthermore, we should note that the conventional MMSE-MOL-DNN is a special case of ML-MOL-DNN where the scale factors are all the same in different targets and different dimensions, and our previous ML-GGD-DNN is also a special case of ML-MOL-DNN where the number of targets is equal to 1. Like [22], the weighting factors of different targets in all approaches are set to 1 in this study, and we will explore the impact of weighting factors on the performance in the future.

2.3. Update of shape factors

In our previous ML-GGD-DNN [1], the shape factors of GGD are artificially set and fixed during training. However, as the prediction error values decrease during the training process, the GGD with fixed shape factors can't fit the real prediction error distribution well. Moreover, it's difficult to artificially find the best combination of shape factors for different types of target features in multi-objective training process. Therefore, we propose an update strategy of shape factors utilizing the one-to-one mapping between the kurtosis and shape factor in GGD. We assume that each dimension of the s-th feature prediction error vector is a random variable denoted e_{s,d_s}, then the kurtosis of e_{s,d_s} is defined as:

$$\text{Kurt}[e_{s,d_s}] = \text{E}\left[\left(\frac{e_{s,d_s} - \mu_{s,d_s}}{\sigma_{s,d_s}} \right)^4 \right] = \frac{\text{E}[(e_{s,d_s} - \mu_{s,d_s})^4]}{(\text{E}[(e_{s,d_s} - \mu_{s,d_s})^2])^2} \quad (10)$$

where μ_{s,d_s} and σ_{s,d_s} are the mean and standard deviation of random variable e_{s,d_s}. Meanwhile, the kurtosis of e_{s,d_s} can also be calculated as follows in GGD:

$$\text{Kurt}[e_{s,d_s}] = \frac{\Gamma(5/\beta_{s,d_s})\Gamma(1/\beta_{s,d_s})}{\Gamma(3/\beta_{s,d_s})^2} - 3 \quad (11)$$

In the training process, we calculate the kurtosis of each dimension in prediction error vector by Eq. (10), then we get the value of the new β_{s,d_s} by looking up the table calculated by Eq. (11). In this way, we have implemented an automatic update of the shape factors on each dimension under all types of target features.

3. EXPERIMENTS

3.1. Experimental conditions

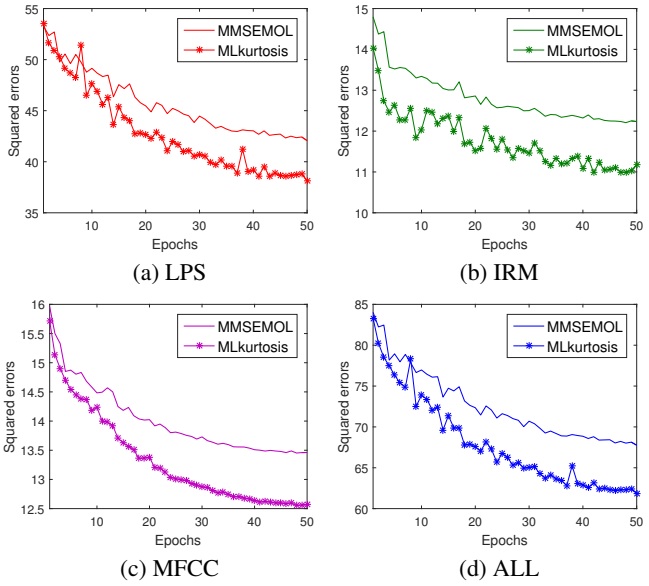
The 115 noise types which included 100 noise types [23] and 15 home-made noise types were adopted for training to improve the robustness to the unseen noise types. The clean speech utterances were derived from the WSJ0 corpus. All 7138 utterances from the training set of WSJ0 corpus were corrupted with the above-mentioned 115 noise types at six levels of SNRs (-5dB, 0dB, 5dB, 10dB, 15dB and 20dB) to build 86-hour multi-condition training set, consisting of pairs of clean and noisy speech utterance. Approximately 400 sentences randomly selected from the 86-hour data set were used as the cross-validation set. The 330 utterances from the core test set of WSJ0 corpus were used to construct the test set for each combination of noise types and SNR levels (-5dB, 0dB, 5dB, 10dB and 15dB). In this experiment, four unseen noise types, namely Pink, Factory1, Destroyerengine and White were adopted for testing. All of them were collected from the NOISEX-92 [24] corpus.

Experiments were conducted on waveforms with 16kHz. The frame length and shift were 256 and 128 samples, respectively. Only the LPS was used in the input layer, and three types of target features (LPS, MFCC and IRM) were used in the output layer. The 257-dimensional feature vector was used for both LPS and IRM. The MFCC used in this experiment had 40 dimensions of static feature and one energy dimension using 40 Mel-filters. Sigmoid was used as the activation function of DNN. DNNs were initialized with random weights. Mean and variance normalization were applied to the input and target feature vectors of the DNN. The DNN configurations were fixed at h=3 hidden layers, 2048 units at each hidden layer, and 7-frame input. We adopted the sigmoid function in the output layer to guarantee the estimated IRM in [0,1], and we used the linear output layer for other types of features. The learning rate for the fine-tuning was set to 0.1 for the first 10 epochs and declined at a rate of 90% after every epoch in the next 40 epochs with the mini-batch size of 128. The enhanced LPS was obtained by a simple average operation between estimated LPS and IRM in LPS domain to fully utilize the complementary targets learned from DNN, just like [22]. We updated the shape factors of ML-MOL-DNN by the strategy in Section 2.3 for every 10 epochs (denoted as MLkurtosis-MOL-DNN) and initialized shape factors were obtained from the well-trained MMSE-MOL-DNN. We also set all shape factors in three types of target features of ML-MOL-DNN to 1 (denoted as ML111-MOL-DNN) and 2 (denoted as ML222-MOL-DNN) as comparison. The enhancement performance was assessed by using PESQ [25] for measuring speech quality, STOI [26] for measuring speech intelligibility, segmental SNR (SSNR in dB) and LSD (in dB) for evaluating signal differences in the time domain and the frequency domain[14], respectively.

3.2. Evaluation on ML-MOL-DNN

Fig.2 illustrates the comparison of learning curves between MMSE-MOL-DNN and MLkurtosis-MOL-DNN using averaged squared errors on the cross-validation set. It is shown that the MLkurtosis-MOL-DNN can achieve better convergence than MMSE-MOL-DNN in all types of target features, which demonstrates the effectiveness of ML-MOL-DNN approach in multi-objective learning.

Table 1 shows the comparison of average prediction error values among MMSE-MOL-DNN, ML222-MOL-DNN, ML111-MOL-DNN and MLkurtosis-MOL-DNN on the cross-validation set in the 50-th epoch which has converged. The prediction errors in MMSE-MOL-DNN and ML-MOL-DNNs were calculated according to Eq. (1) and Eq. (9) respectively, the difference is that Table 1 calculated the error of each type of feature separately. Average operations in dimensions and samples were used for a reasonable comparison. Clearly, the results demonstrate that the ML-MOL-DNNs can better control the dynamic range of prediction error values among different types of target features.



(a) LPS (b) IRM

(c) MFCC (d) ALL

Fig. 2. The comparison of learning curves between MMSE-MOL-DNN (denoted as MMSEMOL) and MLkurtosis-MOL-DNN (denoted as MLkurtosis) in LPS, IRM, MFCC and sum of all features (denoted as ALL) using averaged squared errors on the cross-validation set with respect to the epoch.

Table 1. A comparison of average prediction error values among MMSE-MOL-DNN (denoted as MMSEMOL), ML222-MOL-DNN (denoted as ML222), ML111-MOL-DNN (denoted as ML111) and MLkurtosis-MOL-DNN (denoted as MLkurtosis) in the 50-th epoch

	LPS error	IRM error	MFCC error	max/min
MMSEMOL	0.1637	0.0476	0.3283	6.897
ML222	0.4724	0.4767	0.4980	1.054
ML111	0.9365	0.9509	0.9990	1.067
MLkurtosis	1.1111	1.1060	0.7621	1.458

3.3. Shape factors update results

The change of shape factors after dimension averaging in MLkurtosis-MOL-DNN among different types of features with respect to epoch is shown in Table 2. The experiment results show that the shape factors are gradually becoming smaller during the training process, this means the kurtosis calculated by Eq. (11) becomes larger and the distributions of prediction errors become more centralized, which illustrates the rationality of updating shape factors during training process in MLkurtosis-MOL-DNN.

3.4. Overall comparison

Table 3 compares the PESQ, STOI, SSNR and LSD on the test set of the four unseen noise environments among: MMSE-MOL-DNN, ML222-MOL-DNN, ML111-MOL-DNN and MLkurtosis-MOL-DNN. From this table we can make several observations. First, three ML-MOL-DNNs all yield better average results than MMSE-MOL-DNN except the LSD performance of the ML222-MOL-DNN is worse than that of the MMSE-MOL-DNN. Second, the MLkurtosis-MOL-DNN can achieve consistent improvements on four evaluation metrics over other ML-MOL-DNNs. Finally, unlike ML222-MOL-DNN and ML111-MOL-DNN, the MLkurtosis-MOL-DNN can achieve better SSNR than MMSE-MOL-DNN at all SNR levels.

Table 2. The change of shape factors after dimension averaging in MLkurtosis-MOL-DNN under different types of features (LPS, IRM and MFCC) with respect to the epoch.

Epoch	1 (Init-MMSE)	20	40	50
LPS	0.9365	0.8744	0.8553	0.8542
IRM	1.1854	0.9105	0.8847	0.8732
MFCC	1.3673	1.3404	1.3268	1.3245

Table 3. Performance comparison on the test set at different SNRs among: MMSE-MOL-DNN (denoted as MMSEMOL), ML222-MOL-DNN (denoted as ML222), ML111-MOL-DNN (denoted as ML111) and MLkurtosis-MOL-DNN (denoted as MLkurtosis). **Ave denotes the average of five SNRs (-5dB, 0dB, 5dB, 10dB and 15dB)**.

SNR(dB)		-5	5	15	Ave
PESQ	MMSEMOL	1.650	2.540	3.136	2.463
	ML222	1.762	2.616	3.192	2.544
	ML111	1.834	2.672	3.242	2.603
	MLkurtosis	1.838	2.680	3.253	**2.611**
STOI	MMSEMOL	0.680	0.891	0.968	0.857
	ML222	0.689	0.903	0.974	0.867
	ML111	0.682	0.904	0.976	0.866
	MLkurtosis	0.686	0.905	0.976	**0.868**
SSNR	MMSEMOL	-3.222	0.661	4.966	0.767
	ML222	-3.385	0.689	5.274	0.819
	ML111	-3.341	1.020	5.888	1.149
	MLkurtosis	-3.125	1.179	6.017	**1.316**
LSD	MMSEMOL	6.588	3.380	1.856	3.803
	ML222	6.676	3.570	1.860	3.924
	ML111	6.457	3.432	1.847	3.793
	MLkurtosis	6.087	3.228	1.737	**3.572**

4. CONCLUSION

In this paper, we expand the ML method to multi-objective learning and propose an automatic update strategy for shape factors. On the one hand, compared with the MMSE-MOL-DNN, the ML-MOL-DNNs can adjust the prediction error values under different types of features automatically and achieve performance improvement. On the other hand, compared with the ML-MOL-DNNs with fixed shape factors, the proposed MLkurtosis-MOL-DNN can achieve consistent improvements on four evaluation metrics. In this study, we focus on applying our framework to DNN rather than other complicated architectures such as long short-term memory (LSTM) recurrent neural networks [10] because DNN is easier for deploying especially for speech communication due to its efficiency. Moreover, [22] also shows MMSE-MOL-DNN with ensembling can achieve even better results of different measures over LSTM approach. In the future, we will introduce more types of complementary features and apply the ML-MOL-DNN approach to multi-objective learning and ensembling models with compact neural network architectures.

5. ACKNOWLEDGEMENT

This work was supported in part by the National Key R&D Program of China under contract No. 2017YFB1002202, the National Natural Science Foundation of China under Grants No. 61671422 and U1613211, the Key Science and Technology Project of Anhui Province under Grant No. 17030901005. This work was also funded by Tencent.

6. REFERENCES

[1] Li Chai, Jun Du, Qing-Feng Liu, and Chin-Hui Lee, "Using generalized gaussian distributions to improve regression error modeling for deep learning-based speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1919–1931, 2019.

[2] Jacob Benesty, Shoji Makino, and Jingdong Chen, *Speech enhancement*, Springer Science & Business Media, 2005.

[3] Steven F Boll, "Supression of acoustic noise in speech using spectral subtraction, ieee transaction on assp, vol," *ASSP-2 No*, vol. 2, 1979.

[4] Jae Lim and Alan Oppenheim, "All-pole modeling of degraded speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, 1978.

[5] Yariv Ephraim and David Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE transactions on acoustics, speech, and signal processing*, vol. 33, no. 2, pp. 443–445, 1985.

[6] Yuxuan Wang, Arun Narayanan, and DeLiang Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[7] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

[8] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

[9] Ming Liu, Yujun Wang, Jin Wang, Jing Wang, and Xiang Xie, "Speech enhancement method based on lstm neural network for speech recognition," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 2018, pp. 245–249.

[10] Changyan Zheng, Xiongwei Zhang, Meng Sun, Yibo Xing, and Huawen Shi, "Throat microphone speech enhancement via progressive learning of spectral mapping based on lstm-rnn," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*. IEEE, 2018, pp. 1002–1006.

[11] Ashutosh Pandey and DeLiang Wang, "Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6875–6879.

[12] Szu-Wei Fu, Yu Tsao, and Xugang Lu, "Snr-aware convolutional neural network modeling for speech enhancement.," in *Interspeech*, 2016, pp. 3768–3772.

[13] Phani Sankar Nidadavolu, Jesús Villalba, and Najim Dehak, "Cycle-gans for domain adaptation of acoustic features for speaker recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6206–6210.

[14] Jun Du and Qiang Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[15] Yuxuan Wang, Kun Han, and DeLiang Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 270–279, 2012.

[16] Dongbo Li, Longbiao Wang, Jianwu Dang, Meng Ge, and Haotian Guan, "Distant-talking speech recognition based on multi-objective learning using phase and magnitude-based feature," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018, pp. 394–398.

[17] Yong Xu, Jun Du, Zhen Huang, Li-Rong Dai, and Chin-Hui Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," *arXiv preprint arXiv:1703.07172*, 2017.

[18] Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Monaural speech enhancement using deep neural networks by maximizing a short-time objective intelligibility measure," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5059–5063.

[19] Hiroyuki Manabe and Z Zhang, "Multi-stream hmm for emg-based speech recognition," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2004, vol. 2, pp. 4389–4392.

[20] Juergen Luettin, Gerasimos Potamianos, and Chalapathy Neti, "Asynchronous stream modeling for large vocabulary audio-visual speech recognition," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. IEEE, 2001, vol. 1, pp. 169–172.

[21] John Garofalo, David Graff, Doug Paul, and David Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[22] Qing Wang, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A multiobjective learning and ensembling approach to high-performance speech enhancement with compact neural network architectures," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 7, pp. 1181–1193, 2018.

[23] Guoning Hu, "100 nonspeech environmental sounds," *The Ohio State University, Department of Computer Science and Engineering*, 2004.

[24] Andrew Varga and Herman JM 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 communication*, vol. 12, no. 3, pp. 247–251, 1993.

[25] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. IEEE, 2001, vol. 2, pp. 749–752.

[26] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.