# A PRIORI SNR ESTIMATION FOR SPEECH ENHANCEMENT BASED ON PESQ-INDUCED REINFORCEMENT LEARNING

*Tong Lei*[1,2,3], *Haoxin Ruan*[1,2,3], *Kai Chen*[1,2,3], *and Jing Lu*[1,2,3]

[1]Key Laboratory of Modern Acoustics, Nanjing University, Nanjing 210093, China
[2]NJU-Horizon Intelligent Audio Lab, Horizon Robotics, Beijing 100094, China
[3]Nanjing Institute of Advanced Artificial Intelligence, Nanjing 210014, China
{tonglei, hx_ruan}@smail.nju.edu.cn, {chenkai, lujing}@nju.edu.cn

**ABSTRACT**

Perceptual evaluation of speech quality (PESQ) is widely accepted as an effective objective metric closely related to the speech quality sensed by human listening perception. Due to its evaluation complexity and non-differentiability, PESQ is difficult to include in the cost function for deep learning-based speech enhancement. In this paper, we focus on introducing PESQ to improve Deep Xi, a recently proposed minimum mean square error (MMSE) based speech enhancement with *a priori* signal-to-ratio (SNR) estimated by a deep neural network. Regarding discrete *a priori* SNR as actions, we apply reinforcement learning (RL) to select the optimal SNR at the frame level through the reward function associated with PESQ. The experimental results show that the RL-trained network is able to achieve a better PESQ score, especially in low SNR conditions.

***Index Terms***— Speech enhancement, Reinforcement learning, *A priori* SNR, Sound quality and perceptual score, Deep Xi

## 1. INTRODUCTION

The ubiquitous noise and reverberation in practical applications impair the experience of speech interaction and the performance of automatic speech recognition (ASR) significantly. Speech enhancement aims at extracting clean speech from background interference for higher speech intelligibility and perceptual quality. In recent years, deep neural network (DNN) has achieved impressive results in the field of speech enhancement due to its powerful nonlinear modeling capabilities [1-3]. For single-channel speech enhancement, end-to-end processing is the most straightforward approach, but it faces the challenge of generalization, i.e., the output of DNN might severely deteriorate in unseen noisy conditions. The recently proposed Deep Xi framework [4, 5] can be regarded as an effective hybrid approach combining the rule-based MMSE speech enhancement strategy and the data-driven deep learning approach to estimate *a priori* SNR. The benefit of this hybrid strategy is that it is possible to decouple the output calculation and the inference of the neural network, reducing the risk of deterioration in unseen noisy conditions to some extent. Unlike other noise power spectral density estimators, it does not make any assumption about the characteristics of the speech or noise, does not exhibit any tracking delay, and does not rely on bias compensation. Furthermore, DNN is only used to track the noise power spectral density (PSD) and SNR while the output is calculated based on the rule-based method.

For deep learning-based speech enhancement, it has been pointed out that the processed speech with the common criterion like the mean-square error (MSE) between the estimated signal and the clean speech does not guarantee high speech quality and intelligibility [6, 7]. Among the objective metrics related to human perception, perceptual evaluation of speech quality (PESQ) [8] and short-time objective intelligibility (STOI) [9] are two popular ones to evaluate speech quality and intelligibility. Therefore, the direct use of these two functions to optimize the model is a meaningful task. Several studies [6, 7, 10-12] have focused on the optimization of STOI score improve speech intelligibility. PESQ score cannot be improved by maximizing the STOI score as described in [6]. Some researchers [13] simplified the computation of the symmetrical disturbance vector in PESQ by applying a center-clipping operator over the absolute difference between the loudness spectra so that it can be included in the training target. However, PESQ itself is non-differentiable and the backpropagation derivative cannot be computed, so it is difficult to obtain a universal training scheme.

As a self-optimization approach, RL can be understood as taking actions in a feedback environment that allows the agent to learn the optimal policy to maximize the cumulative reward [14], and has received extensive attention in areas such as robot behavior control [15], intelligent dialogue management [16], letting robots play games [17, 18], and speech recognition [19]. The use of RL has been explored in end-to-end speech enhancement models [7, 20] and it has been verified that enhanced speech does lead to better PESQ scores. Notably, RL enhancement schemes also have the advantage of requiring less training data [20].

In this paper, we introduce PESQ into Deep Xi by regarding the *a priori* SNR as the actions and designing the reward associated with PESQ. The discrete actions are composed of a pre-trained frame-level *a priori* SNR set and the *a priori* SNR obtained by Deep Xi-TCN [4, 5]. Then the double Q-learning strategy is utilized to select the optimal *a priori* SNR with a reward function on PESQ.

## 2. REVIEW OF THE DEEP XI HYBRID METHOD

### 2.1. Problem formulation

The signal model in the time-frequency domain can be obtained by the short-time Fourier transform (STFT):

This work was supported by the National Natural Science Foundation of China under Grant 11874219.

$$Y_l[k] = S_l[k] + D_l[k], \quad (1)$$

where $Y_l[k]$, $S_l[k]$, $D_l[k]$ are the complex-valued STFT coefficients for noisy speech, clean speech and noise, respectively, with time-frame index $l \in \{0,1,...,L\}$ and discrete-frequency index $k \in \{0,1,...,K\}$. We apply the standard assumption of the Deep Xi framework [5] that $S_l[k]$ and $D_l[k]$ are statistically independent across time and frequency, and follow the conditional zero-mean Gaussian distribution with spectral variances of $\lambda_s[l,k]$ and $\lambda_d[l,k]$. Let $R_l[k] = \left|Y_l[k]\right|$, the *a priori* SNR $\xi$ and the *a posteriori* SNR $\gamma$ can be defined as

$$\xi = \frac{\lambda_s}{\lambda_d}, \gamma = \frac{R_l[k]^2}{\lambda_d}. \quad (2)$$

### 2.2. Deep Xi framework

In [4], it is assumed that the instantaneous *a priori* SNR (in dB) obeys the following Gaussian distribution:

$$10\log_{10}(\xi_l[k]) \sim \boldsymbol{\mathcal{N}}(\mu_k, {\sigma_k}^2), \quad (3)$$

with the mean $\mu_k$ and the variance ${\sigma_k}^2$. It is mapped to the interval [0, 1] in order to improve the convergence rate of the stochastic gradient descent algorithm. The cumulative distribution function (CDF) of $10\log_{10}(\xi_l[k])$ is used as the map with

$$\bar{\xi}_l[k] = \frac{1}{2}\left[1 + erf\left(\frac{10\log_{10}\left(\xi_l[k]\right) - \mu_k}{\sqrt{2}\sigma_k}\right)\right], \quad (4)$$

where $erf(\cdot)$ denotes the Gaussian error function. The *a priori* SNR estimation $\hat{\xi}_l[k]$ can be restored by

$$\hat{\xi}_l[k] = 10^{\frac{1}{10}(\sqrt{2}\sigma_k erf^{-1}(2\hat{\bar{\xi}}_l[k]-1)+\mu_k)} \quad (5)$$

where $\hat{\bar{\xi}}_l[k]$ is the estimation of the mapped SNR.

After estimating the *a priori* SNR, a corresponding gain function is required to recover the estimated signal. The minimum mean square error log spectral amplitude (MMSE-LSA) estimator minimizes the MSE between the log spectrum of clean and enhanced speech, which was proven to be one of the best performing rule-based gain functions [21]. The instantaneous *a posteriori* SNR is estimated from the instantaneous *a priori* SNR [4] as $\gamma = \xi + 1$, resulting in a gain function given by

$$G_{\text{MMSE-LSA}}(\xi) = \frac{\xi}{\xi+1}\exp\left(\frac{1}{2}\int_{\xi}^{\infty}\frac{e^{-t}}{t}dt\right). \quad (6)$$

## 3. PROPOSED METHOD

Our proposed RL approach aims at increasing the PESQ score. The Deep Q Network (DQN) is used to identify the clean speech magnitude spectrum from the noisy speech magnitude spectrum and select the *a priori* SNR with the highest reward related to the PESQ score. Figure 1 shows the overall framework of the proposed method, which is formed by an initialization stage and a training stage.

### 3.1 Initialization stage

In the initialization stage, a frame-level mapped *a priori* SNR $\hat{\bar{\xi}}_l$ is obtained by the already trained Deep Xi-TCN network. The resulting $\hat{\xi}_l$ calculated by Eq. (5) is regarded as a candidate action denoted as $\boldsymbol{\mathcal{X}}_0$. To form a complete action set, another *M* candidate actions are formed by the K-means clustering algorithm on the ideal *a priori* SNR $\xi_l^{ideal}$ generated from the ratio between the squared magnitudes of clean speech and noise in the training set. In this way, the finite action set with *M*+1 candidates as $\{\boldsymbol{\mathcal{X}}_0, \boldsymbol{\mathcal{X}}_1, ..., \boldsymbol{\mathcal{X}}_M\}$ is generated. The action of the *l*-th frame $\boldsymbol{a}_l$ is selected in accordance with the observation $\boldsymbol{x}_l$ and its own selection policy $Q(\boldsymbol{x}_l, \boldsymbol{a}_l)$ which is an action-value function fitted by a DQN network.

$$\hat{\boldsymbol{a}}_l \leftarrow \underset{\boldsymbol{a}_l \in \{\boldsymbol{\mathcal{X}}_0, \boldsymbol{\mathcal{X}}_1, ..., \boldsymbol{\mathcal{X}}_M\}}{\arg\max}\ Q(\boldsymbol{x}_l, \boldsymbol{a}_l), \quad (7)$$

The input vector $\boldsymbol{x}_l$, which is passed to the first layer of the network as $\boldsymbol{z}_l^{(1)} = \boldsymbol{x}_l$, is obtained by concatenating several frames of observation features to account for previous and future frames, as

$$\boldsymbol{x}_l = [\boldsymbol{R}_{l-P}, ..., \boldsymbol{R}_l, ..., \boldsymbol{R}_{l+P}]^T, \quad (8)$$

$$\boldsymbol{R}_l = [R_l[0], R_l[1], ..., R_l[K]], \quad (9)$$

with 2P+1 the context window size, and $\boldsymbol{R}_l$ the noisy speech magnitude spectrum vector.

We pre-train the network in order to obtain reasonable initialization parameters $\Theta_q$ for the DQN network before training. The *a priori* SNR of the training set is calculated and mapped to $\{\boldsymbol{\mathcal{X}}_1, ..., \boldsymbol{\mathcal{X}}_M\}$ in the MMSE sense as

$$\boldsymbol{a}_l^{Label} \leftarrow \underset{\boldsymbol{a} \in \{\boldsymbol{\mathcal{X}}_1, ..., \boldsymbol{\mathcal{X}}_M\}}{\arg\min} \sum_{k=0}^{K} \left|\left|S_l[k]\right| - \left|Y_l[k] \cdot g_{\boldsymbol{a}}[k]\right|\right|^2, \quad (10)$$

$$g_{\boldsymbol{a}}[k] = G_{\text{MMSE-LSA}}(\boldsymbol{a}[k]), \quad (11)$$

where $G_{\text{MMSE-LSA}}(.)$ is the MMSE-LSA gain calculated for each frequency using Eq. (6). The index number from 1 to M is used as the label of the corresponding frame in the training set. This process can be viewed as a classification task. The network parameters are updated by back-propagation with cross-entropy loss. The weights and biases of each fully connected layer are initialized following a normal distribution.

### 3.2 Training stage

In the training stage, the parameters of DQN $\Theta_q$ are trained with the goal of maximizing the reward related to PESQ. The reward setting is described in Section 3.3. The double Q-Learning strategy [22] is used for training, which decouples the selection from the evaluation to prevent overfitting without requiring additional networks or parameters. We have two DQN networks with different update rates: the one that is updated at each iteration is called the evaluation DQN (Eval. DQN), and the one that periodically copies the parameters of the Eval. DQN is called the Target DQN. The noisy magnitude spectrum block $\boldsymbol{x}_l$ is fed into both networks simultaneously, generating $\hat{\boldsymbol{a}}_l^{\text{Eval.}}$ from $Q'(\boldsymbol{x}_l, \boldsymbol{a}_l)$ by the Eval. DQN and $\hat{\boldsymbol{a}}_l^{\text{Target}}$ from $Q(\boldsymbol{x}_l, \boldsymbol{a}_l)$ by the Target DQN using Eq. (7). Apart from the updating rates, another difference between these two DQNs is that the Target DQN directly follows the standard process of DQN to select actions, while the Eval. DQN exploits a random action option with probability $\epsilon$.

After making their action choices, both networks generate their estimated speech waveform $\hat{\boldsymbol{s}}^{\text{Eval.}}$ and $\hat{\boldsymbol{s}}^{\text{Target}}$ respectively using the inverse short-time Fourier transform (iSTFT) for the next reward computation as

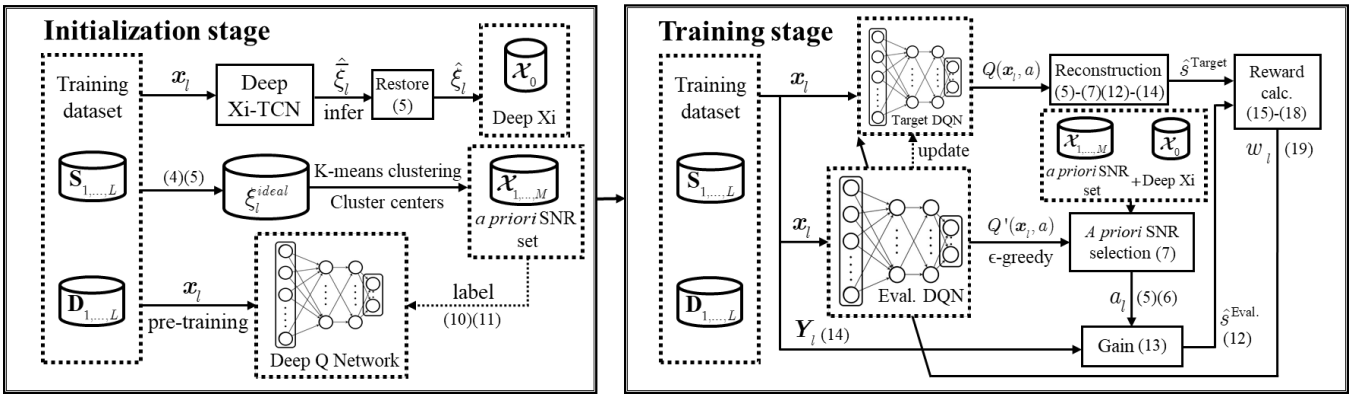


**Fig. 1**. Overview of the procedure of the proposed method

$$\hat{\boldsymbol{s}} = \mathrm{iSTFT}(\hat{\boldsymbol{S}}), \tag{12}$$

$$\hat{\boldsymbol{S}} = \boldsymbol{Y}_l \odot G_{\text{MMSE-LSA}}(\hat{\boldsymbol{a}}_l), \tag{13}$$

$$\boldsymbol{Y}_l = [Y_l[0], Y_l[1], ..., Y_l[K]]^T, \tag{14}$$

where $\odot$ is the Hadamard product, and $\hat{\boldsymbol{a}}_l = [\hat{a}_l[0], \hat{a}_l[1], ..., \hat{a}_l[K]]^T$ is the estimated *a priori* SNR from DQN of the $l$-th frame.

Then a reward $w_l$ based on the difference between their PESQs is calculated, and the DQN parameters are updated in a self-optimization manner.

### 3.3. Reward design for Q-learning

The reward is crucial [23] for RL. To design an appropriate reward for different SNRs and different noise types, we need to constrain the range of the rewards. The relative PESQ value between the evaluation network and the target network, similar to that in [20], is used as the reward:

$$\boldsymbol{\mathcal{W}} = \tanh\{\alpha(\boldsymbol{\mathcal{Z}}^{\text{Eval.}} - \boldsymbol{\mathcal{Z}}^{\text{Target}})\}, \tag{15}$$

where $\alpha > 0$ is a scaling parameter, and $\boldsymbol{\mathcal{Z}}^{\text{Target}}$ and $\boldsymbol{\mathcal{Z}}^{\text{Eval.}}$ are the PESQ values calculated from the estimated waveform $\hat{\boldsymbol{s}}^{\text{Target}}$ and $\hat{\boldsymbol{s}}^{\text{Eval.}}$ respectively. Considering that the *a priori* SNR is time-varying and the PESQ value cannot be computed in one frame, a time-varying reward needs to be computed with multiple frames. We utilize a time-weight $E_l \in [0,1]$ in the reward calculation as

$$\tilde{E}_{l'} = \sum_{k=0}^{K} \left| \ln\left|S_{l'}[k]\right| - \ln\left|Y_{l'}[k] \cdot G_{\text{MMSE-LSA}}(\hat{\boldsymbol{a}}_{l'}^{\text{Eval.}}[k])\right| \right|^2, \tag{16}$$

$$E_l = \frac{\tilde{E}_l}{\max_{l' \in \{0,1,...,L\}}(\tilde{E}_{l'})}, \tag{17}$$

$$w_l = \begin{cases} (1 - E_l)\boldsymbol{\mathcal{W}} & (\boldsymbol{\mathcal{W}} > 0) \\ E_l \boldsymbol{\mathcal{W}} & (\boldsymbol{\mathcal{W}} < 0) \end{cases}. \tag{18}$$

Once the ϵ-greedy strategy has randomly selected the action $a_\varepsilon$ from $\{\boldsymbol{\mathcal{X}}_0, \boldsymbol{\mathcal{X}}_1, ..., \boldsymbol{\mathcal{X}}_M\}$ with probability ϵ that differs from the best action $\hat{\boldsymbol{a}}_l^{\text{Eval.}}$, the desired $Q$ value of the action-value function for the Eval. DQN iteration is updated as

$$\begin{cases} \tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_\varepsilon) = w_l + Q(\boldsymbol{x}_l, \hat{\boldsymbol{a}}_l^{\text{Target}}) & (w_l > 0) \\ \tilde{Q}'(\boldsymbol{x}_l, \hat{\boldsymbol{a}}_l^{\text{Target}}) = Q(\boldsymbol{x}_l, \hat{\boldsymbol{a}}_l^{\text{Target}}) - w_l & (w_l < 0) \end{cases}, \tag{19}$$

where $\tilde{Q}'(\boldsymbol{x}_l, \hat{\boldsymbol{a}}_l^{\text{Target}})$ and $\tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_\varepsilon)$ are the desired $Q$ values of Eval. DQN corresponding to actions $\hat{\boldsymbol{a}}_l^{\text{Target}}$ and $\boldsymbol{a}_\varepsilon$ respectively. When $w_l < 0$, the desired $Q$ values of the Eval. DQN is the maximum $Q$ value of the Target DQN subtracted by $w_l$ ($\boldsymbol{\mathcal{W}} < 0$ in this case), rewarding the action selected by the Target DQN. In addition, to put an upper bound on the $Q$ value of the DQNs, the activation function of its output layer is Softmax. Accordingly, $\tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_l)$ will also be normalized to satisfy $\sum_a \tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_l) = 1$ as

$$\tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_l) = \frac{\exp(\tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_l))}{\sum_{\boldsymbol{a}} \exp(\tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_l))}. \tag{20}$$

Finally, the parameters $\Theta_q$ are updated by minimizing the following equation to make $Q'(\boldsymbol{x}_l, \boldsymbol{a}_l)$ close to the desired value $\tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_l)$.

$$\Theta_q \leftarrow \arg\min_{\Theta_q} \frac{1}{L+1} \sum_{l=0}^{L} \sum_{\boldsymbol{a}_l} \left| \tilde{Q}'(\boldsymbol{x}_l, \boldsymbol{a}_l) - Q'(\boldsymbol{x}_l, \boldsymbol{a}_l) \right|^2. \tag{21}$$

In order to minimize Eq. (21), the RMSProp algorithm with the standard mini-batch stochastic gradient descent (SGD) is used.

During the inferring stage, only the trained Deep Xi-TCN and the Target DQN are used. Instead of only inferring $\boldsymbol{\mathcal{X}}_0$ in Deep Xi-TCN, the Target DQN determines which of the M+1 candidates is the most appropriate for a given frame.

## 4. EXPERIMENTS

### 4.1. Experimental conditions

Our proposed method is named as XiDQN, and we compare its performance with Deep Xi-TCN [5] in this section. In our experiments, the clean speech corpora include the TIMIT speech corpus [24] (6289 utterances) and the *train-clean-100* set from the Librispeech corpus [25] (28539 utterances). The noise recordings include the Nonspeech corpus [26], the Environmental Background Noise dataset [27, 28] and the noise part of the MUSAN corpus [29]. The clean speech and noise are divided into the training set, the validation set and the test set with the ratio of 0.7, 0.1 and 0.2 respectively. In addition, white noise is added to the noise part of the training set and a modulated white noise similar to [5] is added to the noise part of the test set. All speech and noise recordings are unified into a sampling rate of 16kHz (recordings with a sampling frequency higher than 16 kHz are down-sampled to 16 kHz). Noisy speech signals are generated with the following rule: each clean speech is mixed with a randomly selected noise signal at a random SNR from -10 dB to 15 dB with 1 dB increments.

**TABLE 1**. PESQ and STOI (%) scores for the test set.

| Metrics | SNR (dB) | -6 | 0 | 6 | 12 | Avg. |
|---|---|---|---|---|---|---|
| PESQ | Noisy | 1.36 | 1.58 | 1.94 | 2.34 | 1.83 |
| | Deep Xi-TCN | 1.69 | 2.08 | 2.37 | 2.76 | 2.23 |
| | XiDQN | 1.82 | 2.18 | 2.43 | 2.79 | 2.31 |
| STOI (%) | Noisy | 59.02 | 73.50 | 83.31 | 91.71 | 76.89 |
| | Deep Xi-TCN | 72.07 | 82.49 | 88.67 | 92.84 | 84.02 |
| | XiDQN | 72.46 | 82.87 | 88.92 | 92.98 | 84.31 |

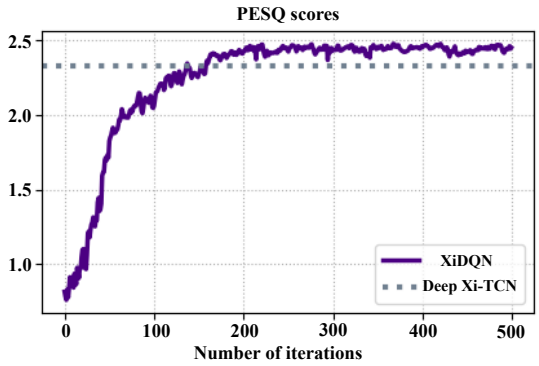


**Fig. 2**. PESQ score of the target DQN with increasing number of iterations during training.

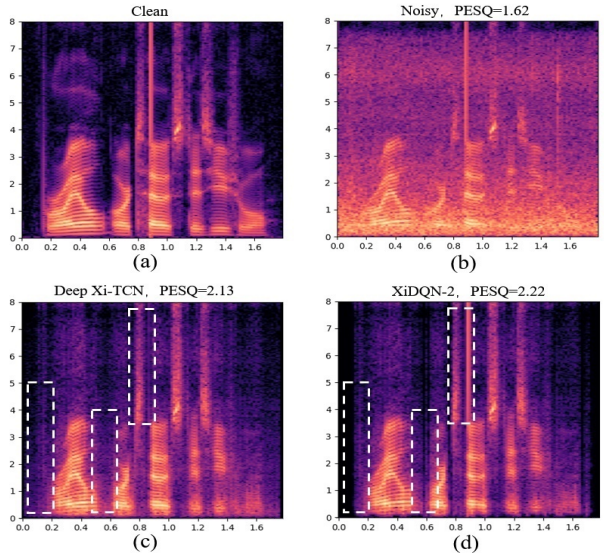


**Fig. 3**. Spectrograms of a TIMIT utterance: (a) clean speech, (b) noisy speech (Drive noise at 0 dB SNR, PESQ=1.62), (c) enhanced speech by Deep Xi-TCN (PESQ=2.13), (d) enhanced speech by XiDQN (PESQ=2.22).

In Deep Xi-TCN, the temporal convolution network (TCN) consists of a fully connected (FC) layer connecting the input spectrum and 40 residual blocks, followed by a fully connected layer that employs sigmoidal units. The parameters of the network are the same as those in [5]. The DQN used in our framework consists of two FC hidden layers with 66 units and the Sigmoid activation functions. The activation function of the output layer is Softmax. The number M of the *a priori* SNR templates in the action set is 32, the adjustable scale parameter $\alpha$ in Eq. (15) is set to 20, and the contextual half-window size $P$ is set to 15. The dropout technique is used in training to avoid overfitting with a probability of 0.5. The greedy parameter $\epsilon$ decreases from 0.20 to 0.01 linearly, and the whole decreasing process takes 200 iterations (close to the empirical number of iterations where the PESQ score is about to converge). The learning rate is set using the 1cycle learning rate method [30] for training acceleration, which increases and then decreases between 0.00001 and 0.0005. A Hanning window is used for STFT, with a 512-sample frame size and a 256-sample shift.

### 4.2. Experimental results

Figure 2 shows the variation of the PESQ score calculated from the estimated speech of the Target DQN during training. For comparison, the fixed average PESQ score calculated from the trained Deep Xi-TCN is also depicted. A mini-batch of 8 training audios is used to iteratively update the Eval. DQN, whose parameters are periodically copied to the Target DQN every 20 updates. It can be seen in Fig. 2 that the PESQ score increases with the number of iterations, and surpasses that of the Deep Xi-TCN after about 160 iterations. XiDQN has about 0.11 overall PESQ improvement over Deep Xi-TCN after convergence. It should be noted that the convergence behavior of the PESQ score is not as smooth as a common learning curve of Deep Xi since the PESQ is calculated by randomly selected samples from the training dataset.

On the test set, we used the STOI as an evaluation metric in addition to PESQ. Table 1 lists the PESQ and STOI (%) scores for the enhanced speech at -6 dB, 0 dB, 6 dB, and 12 dB SNR. The advantage of XiDQN on STOI, though dwarfed by the advantage on PESQ, can be seen. Note that at low SNR, the XiDQN method has a more significant improvement over Deep Xi-TCN, which indicates that the action selection made by the XiDQN network leads to a significantly better gain when the noise energy is relatively high.

Figure 3 shows the spectrograms of a typically processed speech at 0 dB SNR. The benefit of the proposed XiDQN can be seen by comparing sub-figures (c) and (d). The left 2 dashed boxes on each sub-figure demonstrate more effective noise suppression of XiDQN while the right dashed box demonstrates that XiDQN retains the consonant syllable more clearly. More exemplary audio samples are available online at https://github.com/Taltt/XiDQN.

It should be noted that the performance of Deep Xi-TCN is inferior to that shown in Ref. [5] since the training dataset in our experiments is only about a quarter in quantity.

## 5. CONCLUSION

In this paper, we design XiDQN by introducing the PESQ metric into the Deep Xi framework using reinforcement learning. A discrete action set is established by combining the pre-trained *a priori* SNR candidates and the *a priori* SNR obtained from Deep Xi-TCN. The double Q-learning strategy is then utilized to optimally select the *a priori* SNR from the action set. To guarantee a more effective training process, the DQN is also pre-trained. The experimental results validate the benefit of the proposed framework on both PESQ and STOI scores.

## 6. REFERENCES


[1] F. Weninger *et al.*, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *International Conference on Latent Variable Analysis and Signal Separation*, Springer, pp. 91–99, 2015.

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

[3] X. Le, H. Chen, K. Chen, and J. Lu, "DPCRN: Dual-path convolution recurrent network for single channel speech enhancement," in *Interspeech*, pp. 2811–2815, 2021.

[4] A. Nicolson and K. K. Paliwal, "Deep learning for minimum mean-square error approaches to speech enhancement," *Speech Communication*, vol. 111, pp. 44–55, 2019.

[5] Q. Zhang, A. Nicolson, M. Wang, K. K. Paliwal, and C. Wang, "DeepMMSE: A deep learning approach to MMSE-based noise power spectral density estimation," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 28, pp. 1404–1415, 2020.

[6] S. Fu, T. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1570–1584, 2018.

[7] Y. Koizumi, K. Niwa, Y. Hioka, K. Kobayashi, and Y. Haneda, "DNN-based source enhancement to increase objective sound quality assessment score," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1780–1792, 2018.

[8] Recommendation, ITU-T, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[9] 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, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[10] M. Kolbæk, Z.-H. Tan, and J. Jensen, "Monaural speech enhancement using deep neural networks by maximizing a short-time objective intelligibility measure," in *proc. ICASSP*, pp. 5059–5063, 2018.

[11] H. Zhang, X. Zhang, and G. Gao, "Training supervised speech separation system to improve STOI and PESQ directly," in *proc. ICASSP*, pp. 5374–5378, 2018.

[12] Y. Zhao, B. Xu, R. Giri, and T. Zhang, "Perceptually guided speech enhancement using deep neural networks," in *proc. ICASSP*, pp. 5074–5078, 2018.

[13] J. M. Martín-Doñas, A. M. Gomez, J. A. Gonzalez, and A. M. Peinado, "A deep learning loss function based on the perceptual evaluation of the speech quality," *IEEE Signal Processing Lett.*, vol. 25, no. 11, pp. 1680–1684, 2018.

[14] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Mag.*, vol. 12, no. 2, pp. 19–22, 1992.

[15] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *IEEE International Conference on Robotics and Automation,* vol. 3, pp. 2619–2624, 2004.

[16] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 105–133, 2002.

[17] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, pp. 529–533, 2015.

[18] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *nature*, vol. 529, pp. 484–489, 2016.

[19] T. Kala and T. Shinozaki, "Reinforcement learning of speech recognition system based on policy gradient and hypothesis selection," in *proc. ICASSP*, pp. 5759–5763, 2018.

[20] Y. Koizumi, K. Niwa, Y. Hioka, K. Kobayashi, and Y. Haneda, "DNN-based source enhancement self-optimized by reinforcement learning using sound quality measurements," in *proc. ICASSP*, pp. 81–85, 2017.

[21] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[22] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[23] M. Sugiyama, *Statistical Reinforcement Learning: Modern Machine Learning pproaches*. CRC Press, 2015.

[24] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *proc. ICASSP*, pp. 5206–5210, 2015.

[26] G. Hu, "100 nonspeech environmental sounds," *The Ohio State Uni., Dept.Comput. Sci. Eng.*, 2004.

[27] F. Saki and N. Kehtarnavaz, "Automatic switching between noise classification and speech enhancement for hearing aid devices," in *Proc. EMBC*, pp. 736–739, 2016.

[28] F. Saki, A. Sehgal, I. Panahi, and N. Kehtarnavaz, "Smartphone-based real-time classification of noise signals using subband features and random forest classifier," in *proc. ICASSP*, pp. 2204–2208, 2016.

[29] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[30] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.