Interspeech 2018
2-6 September 2018, Hyderabad

# Error Modeling via Asymmetric Laplace Distribution for Deep Neural Network Based Single-Channel Speech Enhancement

*Li Chai*[1], *Jun Du*[1], *and Chin-Hui Lee*[2]

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

cl122@mail.ustc.edu.cn, jundu@ustc.edu.cn, chl@ece.gatech.edu

## Abstract

The minimum mean squared error (MMSE) as a conventional training criterion for deep neural network (DNN) based speech enhancement has been found many problems. In our recent work, a maximum likelihood (ML) approach to parameter learning by modeling the prediction error vector as a Gaussian density was proposed. In this study, our preliminary statistical analysis reveals the super-Gaussianity and asymmetricity of the prediction error distribution. Consequently, we adopt the asymmetric Laplace distribution (ALD) instead of the Gaussian distribution (GD) to model the prediction error vectors. Then the new derivation for optimizing the the proposed ML-ALD-DNN with both DNN and ALD parameters is presented. Moreover, we can well interpret the asymmetry parameter of ALD as the balance control between noise reduction and speech preservation from both formulations and experiments. This implies that the customization of DNN models for the different noise types and levels is possible by the setting of the asymmetry parameter. Finally, our ML-ALD-DNN approach achieves better STOI and SSNR measures over both MMSE-DNN and ML-GD-DNN approaches.

**Index Terms**: prediction error modeling, asymmetric Laplace distribution, maximum likelihood, deep neural network, speech enhancement

## 1. Introduction

The main objective of speech enhancement is to improve the performance of speech communication systems or enhance the recognition accuracy of automatic speech recognition systems in noisy environments. Depending on the specific application, the goal of an enhancement system may be to improve the overall quality, increase intelligibility, reduce listener fatigue, or a combination of these [1]. Considering the various complicated situation, the enhancement performance in real acoustic environments is still unsatisfactory and many problems should be solved.

Numerous speech enhancement approaches have been proposed to solve the problems. Among them, traditional algorithms include spectral subtraction [2, 3, 4], Wiener filtering [5, 6], a MMSE estimator [7, 8], an optimally-modified log-spectral amplitude (OM-LSA) speech estimator [9] and other statistical-model-based methods [10, 11]. These conventional methods often fail to track non-stationary noise for real-world scenarios in unexpected acoustic conditions.

Recently, following the successes in speech recognition [12], deep learning techniques were also applied in speech enhancement. Some deep architectures, such as deep neural network (DNN) [13, 14], deep auto encoder (DAE) [15], recurrent neural network (RNN) [16, 17, 18, 19] and convolutional neural network (CNN) [20, 21] were adopted to model the relationship between the noisy signals and the clean speech signals. From the perspective of machine learning, one key challenge of deep learning based speech enhancement is the optimization of the complicated and non-convex objective function. The MMSE between the target features and the predicted features is widely used as an optimization criterion for regression neural network [13]. However, the MMSE-DNN approach is not robust in adverse acoustic environments, e.g., leading to the over-smoothing problem and speech information lost in low signal-to-noise ratio (SNR) conditions [22]. Consequently, better objective function designs attracted a considerable amount of attention recently [16, 17, 21, 23, 24, 25, 26]. Different from these conventional approaches, our recent work [27] investigated the maximum likelihood (ML) solution within the probabilistic learning framework to optimize neural network parameters with the assumption that each dimension of the prediction error vector at the neural network output follows a zero mean Gaussian density. Experiments demonstrated its superiority of better generalization capability and less speech distortions especially in low SNR environments over the MMSE-DNN approach.

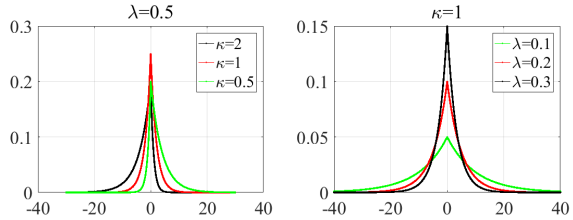


Figure 1: *The ALD PDF family.*

In this study, a further statistical analysis reveals the super-Gaussianity and asymmetricity of the prediction error distribution from MMSE-DNN. Accordingly, we replace the Gaussian distribution (GD) with the asymmetric Laplace distribution (ALD) [28] to well model the prediction error distribution. The probability density function (PDF) of the univariate ALD with zero mean is

$$p(x|\lambda,\kappa) = \frac{\lambda}{\kappa + \frac{1}{\kappa}} \exp\left(-x\,\mathrm{sgn}(x)\lambda\kappa^{\mathrm{sgn}(x)}\right), \quad (1)$$

where $\lambda > 0$ is a scale parameter that plays the role of a variance and $\kappa$ is an asymmetry parameter which measures the skewness and controls the deviation of distribution from symmetry as intuitively shown in Figure 1, where $\kappa = 1$ corresponds to Laplace distribution (LD). Then an alternating two-step optimization scheme is adopted to update both DNN and ALD parameters. For speech enhancement, we will interpret the asymmetry parameter $\kappa$ as the balance control between noise re-



10.21437/Interspeech.2018-1439

duction and speech preservation in the following sections. This implies that the customization of DNN models for the different noise types and levels is possible by the setting of the asymmetry parameter. Experiments show that our ML-ALD-DNN approach achieves better STOI and SSNR measures over both MMSE-DNN [13] and ML-GD-DNN [27] approaches.

## 2. The Proposed ML-ALD-DNN Approach

### 2.1. Statistical analysis of prediction errors

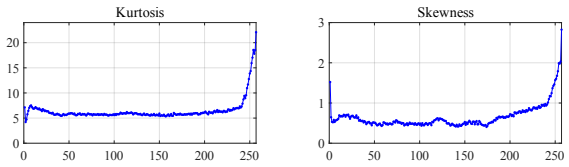


Figure 2: *The kurtosis/skewness of each dimension of the prediction error vector on the cross validation set from the well-trained MMSE-DNN [13].*

As shown in Figure 2, a statistical analysis for the kurtosis of the prediction error vector in each dimension reveals the super-Gaussianity of their distributions based on the fact that all the kurtosises are larger than 3 (the kurtosis of GD). Moreover, Figure 2 also illustrates that the skewness for each dimension is not 0 (the skewness of GD), implying the distribution of each dimension is not symmetric. This is the motivation why we choose ALD to model the prediction errors in this study.

### 2.2. Derivation for ML-ALD-DNN

In conventional MMSE-DNN, a stochastic gradient descent algorithm is performed in mini-batches with multiple epochs to minimize the following error function:

$$E = \frac{1}{N}\sum_{n=1}^{N}\sum_{d=1}^{D}(x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}))^2, \tag{2}$$

where $E$ is the mean squared error, $N$ is the mini-batch size, $D$ is the dimension of log-power spectra (LPS) features, $\boldsymbol{W}$ is the DNN parameter set to be learned. $y_{n,d}$, $\hat{x}_{n,d}$ and $x_{n,d}$ denote the $d$-th dimension of noisy, enhanced and reference normalized LPS features at sample index $n$ respectively. Then the prediction error $e_{n,d}$ could be defined as:

$$e_{n,d} = x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}), \tag{3}$$

which is treated as a random variable following a univariate zero mean ALD with an unrestricted scale parameter $\lambda_d$ and a known asymmetry parameter $\kappa$:

$$p(e_{n,d}|\lambda_d) = \frac{\lambda_d}{\kappa + \frac{1}{\kappa}} \exp\left(-e_{n,d}\,\mathrm{sgn}(e_{n,d})\lambda_d \kappa^{\mathrm{sgn}(e_{n,d})}\right). \tag{4}$$

We also assume that the prediction errors in all dimensions are independently and identically distributed variables. Therefore we can get the joint distribution of the prediction errors for all dimensions at sample index $n$, as represented by the vector $\boldsymbol{e}_n$:

$$p(\boldsymbol{e}_n|\boldsymbol{\lambda}) = \prod_{d=1}^{D} \frac{\lambda_d}{\kappa + \frac{1}{\kappa}} \exp\left(-e_{n,d}\,\mathrm{sgn}(e_{n,d})\lambda_d \kappa^{\mathrm{sgn}(e_{n,d})}\right), \tag{5}$$

where $\boldsymbol{\lambda} = \{\lambda_d | d = 1, 2, ..., D\}$. If the reference vector $\boldsymbol{x}_n$ is also a random variable, then we derive the conditional target distribution (CTD) [29] as a function of the input features $\boldsymbol{y}_n$ with the parameter set $(\boldsymbol{W}, \boldsymbol{\lambda})$:

$$p(\boldsymbol{x}_n|\boldsymbol{y}_n, \boldsymbol{W}, \boldsymbol{\lambda}) = \prod_{d=1}^{D} \frac{\lambda_d}{\kappa + \frac{1}{\kappa}} \exp\left(-(x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}))v_{n,d}\lambda_d \kappa^{v_{n,d}}\right), \tag{6}$$

where $v_{n,d} = \mathrm{sgn}(x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}))$. Given a 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. (6), the corresponding likelihood function is:

$$p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{\lambda}) = \prod_{n=1}^{N}\prod_{d=1}^{D} \frac{\lambda_d}{\kappa + \frac{1}{\kappa}} \exp\left(-(x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}))v_{n,d}\lambda_d \kappa^{v_{n,d}}\right), \tag{7}$$

where the parameter set $(\boldsymbol{W}, \boldsymbol{\lambda})$ is to be optimized. It is equivalent to maximizing the log-likelihood as follows:

$$\ln p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{\lambda}) = N\sum_{d=1}^{D}\ln\lambda_d - ND\ln(\kappa + \frac{1}{\kappa}) - \sum_{n=1}^{N}\sum_{d=1}^{D}(x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}))v_{n,d}\lambda_d\kappa^{v_{n,d}}. \tag{8}$$

### 2.3. The training procedure of ML-ALD-DNN

We design a procedure to alternatively optimize $\boldsymbol{W}$ and $\boldsymbol{\lambda}$ in mini-batch mode as shown in Algorithm 1. To maximize Eq. (8) with respect to $\boldsymbol{\lambda}$, we can derive the update formula:

$$\lambda_d = \frac{N}{\sum_{n=1}^{N}(x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}))v_{n,d}\kappa^{v_{n,d}}}, \tag{9}$$

Alternatively, we can also maximize Eq. (8) with respect to $\boldsymbol{W}$, it is equivalent to minimizing the following expression:

$$\mathcal{L}(\boldsymbol{W}) = \sum_{n=1}^{N}\sum_{d=1}^{D}(x_{n,d} - \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}))v_{n,d}\lambda_d\kappa^{v_{n,d}}. \tag{10}$$

Then the back-propagation procedure is used to optimize $\boldsymbol{W}$. The gradient of $\boldsymbol{W}$ is usually obtained by using the chain rule, where only the gradient of the objective function with respect to the network output needs to be modified accordingly as shown in Eq. (11), whereas all other derivatives are unaffected.

$$\frac{\partial\mathcal{L}(\boldsymbol{W})}{\partial\hat{x}_{n,d}} = \begin{cases} \frac{\lambda_d}{\kappa} & \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}) > x_{n,d} \\ 0 & \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}) = x_{n,d} \\ -\lambda_d\kappa & \hat{x}_{n,d}(y_{n,d}, \boldsymbol{W}) < x_{n,d} \end{cases} \tag{11}$$

**Algorithm 1** Training procedure of ML-ALD-DNN

**Step 1: Initialization**
Initialize the DNN parameter set $\boldsymbol{W}$ randomly.
**Step 2: Alternative optimization in mini-batch mode**
Step 2.1: Fix $\boldsymbol{W}$ and update $\boldsymbol{\lambda}$ via Eq. (9)
Step 2.2: Fix $\boldsymbol{\lambda}$ and update $\boldsymbol{W}$ via Eq. (11)
**Step 3: Go to Step 2 for the next epoch**

Table 1: *Performance comparison on the test set (A: Destroyer engine, B: HF channel, C: Volvo, D: Machine gun).*

| SNR(dB) | Metrics | Noise | MMSE-DNN | ML-GD-DNN | ML-ALD-DNN ($\kappa$) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0.7 | 0.8 | 0.9 | 1 | 1.1 | 1.2 | 1.3 |
| -5 | SSNR(dB) | A | -2.16 | -2.26 | **-2.03** | -2.42 | -2.67 | -2.58 | -2.94 | -3.35 | -3.31 |
| | | B | -3.37 | -4.71 | **-3.01** | -3.81 | -4.20 | -3.82 | -4.80 | -5.14 | -5.22 |
| | | C | 6.92 | 7.80 | 8.73 | 8.89 | 9.04 | 9.23 | **9.31** | 9.27 | 9.23 |
| | | D | 7.13 | 7.94 | 9.67 | 10.10 | 10.26 | 10.48 | **10.66** | 10.56 | 10.63 |
| | STOI (%) | A | 59.4 | 62.3 | 63.2 | 62.9 | 62.9 | 63.2 | 63.6 | **63.7** | 63.4 |
| | | B | 63.4 | 63.2 | 66.4 | 66.3 | 65.2 | **66.8** | 64.3 | 63.8 | 64.1 |
| | | C | 91.6 | 92.5 | 93.3 | 93.4 | 93.5 | 93.6 | **93.8** | **93.8** | **93.8** |
| | | D | 88.7 | 89.6 | 90.3 | 90.3 | **90.7** | 90.2 | 90.4 | 90.4 | 90.4 |
| 5 | SSNR(dB) | A | 2.10 | 2.50 | **2.91** | 2.82 | 2.70 | 2.73 | 2.47 | 2.25 | 2.20 |
| | | B | 1.33 | 1.07 | **1.93** | 1.82 | 1.26 | 1.40 | 1.05 | 0.57 | 0.47 |
| | | C | 9.09 | 10.31 | 11.73 | 12.22 | 12.47 | 12.83 | 12.96 | 12.96 | **12.99** |
| | | D | 8.94 | 9.97 | 11.91 | 12.52 | 12.73 | 13.15 | 13.26 | 13.29 | **13.35** |
| | STOI(%) | A | 84.3 | 85.6 | 85.9 | 85.9 | 86.1 | 86.2 | **86.3** | **86.3** | 86.2 |
| | | B | 84.2 | 84.3 | 85.3 | **85.5** | 84.7 | 85.3 | 84.7 | 84.1 | 84.4 |
| | | C | 95.0 | 95.8 | 96.6 | 96.7 | 96.8 | 96.8 | **96.9** | **96.9** | **96.9** |
| | | D | 92.8 | 93.6 | 94.4 | 94.4 | **94.5** | 94.3 | 94.3 | 94.3 | 94.3 |
| 15 | SSNR(dB) | A | 6.47 | 7.12 | 7.97 | 8.09 | 8.11 | **8.17** | 7.99 | 7.92 | 7.78 |
| | | B | 6.04 | 6.60 | 7.35 | **7.59** | 7.21 | 7.35 | 7.36 | 6.96 | 6.87 |
| | | C | 10.65 | 12.07 | 14.05 | 14.86 | 15.28 | 15.83 | 15.99 | 16.14 | **16.20** |
| | | D | 10.52 | 11.74 | 13.89 | 14.67 | 14.98 | 15.53 | 15.65 | 15.79 | **15.87** |
| | STOI(%) | A | 94.4 | 95.1 | 95.4 | 95.6 | 95.7 | 95.9 | 95.9 | **96.0** | 95.9 |
| | | B | 94.0 | 94.6 | 94.8 | 95.1 | 95.0 | **95.2** | **95.2** | 95.1 | **95.2** |
| | | C | 96.6 | 97.3 | 98.2 | 98.3 | 98.4 | 98.5 | 98.5 | 98.5 | **98.6** |
| | | D | 95.5 | 96.1 | 96.9 | 96.9 | 97.0 | **97.0** | 96.9 | 96.9 | 96.9 |

# 3. Experiments

## 3.1. Experimental conditions

Experiments were conducted on waveforms with 16kHz. The 115 noise types which included 100 noise types [30] and 15 home-made noise types were adopted for training to improve the robustness to the unseen noise types. The 4620 utterances from the training set of the TIMIT corpus were corrupted with the above-mentioned 115 noise types at six levels of SNRs (-5dB, 0dB, 5dB, 10dB, 15dB and 20dB) to build a 80-hour training set, consisting of pairs of clean and noisy speech utterances. The 192 utterances from the core test set of TIMIT database were used to construct the test set for each combination of noise types and SNR levels. In this experiment, four unseen noise types (Destroyer engine, HF channel, Volvo, Machine gun) which were all collected from the NOISEX-92 corpus [31] were adopted for testing.

A short-time Fourier transform was used to compute the spectra of each overlapping windowed frame. Then 257-dimensional ($D = 257$) LPS features were used to train DNNs. Mean and variance normalization was applied to the input and reference feature vectors of the DNN. Sigmoid was used as the activation function of DNN. All DNN configurations were fixed at three hidden layers, 2048 units for each hidden layer, and 7-frame input. DNNs were initialized with random weights. The learning rate for the supervised 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 ($N = 128$). Original phase of noisy speech was adopted with the enhanced LPS for the waveform reconstruction. Segmental SNR (SSNR in dB) [13] for measuring noise reduction and short-time objective intelligibility (STOI in %) [32] for measuring speech intelligibility were used to evaluate performance.

## 3.2. Experimental results and analysis

In spite of the advantage of the ML-GD-DNN approach [27] in less speech distortions over the conventional MMSE criterion, there is an unavoidable problem that it may introduce more noises correspondingly. For some noise types (e.g. HF Channel) which are extremely difficult to remove, our ML-GD-DNN approach may not bring enhancement performance improvements in very low SNR environments (e.g., -5dB) since less speech distortions can not make up more noise preservation. However, for other noise types which are easy to remove, the ML-GD-DNN approach could work across the global SNR levels. These can be observed in Table 1. But our proposed ML-ALD-DNN approach can bring further improvements over ML-GD-DNN and it can even yield significant improvements when the ML-GD-DNN approach dose not outperform MMSE-DNN, e.g., the ML-ALD-DNN ($\kappa = 1$) system improves STOI by 3.4% for HF channel noise over the MMSE-DNN system at -5dB where the ML-GD-DNN system does not bring improvements. This implies that our ML-ALD-DNN ($\kappa = 1$) approach can remove more noise and lead to less speech distortions compared with the ML-GD-DNN approach. From the comparison of spectrograms in Figure 3, we can further observe that the proposed ML-ALD-DNN ($\kappa = 1$) approach achieves less speech distortions and more noise reduction over the ML-GD-DNN approach.

Please note that the asymmetry parameter $\kappa$ controls the noise reduction and speech distortions, i.e., the smaller $\kappa$ is, the more noise reduction and speech distortions the ML-ALD-DNN system of $\kappa < 1$ achieves and the larger $\kappa$ is, the less noise reduction and speech distortions the ML-ALD-DNN system of $\kappa > 1$ achieves. For example in Table 1, ML-ALD-DNN ($\kappa = 0.7$) yields the best SSNR results for both Destroyer engine and HF channel noises at -5dB and 5dB SNRs. Also Figure 3 is

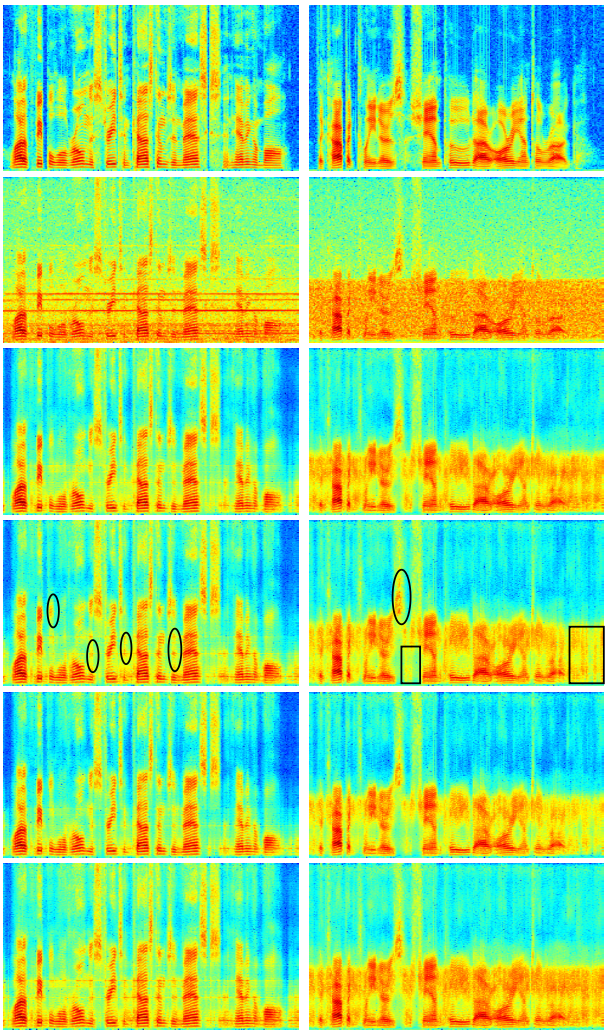

Figure 3: *Comparison of spectrograms of two 16kHz TIMIT utterances corrupted by Destroyer engine (left column) and HF channel (right column) noise at 5dB respectively (from top to bottom): clean speech, noisy speech, ML-GD-DNN enhanced speech, ML-ALD-DNN ($\kappa = 1$) enhanced speech, ML-ALD-DNN ($\kappa = 0.7$) enhanced speech, ML-ALD-DNN ($\kappa = 1.3$) enhanced speech.*

a more intuitive illustration. Since noise retention is small in high SNR environment or for the noise types easy to remove, the best performance is usually achieved for ML-ALD-DNN approach when $\kappa > 1$ in this condition due to its superiority of less speech distortions. This could be also observed from Table 1, where the best SSNR and STOI results for Volvo noise at 15dB are achieved when $\kappa = 1.3$.

Figure 4 shows the distributions for selected dimensions (1, 60, 150, 240) of the prediction error vector which is calculated by subtracting the reference feature vector from the enhanced feature vector using well-trained ML-ALD-DNN on the cross validation set. We can see that the histograms move right consistently as $\kappa$ becomes larger. This implies that the number of bins where the enhanced features are smaller than the reference features becomes less and less as $\kappa$ becomes larger and larger. Correspondingly, the number of bins where the enhanced features are larger than the reference features becomes more and

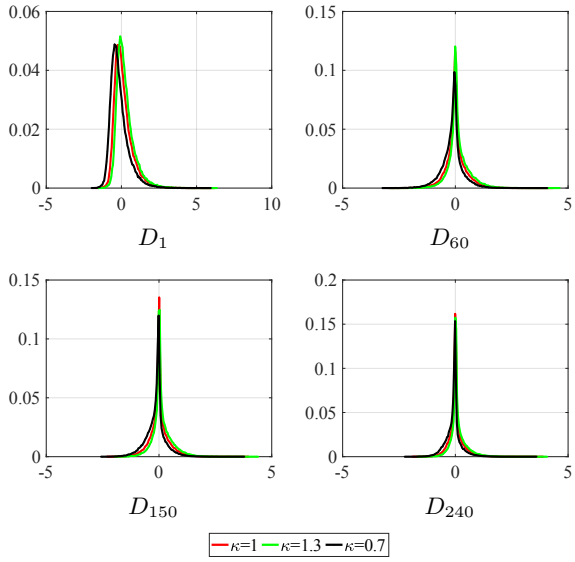


Figure 4: *The distributions for selected dimensions of the prediction error vector from well-trained ML-ALD-DNN on the cross validation set.*

more as $\kappa$ becomes larger and larger. This well explains for the above-mentioned experimental results that the ML-ALD-DNN system tends to do less harm to speech and preserve more noise when $\kappa$ becomes large. Another explanation is as follows. From Eq. (11), please note that the penalty for clipping off a speech segment is the same as the penalty for clipping off noise when $\kappa = 1$. Compared with this, the objective function assigns higher penalty against noise preservation and lower penalty against speech removal when $\kappa < 1$, and assigns higher penalty against speech removal and lower penalty against noise preservation when $\kappa > 1$. Consequently, $\kappa$ has an effect on the noise removal and speech distortions and we can potentially choose the optimal value based on different scenarios.

## 4. Conclusion

In this paper, we replace GD with ALD to model the prediction error at the DNN output in ML framework. Statistical analysis shows the reasonability of the assumption that the prediction error vector of SE-DNN follows the ALD. Moreover, experiments demonstrate the superiority of our ML-ALD-DNN approach in better generalization capability and robustness. The proposed ML-ALD-DNN can achieve less speech distortions and larger noise reduction over ML-GD-DNN approach. Furthermore, the asymmetry parameter of ALD can control the balance between noise reduction and speech preservation, which implies that the customization of DNN models for the different noise types and levels is possible by the setting of the asymmetry parameter.

## 5. Acknowledgment

This work was supported partly 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, and MOE-Microsoft Key Laboratory of USTC. This work was also funded by Huawei Noah's Ark Lab.

# 6. References


[1] R. Le Bouquin, "Enhancement of noisy speech signals: Application to mobile radio communications," *Speech Communication*, vol. 18, no. 1, pp. 3–19, 1996.

[2] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[3] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'79.*, vol. 4. IEEE, 1979, pp. 208–211.

[4] K. Paliwal, K. Wójcicki, and B. Schwerin, "Single-channel speech enhancement using spectral subtraction in the short-time modulation domain," *Speech communication*, vol. 52, no. 5, pp. 450–475, 2010.

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

[6] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[7] Y. Ephraim and D. 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.

[8] T. Gerkmann and R. C. Hendriks, "Unbiased mmse-based noise power estimation with low complexity and low tracking delay," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1383–1393, 2012.

[9] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Transactions on speech and audio processing*, vol. 11, no. 5, pp. 466–475, 2003.

[10] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, 1992.

[11] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 2, pp. 137–145, 1980.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[13] Y. Xu, J. Du, L.-R. Dai, and C.-H. 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.

[14] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[15] M. Sun, X. Zhang, T. F. Zheng *et al.*, "Unseen noise estimation using separable deep auto encoder for speech enhancement," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 1, pp. 93–104, 2016.

[16] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1562–1566.

[17] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*. IEEE, 2014, pp. 577–581.

[18] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks." in *ISMIR*, 2014, pp. 477–482.

[19] ——, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.

[20] S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks," *arXiv preprint arXiv:1703.02205*, 2017.

[21] S.-W. Fu, Y. Tsao, and X. Lu, "Snr-aware convolutional neural network modeling for speech enhancement." in *Interspeech*, 2016, pp. 3768–3772.

[22] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee, "A unified dnn approach to speaker-dependent simultaneous speech enhancement and speech separation in low snr environments," *Speech Communication*, vol. 95, pp. 28–39, 2017.

[23] W. Han, X. Zhang, M. Sun, L. Li, and W. Shi, "An improved supervised speech separation method based on perceptual weighted deep recurrent neural networks," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 100, no. 2, pp. 718–721, 2017.

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

[25] K. Zhen, A. Sivaraman, J. Sung, and M. Kim, "On psychoacoustically weighted cost functions towards resource-efficient deep neural networks for speech denoising," *arXiv preprint arXiv:1801.09774*, 2018.

[26] W. Han, X. Zhang, G. Min, X. Zhou, and W. Zhang, "Perceptual weighting deep neural networks for single-channel speech enhancement," in *Intelligent Control and Automation (WCICA), 2016 12th World Congress on*. IEEE, 2016, pp. 446–450.

[27] L. Chai, J. Du, and Y.-n. Wang, "Gaussian density guided deep neural network for single-channel speech enhancement," in *Machine Learning for Signal Processing (MLSP), 2017 IEEE 27th International Workshop on*. IEEE, 2017, pp. 1–6.

[28] T. J. Kozubowski and K. Podgorski, "Asymmetric laplace distributions," *Mathematical Scientist*, vol. 25, no. 1, pp. 37–46, 2000.

[29] D. A. Nix and A. S. Weigend, "Learning local error bars for nonlinear regression," in *Advances in neural information processing systems*, 1995, pp. 489–496.

[30] G. Hu, "100 nonspeech environmental sounds,[online] available: http://web. cse. ohio-state. edu/pnl/corpus/hunonspeech," *HuCorpus. html*, 2004.

[31] 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 communication*, vol. 12, no. 3, pp. 247–251, 1993.

[32] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. 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.