Detection of infeasible input in speech enhancement with deep complex U-Net

Satoru Emura
Kyoto university of advanced science
Kyoto, Japan
0000-0002-3102-6508

***Abstract*—In this study, the possibility of detecting infeasible inputs for speech enhancement using deep complex U-Net (DCUNet) is explored because the effectiveness of neural-network based speech enhancement depends on the coverage of the training dataset. This study focuses on the distribution of points in a feature space in DCUNet and aims to detect infeasible inputs using the Kullback-Leibler divergence between the point distribution of the training dataset and that of a test sample in the feature space modeled as a mixture of Laplacians and a Laplacian, respectively.**

***Index Terms*—speech enhancement, DCUNet, batch normalization, Kullback-Leibler divergence**

I. Introduction

Speech enhancement (SE) aims to recover clean speech from recordings that are compromised by acoustic noise [1] [2]. This process leverages different statistical properties of the target speech and interference signals [3]. Machine-learning methods extract these properties by learning from large datasets using discriminative or generative approaches. The discriminative approach learns the direct mapping from noisy speech to clean speech [4]. The generative approach learns prior knowledge of the speech to separate clean speech from noise [5]. A discriminative approach known as deep complex U-Net (DCUNet) [6] provides effective complex-valued masking and drastically improves the output quality. Neural network (NN) based SE methods have made further progress in response to the international challenges [7].

The effectiveness of SE depends on various factors, such as the signal-to-noise ratio (SNR) and noise type [8]. NN-based approaches depend on the training dataset. These methods are highly effective for in-domain (ID) samples covered by a training dataset; however, they tend to be less effective for uncovered out-of-domain (OOD) samples. Hence, when an NN-based SE method is deployed in an actual situation in which no corresponding clean speech is available, it is beneficial to judge whether noisy speech input is ID. This enables us to detect an infeasible noisy speech sample, which is insufficient because the sample is OOD or ID but learned inadequately. Gathering and analyzing such samples enable us to build up the training dataset by widening its variety and to further improve the NN-based model further by retraining.

The detection of OOD samples in classification problems has been rigorously studied [9] [10] [11]. One approach is to measure the predictive uncertainty by modeling the distribution of data features. Hendrycks et al. [12] proposed using the maximum softmax probability as the confidence score. ODIN [13] uses temperature scaling and input perturbation to amplify the ID/OOD separability of the softmax probability. Lee et al. [14] proposed modeling the input to the softmax layer as class-conditional probabilistic distribution functions of Gaussian densities for ID samples. Another approach is ensemble deep NNs [15] [16], in which the outputs of multiple individually trained NNs or statistical NNs are combined to estimate the uncertainty.

Batch normalization (BN) layers have been investigated from the perspectives of domain shift [17] and OOD detection [18] [19]. Based on the hypothesis that the domain shift from the training dataset to the target dataset is reflected in the statistics estimated at the BN layer, Li et al. [17] demonstrated that adapting only batch statistics is effective for domain adaptation. Using mismatched statistics at the BN layer was shown to be a cause of the overconfidence issue in OOD [20].

However, conventional OOD detection methods are not directly applicable to SE. In the first approach, the softmax layer is not included in the NNs for SE. In the second ensemble approach, several stochastic NNs can be used for SE. However, it has been reported that even the latest diffusion-based SE method [8] has difficulty in estimating its performance to unmatched samples properly [21]. In addition, in SE, it is common for an input signal to be divided into frames and then processed. An input signal is mapped to points in the feature space, and not to a point. Hence, the OOD detection of a distribution of points is required.

This study focuses on the distribution of points in the feature space after BN in DCUNet and aims to detect infeasible samples using the Kullback-Leibler (KL) divergence between the point distribution of the training dataset and that of a test sample inspired by [22] as shown in Fig. 1. In this case, infeasible samples correspond to samples that are not covered by the training dataset or samples that are covered but not learned effectively. Specifically, the feature space in the next-to-last decoder of DCUNet is used. The point distribution is modeled by a single multivariate Laplacian distribution or a mixture of such distributions.

Although DCUNet is not a state-of-the-art method in the context of the international challenges [7], this study focuses on DCUNet for following reasons: 1) DCUNet represents an important milestone because it enables the effective handling of complex-domain representations and drastically improves the output speech quality. 2) The complex-domain convolutional encoder-decoder structure with skip connections in DCUNet is simple yet the basis building block of many other DNN methods. This structure plays an important role when

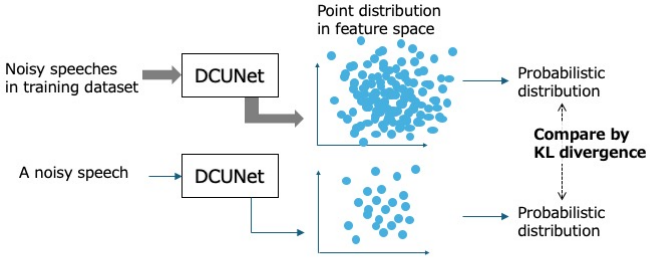


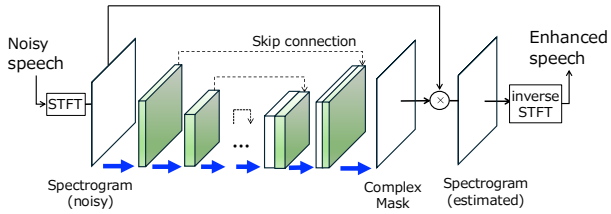
Fig. 1. Overview of proposed method

combined with a recurrent NN [23] and a diffusion network [8] in recent SE methods.

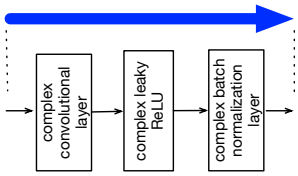
II. Previous Studies

A. DCUNet

DCUNet [6] is an extension of the U-Net structure [24], which comprises convolutional encoders and decoders with skip connections. DCUNet refines U-Net by explicitly handling complex-domain operations using complex building blocks [25]. Figure 2(a) shows the manner in which skip connections combine multi-channel features in DCUNet, where each thick blue arrow corresponds to the operations performed in each building block. The building block comprises a complex convolutional layer, complex BN, and complex leaky ReLU, as shown in Fig. 2(b), except for the final decoder, in which complex BN is omitted.



(a)



(b)

Fig. 2. Structure of DCUNet: (a) In encoding, the frequency information is reduced while feature information is increased. Skip connection enables to combine the low-resolution feature from previous decoder and high-resolution feature from corresponding encoder. Each arrow corresponds to DCUNet's complex building block. (b) inside a building block of DCUNet.

B. BN

The BN layer was originally designed to alleviate internal covariate shifting during training [26]. BN improves the convergence speed of learning and facilitates a regularizing effect. Let us consider the case in which the inputs to a BN layer are a set of convolutional NN features X. Its shape is expressed as (K, C, F, T), where K is the number of samples per batch, C is the number of channels, and $F \times T$ matrices correspond to the spectrogram. Furthermore, F and T indicate the numbers of frequencies and frames, respectively (i.e., the height and width of the spectrogram, respectively). The BN layer normalizes $\boldsymbol{X}[k, 0{:}C, 0{:}F, 0{:}T] \in \mathbb{R}^{1\times C\times F\times T}$ using per-channel statistics, as follows:

$$\hat{X}[k, c, :, :] = \frac{X[k, c, :, :] - \mathbb{E}\{X[:, c, :, :]\}}{\mathrm{Std}[X[:, c, :, :] + \epsilon}, \tag{1}$$

where $\mathbb{E}\{\ \}$ and $\mathrm{Std}\{\ \}$ denotes the expectation and standard deviation, respectively, and ϵ is a regularization term. The ranges of the indices are specified using Python style. The BN outputs the following:

$$Y[k, c, :, :] = \gamma_c \hat{X}[k, c, :, :] + \beta_c, \tag{2}$$

where γ_c and β_c are parameters to be learned in the BN layer.

During training, the empirical mean and standard deviation vectors of a mini-batch are used. Furthermore, during inference, population statistics estimated from the entire training dataset were used in the original study [26] and were reported to be better than the statistics estimated using the exponential moving average (EMA) [27], although the use of the EMA remains popular.

III. Feature Space and Its Distribution Statistics

This study focuses on the point distribution in a feature space in DCUNet and aims to detect infeasible inputs using the KL divergence between the point distributions of the training dataset and a test sample. Specifically, the feature space after BN in the next-to-last decoder of DCUNet is used. The point distribution is modeled by a single multivariate Laplacian distribution or a mixture of such distributions.

The next-to-last decoder is selected based on the analogy with the case of classification tasks, where the last softmax layer or the input to this layer is the focus. In the case of DCUNet, the focus is on the output of the next-to-last decoder used in the last decoder layer.

The proposal of this study consists of the following three parts:

- Step A: pre-processing of the training dataset and pre-trained NN before deployment,
- Step B: processing of a noisy input when deployed,
- Step C: evaluation of the proposed method (Steps A and B) using clean oracle speech.

Steps A and B involve mapping from a noisy input to points in the feature space and fitting the point distribution to a probabilistic distribution.

Step A

Mapping: Let the output of the BN layer correspond to the noisy input sample n $(0 \leq n < N)$ be $\hat{\boldsymbol{X}}_n \doteq \hat{\boldsymbol{X}}[n, 0{:}C, 0{:}F, 0{:}T] \in \mathbb{C}^{1\times C\times F\times T}$. As its summary statistics, the vectors of the feature-wise means and standard deviations $\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n \in \mathbb{C}^C$ are expressed as

$$\boldsymbol{\mu}_n = \left[\ \mu_n[0] \quad \cdots \quad \mu_n[C-1]\ \right], \tag{3}$$

$$\boldsymbol{\sigma}_n = \left[\ \sigma_n[0] \quad \cdots \quad \sigma_n[C-1]\ \right], \tag{4}$$

where

$$\mu_n[c] = \mathbb{E}\left\{\hat{\boldsymbol{X}}[n, c, 0{:}F, 0{:}T]\right\}, \tag{5}$$

$$\sigma_n[c] = \mathrm{Std}\left\{\hat{\boldsymbol{X}}[n, c, 0{:}F, 0{:}T]\right\}. \tag{6}$$

$\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n \in \mathbb{C}^{CB}$ are computed from $F \times T$ points.

This study further considers dividing the axis of frequency F into B blocks to improve the representation ability of the summary statistics inspired by [28] [29]. $\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n \in \mathbb{C}^{CB}$ are computed from $F \times T/B$ points and expressed as

$$\boldsymbol{\mu}_n = [\mu_n[0,0] \cdots \mu_n[C{-}1,0]\ \mu_n[0,1] \cdots\cdots \mu_n[C{-}1,B{-}1]], \tag{7}$$

$$\boldsymbol{\sigma}_n = [\sigma_n[0,0] \cdots \sigma_n[C{-}1,0]\ \sigma_n[0,1] \cdots\cdots \sigma_n[C{-}1,B{-}1]], \tag{8}$$

where

$$\mu_n[c,b] = \mathbb{E}\left\{\hat{\boldsymbol{X}}[n, c, \frac{b}{B}F{:}\frac{b+1}{B}F, 0{:}T]\right\}, \tag{9}$$

$$\sigma_n[c,b] = \mathrm{Std}\left\{\hat{\boldsymbol{X}}[n, c, \frac{b}{B}F{:}\frac{b+1}{B}F, 0{:}T]\right\}. \tag{10}$$

Fitting: In this study, the M-cluster model is applied to the point distribution of the training dataset because it has a higher representation ability and can handle complex point distributions. When $M = 1$, the vectors of the means and standard deviations of the cluster are computed straightforwardly from $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n$ $(0 \le n < N)$. When $M > 1$, we consider N points expressed by the vectors $\boldsymbol{p}_n = [\boldsymbol{\mu}_n\ \boldsymbol{\sigma}_n]$. The k-means++ algorithm [30] [31] is applied to obtain M clusters from these N points. The summary statistics of the m-th cluster $(1 \le m \le M)$ are computed as

$$\boldsymbol{\mu}^{(m)} = \mathbb{E}\left\{\boldsymbol{\mu}_{\tilde{n}}\right\}, \tag{11}$$

$$\boldsymbol{\sigma}^{(m)} = \mathbb{E}\left\{\boldsymbol{\sigma}_{\tilde{n}}\right\} \tag{12}$$

where $\tilde{n} \in$ cluster m. A multivariate Laplacian distribution with a feature-wise mean $\boldsymbol{\mu}^{(m)}$ and scale $\boldsymbol{\sigma}^{(m)}/\sqrt{2}$ is assigned to the cluster m because the use of the Laplacian distribution is expected to lead to more robust results than that of the Gaussian distribution. The threshold of the KL divergence d_{th} is set to a certain value.

Step B:

A noisy input is mapped to a point distribution in the feature space, as in Step A. The feature-wise mean and standard deviation are computed and fitted to a multivariate Laplacian distribution, as in Step A. The KL divergence [32] between the above Laplacian distribution and the closest one of the training dataset [14] is computed using the summary statistics. When the KL divergence $> d_{th}$, the sample is estimated to be infeasible.

Step C:

Instead of selecting a single threshold d_{th}, the standard evaluation method for OOD detection uses a set of different thresholds and compares the resulting performances [31]. We can run the detector for a set of thresholds and plot the true positive rate (TPR) vs. the false positive rate (FPR) as an implicit function of d_{th}. This is known as the receiver operating characteristic (ROC) curve. Its quality is summarized using the area under the curve (AUC). Higher AUC scores are better with a maximum of 1. To apply the aforementioned standard method, the feasible samples are defined using the training dataset.

Training data: Samples in the training dataset are divided into segments according to the input SNR because the improvement in the scale-invariant signal-to-distortion ratio (SI-SDR) [33] tends to be larger for a lower SNR. In each segment, the lower limit of the expected SI-SDR improvement is set to $\mu - \sigma$, where μ and σ are the mean and standard deviation of the SI-SDR improvement in that segment, respectively. Samples satisfying the SI-SDR improvement $\ge \mu - \sigma$ are classified as feasible. The reason for setting the lower limit is that samples that are inadequately learned by the NN in the training dataset remain.

Test data: Noisy speech input in matched and unmatched test datasets is classified as feasible or infeasible based on its input SNR, expected SI-SDR improvement, and actual SI-SDR improvement. This classification result is used to evaluate the estimate based on the KL divergence. The TPR and FPR are obtained by aggregating comparisons. The AUC is obtained by plotting the ROC curve.

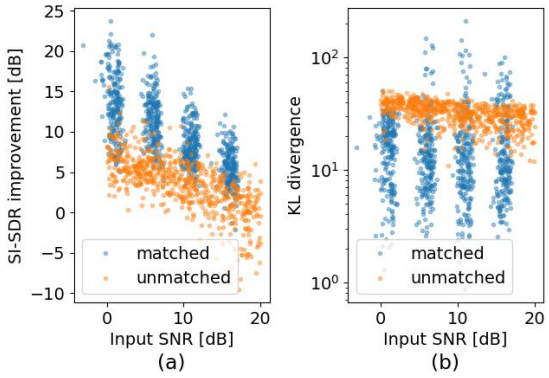


Fig. 3. Dependency of (a) SI-SDR improvement of each sample on input SNR and (b) KL divergence of each sample on input SNR for DCUNet-V. The blue dots indicate the samples from the matched dataset and the orange dots indicate those from the unmatched dataset.

IV. Evaluation

A. Settings

In this study, DCUNet-16 [6] was used as the SE method. DCUNet-16 uses eight building blocks for encoding and eight blocks for decoding. We used a 1024-point short-term Fourier transform (STFT) with a 256-point shift. DCUNet was trained using the WSJ0-CHiME3 and VB-DMD training datasets. The WSJ0-CHiME3 dataset combines clean speech utterances from the Wall Street Journal (WSJ0) dataset [34] with noise signals from the CHiME3 dataset [35]. Similarly, the VB-DMD dataset, which is derived from the publicly available VoiceBank-DEMAND collection [36], is a widely recognized benchmark for evaluating single-channel SE techniques. The input SNR was divided into four segments: 0∼5 dB, 5∼10

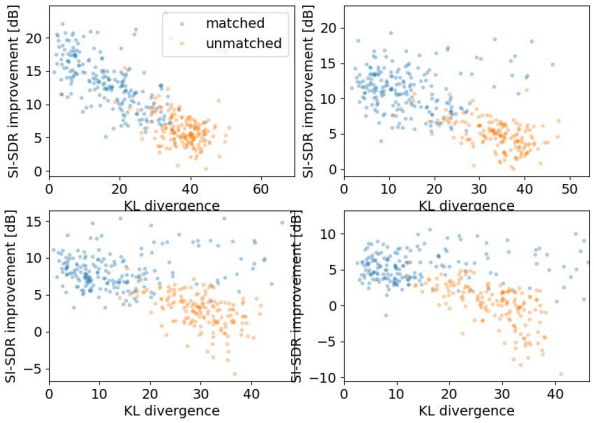


Fig. 4. Relation between KL divergence and SI-SDR improvement of DCUNet-V with $M = 1$ and $B = 2$ in four input SNR segments.

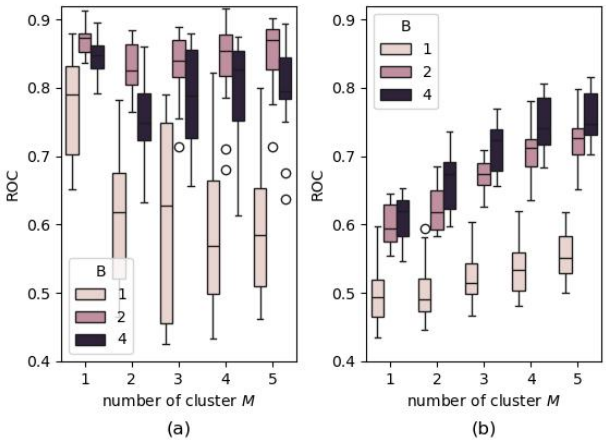


Fig. 5. Variance of AUC on training dataset, number of clusters M, and number of frequency division B: (a) DCUNet-V, (b) DCUNet-W.

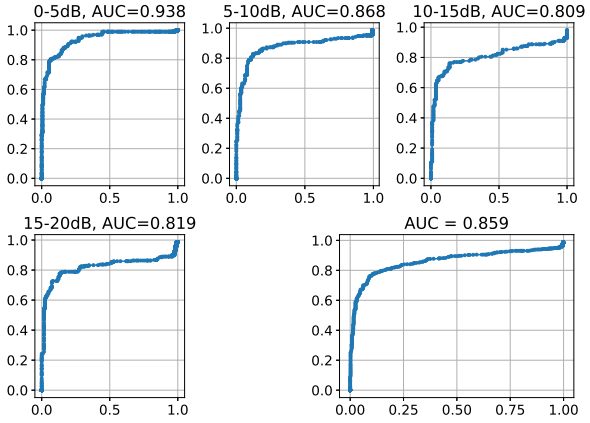


Fig. 6. Typical ROC curves and AUCs of DCUNet-V with $M = 1$ and $B = 2$.

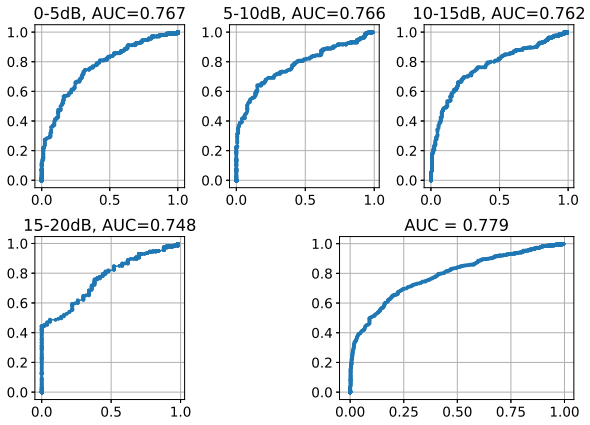


Fig. 7. Typical ROC curves and AUCs of DCUNet-W with $M = 5$ and $B = 4$.

dB, 10∼15 dB, and 15∼20 dB. The training datasets were divided accordingly and the expected SI-SDR improvements were defined for each segment. Hereafter, DCUNet-16 trained with VB-DMD and WSJ0-CHiME3 are denoted as DCUNet-V and DCUNet-W, respectively. Both DCUNet-V and DCUNet-W were tested using the test datasets VB-DMD and WSJ0-CHiME3. The proposed method has two parameters: the number of clusters M (=1 $\sim$ 5) and the number of frequency divisions B (= 1, 2, 4).

B. Experiments

First, the properties of the KL divergence in the matched and unmatched datasets were evaluated. Figure 3 shows the scatter plots of DCUNet-V (a) between the input SNR and SI-SDR improvement, and (b) between the input SNR and KL divergence, where $M = 1$ and $B = 2$. As shown in Fig. 3, the SI-SDR improvement was dependent on the input SNR. Matched and unmatched samples were not completely separated but partially overlapped along the axis of the SI-SDR improvement. The samples from the unmatched dataset tended to have a lower SI-SDR improvement and higher KL divergence than those from the matched dataset.

Figure 4 shows the relationship between the KL divergence and SI-SDR improvement for DCUNet-V with $M = 1$ and $B = 2$. The samples from the matched and unmatched datasets formed two clusters. The unmatched cluster exhibited a lower SI-SDR improvement and higher KL divergence. This suggests that the KL divergence may be a key to detecting infeasible samples.

Next, the proposed method was evaluated using the AUC, where the union of the test datasets of VB-DMD and WSJ0-CHiME3 were used. The AUC is dependent on the training dataset, number of clusters, and frequency division, which are given as the parameters of the proposed method, and the variance of the coefficients of DCUNet. To observe the variance of the AUC, NN coefficients that were trained independently four times up to 200 epochs were used. The proposed method was applied to the NN coefficients for each training run at epochs= 120, 140, 160, 180, and 200. The results are shown in Fig. 5. For DCUNet-V, the AUC was best at $M = 1$ and

$B = 2$. For DCUNet-W, increasing the number of clusters M resulted in a better AUC. An increase in B also resulted in a better AUC. As shown in Fig. 5, the point distribution in the feature space of DCUNet-V was sufficiently modeled using a single multivariate Laplacian distribution. In contrast, the point distribution of DCUNet-W appeared to be more complex and required more clusters. This suggests that training with different datasets leads to different point distributions in the feature space and that a more complex probabilistic model is necessary for DCUNet-W.

Finally, Figs. 6 and 7 show the typical best ROC curves of DCUNet-V with $M = 1$ and $B = 2$ and DCUNet-W with $M = 5$ and $B = 4$. The AUCs of DCUNet-V and DCUNet-W were 0.859 and 0.779, respectively.

V. Conclusion

This study has proposed a method for detecting infeasible inputs to DCUNet. The method focuses on the point distribution in the feature space of the next-to-last decoder of DCUNet, models the point distribution as a mixture of multivariate Laplacians for the training dataset and a multivariate Laplacian for a test sample, and detects infeasible inputs using the KL divergence. DCUNet trained with VB-DMD yielded an AUC of 0.859 when the point distribution of the training dataset was modeled by one cluster. That with WSJ0-CHiME3 yielded an AUC of 0.779 when the point distribution of the training dataset was modeled using five clusters.

References

[1] R. C. Hendriks, T. Gerkmann, and J. Jensen, *DFT-Domain Based Single Microphone Noise Reduction for Speech Enhancement: A Survey of the State-of-the-Art*, Morgan & Claypool, San Rafael, CA, USA, 2013.

[2] C. Zheng, H. Zhang, W. Liu, X. Luo, A. Li, X. Li, and B. C. J. Moore, "Sixty years of frequency-domain monaural speech enhancement: From traditional to deep learning methods," *Trends in Hearing*, vol. 27, Jan. 2023.

[3] T. Gerkmann and E. Vincent, *Spectral masking and filtering*, Wiley, Hoboken, NJ, USA, 2018.

[4] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. on Audio, Speech, and Lang. Proc.*, vol. 26, no. 10, pp. 1702–1726, 2018.

[5] Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 716–720.

[6] H-S Choi, J-H Kim, J. Huh, A. Kim, J-W Ha, and K. Lee, "Phase-aware speech enhancement with deep complex U-net," in *International Conference on Learning Representations (ICLR)*, 2019.

[7] D. Harishchandra, A. Ashkan, G. Vishak, N. Babak, B. Sebastian, C. Ross, J. Alex, Z. Mehdi, T. Min, G. Hannes, G. Mehrsa, and A. Robert, "ICASSP 2023 deep speech enhancement challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[8] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann, "Speech enhancement and dereverberation with diffusion-based generative models," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 2023.

[9] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterch, and K. R. Mueller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.

[10] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. H. Rohban, and M. Sabokrou, "A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: solutions and future challenges," in *Transaction on machine Learning*, 2022.

[11] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv: 2110.11334*, 2021.

[12] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[13] S. Liang, Y. Li, and R Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[14] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *NeurIPS*, 2018, pp. 7167–7177.

[15] Y. Wen, D. Tran, and J. Ba, "BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning," in *International Conference on Learning Representations (ICLR)*, 2020.

[16] X. Chen, Y. Li, and Y. Yang, "Batch-ensemble stochastic neural networks for out-of-distribution detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[17] Y. Li, N. Wang, J. Shi, J. Lu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," in *International Conference on Learning Representations (ICLR)*, 2017.

[18] Y. Zhu, Y. Chen, C. Xie, X. Li, R. Zhang, H. Xue, X. Tian, B. Zheng, and Y. Chen, "Boosting out-of-distribution detection with typical features," in *NeurIPS*, 2022.

[19] A. Li, C. Qiu, M. Kloft, P. Smyth, M. Rudolph, and S. Mandt, "Zero-shot anomaly detection via batch normalization," in *NeurIPS*, 2023.

[20] Y. Sun, C. Guo, and Y. Li, "ReAct: out-of-distribution detection with rectified activations," in *NeurIPS*, 2021.

[21] S. Emura, "Estimation of output SI-SDR solely from enhanced speech signals in diffusion-based generative speech enhancement method," in *European Signal Processing Conference (EUSIPCO)*, 2024.

[22] Y. Zhang, J. Pan, W. Liu, Z. Chen, K. Li, and J. Wang, "Kullback-Leibler divergence-based out-of-distribution detection with flow-based generative models," *IEEE Trans. on Knowledge and Data Engineering*, vol. 36, no. 4, pp. 1683–1697, April 2024.

[23] S. Zhao, B. Ma, K. N. Watcharasupat, and W. S. Gan, "FRCRN: Boosting feature representation using frequency recurrence for monaural speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[25] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[27] Y. Wu and J. Johnson, *Rethinking "Batch" in BatchNorm*, arXiv:2105.07576, 2021.

[28] Y. Wu and K. He, "Group normalization," in *European Conference on Computer Vision (ECCV)*, 2018.

[29] X-Y Zhou, J. Sun, N. Ye, X. Lan, Q. Luo, B-L. Lai, G-Z. Yang P. Esperanaca, and Z. Li, "Batch group normalization," *arXiv: 2012.02782*.

[30] D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *Proc. 18th ACM-SIAM symp. on Discrete Algorithms*, 2007, pp. 1027–1035.

[31] K.P. Murphy, *Probabilistic machine learning Advanced topics - An Introduction*, MIT Press, 2023.

[32] G. P. Meyer, "An alternative probabilistic interpretation of the Huber loss," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[33] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, 2019, pp. 626–630.

[34] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) complete," [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S6A, 1993.

[35] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 504–511.

[36] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating RNN-based speech enhancement methods for noise-robust text-to-speech," in *Proc. ISCA Speech Synth. Workshop*, 2016, pp. 146–152.