Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

Applied Acoustics 241 (2026) 111022

Contents lists available at ScienceDirect

Applied Acoustics

journal homepage: www.elsevier.com/locate/apacoust

STEM: spatial speech separation using twin-delayed DDPG reinforcement learning and expectation maximization

Muhammad Salman Khan [a,*], Sania Gul [b]

[a] Department of Electrical Engineering, College of Engineering, Qatar University, Doha, Qatar
[b] Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan

ARTICLE INFO

Keywords:
Speech separation
Reinforcement learning
Continuous action space
Spatial cues
Reward function
Time–frequency masking

ABSTRACT

Although many high-performing speech separation models have been proposed recently, little attention has been paid to making them lightweight. In this paper, a novel speech separation algorithm is proposed that integrates the twin-delayed deep deterministic (TD3) policy gradient reinforcement learning (RL) agent with the expectation maximization (EM) algorithm for clustering the spatial cues of individual sources separated on azimuth. For stationary sources, the proposed system gives satisfactory performance in terms of quality, intelligibility, and separation speed, and generalizes well with the test data from a mismatched speech corpus. Its perceptual evaluation of speech quality (PESQ) score is 0.55 points better than a self-supervised learning (SSL) model and almost equivalent to the diffusion models at computational cost and training data which is many folds lesser than required by these algorithms. Additionally, it reduces the required training data by 39 times, training time by 36 times, model size by 6 times, real time factor (RTF) by 1 point, and multiply-accumulate operations (MACs) by 9 times compared to a recently proposed lightweight transformer-based encoder-decoder framework, while offering a slight decrease in PESQ score (by 0.45 points).

1. Introduction

The cocktail party effect (the term first coined by Colin Cherry [1]) or selective auditory attention [2], is defined as the psychoacoustic phenomenon that refers to the amazing human ability to selectively attend and recognize a specific sound and filter out the rest from the cacophony of competing talkers and ambient noises that are often independent of one another [3]. For the people blessed with normal hearing, this task seems to be a trivial one but its complexity is revealed in environments where the interfering noise is loud or when the listener has any hearing impairment [4]. Similarly, the machines need to be equipped with selective attention capability to make possible a seamless interaction between man and machine in a noisy environment [5].

Blind source separation (BSS) refers to the algorithms designed to recover individual sources from an audio mixture with little or no information about the mixing methodology or the sources themselves [6]. Although the goal of BSS is similar to beamforming i.e. to reduce the interference, it does not require any information about the target's position. It is sometimes also called "blind beamforming". Secondly, the BSS recovers all sources simultaneously from the mixture, while the beamforming extracts only the target source [6]. BSS is used in seismic signal processing, separation of lyrics from the background music, image processing, medical signal processing, telecommunications, and cocktail party problem [7], which in turn improves the efficiency of automatic speech recognition (ASR), automatic speaker verification (ASV), virtual assistants, hearing aids, cochlear implants, speech to text converters [5], underwater acoustics [8] and animal sound classification [9]. The research on the cocktail party problem dates back to the early 1950s and originated from the complaints received by air traffic controllers who faced difficulty understanding the messages received simultaneously from pilots over multiple loudspeakers installed in control towers. Since then it has been addressed by conventional signal processing methods (based on frequency-domain filtering) [3] to the recent trend of machine and deep learning methods. Although the deep learning methods offer improved separation performance than the signal processing and machine learning methods, their downside is that these models require a large amount of data, lengthy durations, and high-end computational resources for training, and their high separation performance is achieved at the cost of networks with a very large number of trainable parameters [10]. On the other hand, although the quality of the estimated output of unsupervised conventional machine learning source separation models is not on par with the deep learning models, their few

* Corresponding author.
E-mail address: salman@qu.edu.qa (M.S. Khan).

https://doi.org/10.1016/j.apacoust.2025.111022
Received 3 March 2025; Received in revised form 22 July 2025; Accepted 15 August 2025
Available online 23 August 2025
0003-682X/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

parameters can be estimated on as little as a single frame of an audio mixture [11]. So, in many source separation models, deep learning is integrated with machine learning, a combination that outperforms either of its components [12].

The domain of machine learning models can be broadly classified into three categories: 1) supervised, 2) unsupervised, and 3) reinforcement learning (RL). While supervised and unsupervised learning have been used extensively for almost all kinds of audio applications, RL has been used on a very limited scale for such tasks [13]. The concept of RL originated from animal learning [14] and is defined as learning by trial and error [15]. RL allows autonomous systems to learn from their experiences rather than exclusively from the supervision of a 'knowledgeable' teacher [16]. This makes RL learning adaptive to the dynamic environments [17]. An RL agent placed in an unknown environment chooses an action (from a predefined action space) to maximize the received reward, over the long run [17]. To accomplish its task, the agent interacts with the environment by taking an action and modifies its behavior according to the reward received from the environment. The agent must learn the sequence of actions (called policy), which maximizes the overall cumulative reward over time [18]. In supervised learning or 'learning with the master', the difference between the actual and the reference output is used to modify the learning system's parameters, while in reinforcement learning or 'learning with critic', the reward signal is a sanction of the agent's behavior. It tells the agent 'what to do' instead of 'how to do' it. RL is mainly used for control system applications, navigation systems, robotics, and video games [19]. The introduction of deep neural networks in the RL domain has given birth to a new class of RL, called deep reinforcement learning (DRL) [20]. DRL has made possible the application of RL to the problems that were previously intractable due to their high dimensional state and action spaces [20]. An on-policy DRL agent will learn its policy by directly interacting with its environment. On the other hand, an offline policy agent will learn from an offline database (called experience buffer) of the previously collected set of experiences. The offline paradigm is extremely valuable in situations where online interaction is impractical due to the data collection being extremely dangerous or expensive [21]. Also, it results in improved generalization in complex domains. Both on and off-policy agents continuously interact with the environment to update their policy [21]. After learning an offline policy, the offline agent can be tuned later online with the added benefit of avoiding nonsensical behavior due to adopting an entirely random initial policy [21]. SARSA, REINFORCE policy gradient (PG), actor-critic (AC), and proximal policy optimization (PPO) are few examples of online policy agents, whereas Q-Learning, deep Q-network (DQN), deep deterministic policy gradient (DDPG), twin-delayed deep deterministic (TD3) policy gradient, soft actor-critic (SAC) are offline agents [22].

The goal of the most recent state-of-the-art (SOTA) deep learning speech separation models e.g. permutation invariant training (PIT) [23], WaveSplit [24], Conv TASNet [25], DPTNet [26], SepFormer [27], TFGridNet [28] and MossFormer [29] is to optimize the scale-invariant signal-to-noise ratio (SI-SNR) and trained on WSJ0-2 mix [30]. However, it is found that these models do not generalize well to a wide range of speakers (obtained from other datasets or real recordings) and recording conditions [10]. Also, the SI-SNR metric does not align well with the human perception of speech quality and the models trained to optimize it may generate perceptually unnatural distortions [30]. Directly using the metrics that correlate well with the human perception of quality (e.g. perceptual evaluation of speech quality (PESQ) score) as an objective function for deep learning models is not possible due to 1) the metric being non-differentiable and 2) extremely complex correlation that exists between the input and output of deep neural networks [31]. RL provides the option to implement such non-differentiable metrics in calculating its step reward function (defined as the reward generated by the environment, after an action taken by the agent) to update the parameters of the network.

In this paper, an RL 'TD3' agent is integrated with a machine learning (ML) algorithm called expectation maximization (EM) to achieve the goal of speech separation from an audio mixture. The separation is done on the basis of differences in spatial cues of the sources separated in azimuth. In ideal conditions, binaural listening not only improves the ability to detect a signal in the presence of interfering signals by 25 dB over monaural listening [32] but the binaural (also known as spatial or interaural) cues also play a vital role in locating and segregating the simultaneously active sources in azimuth [33]. The two most important spatial cues are: 1) the interaural level difference (ILD), and 2) the interaural time difference (ITD). ILD is defined as the difference in the intensity level of the signals received by both ears while the ITD is defined as the time difference in the arrival of signal at the two ears [8]. In the past, both machine and deep learning models have used these cues for speech separation. Examples of spatial-cue-based models using the machine learning (ML) algorithms are BSS algorithms [34] (based on non-negative matrix factorization (NMF)), [9] (based on non-negative tensor factorization (NTF)), [35] (based on expectation maximization (EM)) and [36] (based on Bayesian inference), while the models utilizing solely the deep neural networks are models [37] (using convolutional neural network (CNN)), [38] (based on autoencoders), [39] (using recurrent neural network), [40] (using transformers), and [41] (based on diffusion networks). Few spatial-cue-based BSS models have also integrated machine learning with deep learning e.g. [11,12,42–44] to achieve the benefits of both domains.

Among speech enhancement (SE) algorithms (the algorithms designed to enhance the quality of speech corrupted by ambient noise (speech denoising (SD), and competing talkers (speech separation (SS) and recover it in case of loss (speech inpainting)). Pioneered by Koizumi et al. [31] for SD applications, a Q-learning agent (with a discrete action space; determined by the K-means algorithm (an ML algorithm)) is employed for optimizing a deep feed-forward neural network-based model. After that many SD models using RL agents have been proposed e.g. SD models [45] and [46] are based on deep Q-network (DQN) agents, SD algorithm [47] uses a PG agent, and the speech denoisers [48–51] use a Q-learning agent. The model [52] is based on a REINFORCE agent, while the model [53] uses a PPO agent, and the model [54] utilizes an actor-critic (AC) agent. In all these models, an RL agent does not act as a speech denoiser itself but is only used to enhance the performance of another deep learning-based SD system. Till now, RL has never been tried for speech inpainting and has been used in very few BSS models only to upgrade their performance. They have never been tried for the design of autonomous speech separation networks. A brief introduction of the BSS models using RL is given below.

2. Related work

In the case of BSS, RL is used for the first time in model [55]. In this model, a module called 'future success' (motivated by the reward concept and similar to the actor-critic agent of RL) is used in conjunction with a conditional generative adversarial network (cGAN) to enhance the performance of an already trained speech separation model. The role of the future success module is to directly optimize the performance metric without adding complexity to the core audio separation network. The future success module is not an RL agent in its true sense as it does not rely on policy gradient to update its generator (equivalent to the actor of AC), but rather uses the loss from the critic to do so.

The audio-visual source separation (AVSS) model [56], designed for an active robot's listening, uses a PPO agent to control the movement of the microphone and camera for efficient beamforming and steering while walking towards the desired target. However, this model is only effective for separating static sound sources such as a fire alarm or a ringing phone but not for the dynamic sources (non-periodic and time-varying) e.g. speech and music [57]. The AVSS model [57] is an improved version of [56], using a decentralized distributed PPO (DD-PPO) agent for steering the robot motion toward the target. However, both these models ([56;57]) are not based on RL for their audio

Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

separation. They have dedicated audio separation networks (using U-Net and transformer-based complex networks respectively) and their outputs are used in conjunction with the video cues to design the navigation policy of an RL-based audio-visual controller to direct the robot to a location where it can better hear the sound-of-interest in the presence of other noisy actors.

Our proposed model uses RL directly as a source separation network instead of only relying upon it to boost the performance of another source separation model as done in models [55–57]. In addition, as opposed to [56] and [57], which are semi-blind models (using video cues along with the audio cues), our proposed algorithm is completely blind as it is based only on audio cues. To the best of our knowledge, it is the first time an RL agent has been used as an autonomous audio separator.

2.1. Our contribution

The main contributions of this paper are summarized below:

1. It is the first time that the TD3 agent is used for any audio application. We believe, it is also the first time that any off-policy agent (TD3) is used for an SE model let alone BSS.
2. Unlike the models [56] and [57], our proposed model does not use video cues and is a purely audio-based model.
3. As stated above, only supervised and unsupervised learning models were used previously in conjunction with the ML algorithms for clustering 'spatial cues' e.g. [11,12,42–44]. It is also the first time that an RL agent integrated with an ML algorithm is applied for clustering these cues.
4. Also unlike the previously proposed RL-based SD and BSS models, where an RL agent is used only to enhance the performance of existing speech denoisers and audio separators, our proposed system is a completely independent model in its own right, using an RL agent to directly process the audio.

The rest of the paper is organized as follows. In the next section, an overview of our proposed BSS algorithm is given. The agent's training details, dataset used, network settings, evaluation metrics, and the

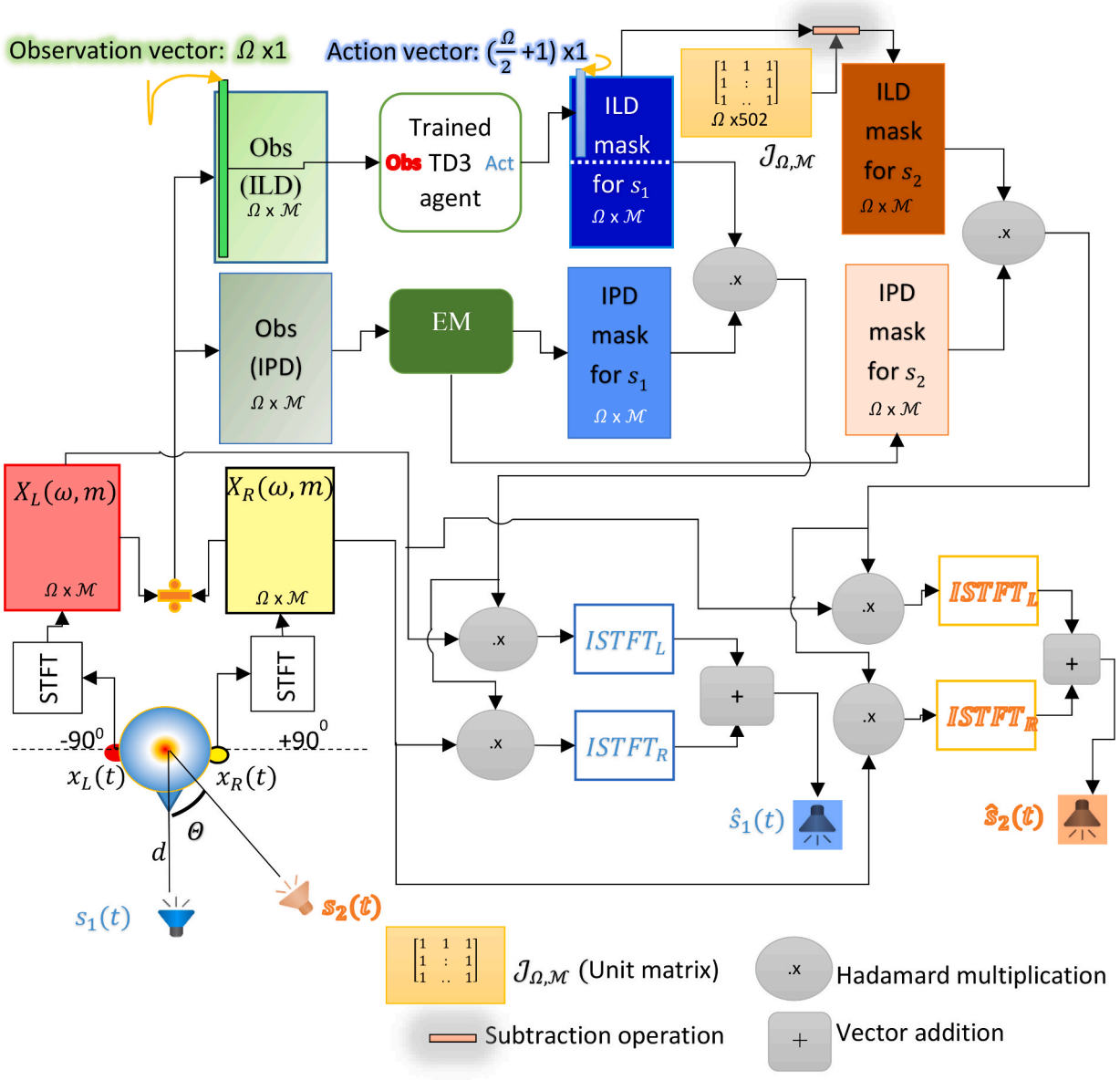

Fig. 1. Testing phase of STEM. The dotted line in the 'ILD mask for s_1' block is the action flipping line. $\Omega \times \mathcal{M}$ represents the matrix size (in the form of rows x columns) of the variable inside each block.

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

baseline algorithms used for comparison are described in section 3. Experimental results are presented in section 4 and the discussion and future work are detailed in section 5. The paper is concluded in section 6.

3. System overview

The proposed BSS model using binaural cues is called 'STEM'; an acronym for 'Spatial-cue-based speech separation by TD3 and EM'. The block diagram of STEM is given in Fig. 1 and it is designed for separating the speech of two sources placed at different angles in azimuth. The elevation of both sources is the same as the binaural setup.

3.1. Problem formulation

Assume there are two active sources s_1 and s_2 separated by an azimuthal angle Θ placed at a radial distance d in front of a binaural setup. The source s_1 is the target source while the source s_2 is the interferer. The signals from both sources are normalized to make the target-to-interferer ratio (TIR) equal to 0 dB. The TIR of 0 dB is a challenging case, where the target and interferer are not differentiable by looking at their energy levels [58]. This choice is made to evaluate the performance of the BSS method by suppressing the effects of its dependency on source distance and utterance energy. Each source is then convolved with its respective room impulse response (RIR) before being mixed at the two microphones installed in a binaural setup.

Assuming the noiseless conditions, each mixture x_k is given as

$$x_k(t) = s_1(t)*h_{k1}(t) + s_2(t)*h_{k2}(t) where k = [L, R] \quad (1)$$

In eq. (1), t represents continuous time, * represents convolution operation, h_{k1} is the RIR from source s_1 to the k^{th} microphone, h_{k2} is the RIR from source s_2 to the k^{th} microphone, L represents the left microphone, while R represents the right microphone.

Each mixture $x_k(t)$ is then converted from the time domain to the time–frequency (TF) domain $X_k(\omega, m)$ by taking its short-time-Fourier-transform (STFT), after segmenting it into overlapping frames and windowing these frames with the window function $w(t)$.

$$X_k(\omega, m) = STFT(x_k(t)) \quad (2)$$

The window function over which the STFT is taken in eq. (2) is the Hamming window function given as $w(t) = 0.54 - 0.46\cos\left(\frac{2\pi t}{L-1}\right)$, $0 \le t \le L-1$, where L is the frame length, and ω and m are discrete angular frequency and time bin indices respectively in the STFT domain.

The ratio of the STFTs of the left and right mixtures is taken to obtain the spatial (interaural) spectrogram as given in eq. (3).

$$\frac{X_L(\omega, m)}{X_R(\omega, m)} = \alpha(\omega, m)e^{j\varphi(\omega, m)} \quad (3)$$

where φ is the interaural phase difference (IPD) and α is the ILD at each TF unit of the spatial spectrogram indexed by ω and m. ILD is converted to decibels by $20log\alpha(\omega, m)$ and the resulting ILD spectrogram is given to the trained RL agent, while the IPD spectrogram is given to the EM algorithm. As ITDs are not directly measurable from the mixtures [35], the IPD values are translated to ITDs as described in the next subsection.

a. IPD Cues Processing

The IPD cues obtained from equation (3) face the problem of phase wrap due to the limitation of phase values in the range $(-\pi, \pi)$. Phase unwrapping is a challenging problem due to rapidly varying phase changes, phase discontinuities, and the presence of severe noise [59]. The EM algorithm has been effective in resolving the phase wrap issue in source separation models [42] and [35]. Therefore, inspired by these models, the STEM also processes the IPD cues (obtained from eq. (3)) by the EM algorithm used by a pure spatial-cue-based BSS model [35]. Although a separate RL agent was trained for IPD cues, our initial experiments (see the ablation study 'mask generation method' in the 'experiments and results' section) demonstrate that processing of phase by deep learning does not generally generate good results. This limited success stems from the inherent random characteristics exhibited by the phase spectra in the TF domain [60].

The EM algorithm is an iterative procedure of maximizing the log-likelihood function. Each iteration of the EM algorithm consists of two processes [61]. The first is the E-step where given the observed data, the missing data and current model parameters are estimated. The second is the M−step, where the log-likelihood function is maximized under the assumption that the missing data is known. Convergence is guaranteed since the algorithm increases the log-likelihood at each iteration [61]. The EM algorithm is sensitive to initialization and requires knowledge about the number of classes in data and the distribution type [35]. Our model uses the PHAT algorithm [62] for EM initialization.

Due to spatial aliasing, the IPD values from eq. (3) do not always correspond to the correct ITD values. So the top-down approach is adopted in STEM for the calculation of IPDs as in [35], where different values of delay τ are plugged in to determine the true value of IPD. The delay τ is a frequency-independent delay used for localizing the source. To make deduction tractable, it is modelled as a discrete random variable, where the set of allowable delays is specified apriori. In our case, the allowable values of τ are taken from the set {-15:0.5:+15} samples. At a sampling frequency of 8 kHz, these limits correspond to approximately ± 2 ms in the increment of 62.5μseconds [42]. The value of τ which generates the closest match to the observed IPD value is selected as the correct ITD value. The difference between the observed IPD (φ) and the estimated IPD (calculated by the delay of τ samples) is called the phase residual error $\hat{\varphi}$ and is given as

$$\hat{\varphi}(\omega, m; \tau) = \arg\left(e^{j\varphi(\omega, m)}e^{-j(\omega\tau)}\right) \quad (4)$$

$\hat{\varphi}$ also lies in the range $(-\pi, \pi)$ and is modelled as a normal distribution with mean and variance given by $\varepsilon(\omega)$ and σ^2. The reason for modelling the phase residual error by normal distribution instead of circular distributions e.g. von Mises and higher order maximum entropy distributions or linear distributions based on a Gaussian scale mixture model (a Gaussian mixture model (GMM) where all Gaussians share the same mean) is inspired by the Mandel's work [69], which shows that for the separation task, a single Gaussian works well than the above-mentioned computationally complex distributions. For source s_i, the model parameters $\hat{\theta}$ are given as:

$$\hat{\theta} = \left(\varepsilon_{i,\tau}(\omega), \sigma^2_{i,\tau}(\omega), \psi_{i,\tau}\right) \quad (5)$$

where the mean and variance are the functions of both the frequency ω and delay τ, and $\psi_{i,\tau}$ is the mixing weight i.e. the proportion of the total TF points that belong to the source s_i at the delay τ in the Gaussian mixture model (GMM), which exhibits itself due to the mixing of many such Gaussian distributions, each belonging to a unique combination of a source i and delay τ. As the STEM is trained and tested only for mixtures with two sources, the value of i (the source index) is taken from the set {1, 2}. Both i and τ are combined into a hidden variable $z_{i\tau}(\omega, m)$, which is 1 if the given TF unit of the spectrogram comes from both source i and delay τ and 0 otherwise. Each TF point must come from a single source and delay so the $\sum_{i,\tau} z_{i\tau}(\omega, m) = 1$. By marginalizing over the hidden variable $z_{i\tau}$, the log-likelihood of a given TF point is given as:

$$L(\hat{\theta}) = \log p(\hat{\varphi}(\omega, m)|\hat{\theta}) \quad (6)$$

$$L(\hat{\theta}) = \sum_{\omega, m} log \sum_{i,\tau} [p(\hat{\varphi}(\omega, m), z_{i\tau}(\omega, m), \hat{\theta}).p(z_{i\tau}(\omega, m)|\hat{\theta})] \quad (7)$$

where p represents the probability and $p(\hat{\varphi}(\omega, m))$ is the probability of each TF point belonging to the distribution $\hat{\varphi}$.

The maximum log-likelihood solution is obtained from the EM al-

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

gorithm as given in eq. (8) as:

$$L(\widehat{\theta}) = max_{\theta} \log p(\widehat{\varphi}(\omega, m)|\widehat{\theta}) \tag{8}$$

In the E-step, the likelihood v of each TF point belonging to s_i and delay τ is given as

$$v_{i,\tau}(\omega, m) \equiv p(z_{i\tau}(\omega, m)|\widehat{\varphi}(\omega, m), \theta_{\mathscr{J}}) \tag{9}$$

Where $\theta_{\mathscr{J}}$ is the estimate of model parameters after $\mathscr{J}$ iteration. Eq. (9) is approximately equal to eq. (10) [35] as:

$$v_{i,\tau}(\omega, m) = \psi_{i,\tau} \mathscr{N}\left(\widehat{\varphi}(\omega, m; \tau)|\varepsilon_{i,\tau}(\omega), \sigma^2_{i,\tau}(\omega)\right) \tag{10}$$

$\mathscr{N}$ represents Gaussian distribution. As $z_{i\tau}(\omega, m)$ is a binary random variable, the probability $v_{i,\tau}(\omega, m)$ is equal to its expectation; hence it is the 'expectation or E' step. This expectation is used in the M−step, to estimate maximum likelihood parameters as the weighted means of sufficient statistics.

In the M−step, new model parameters $\widehat{\theta}$ are estimated from all the TF points by the formulae given in eqs. (11) to (13).

Let the operator $< x >_{m,\tau} = \frac{\sum_{m,\tau} x v_{i,\tau}(\omega,m)}{\sum_{m,\tau} v_{i,\tau}(\omega,m)}$ be the weighted mean over the specified variables m and τ. Using this notation, the IPD mean and variance is given as:

$$\varepsilon_{i,\tau}(\omega) = < \widehat{\varphi}(\omega, m; \tau) >_{\omega,m,\tau} \tag{11}$$

$$\sigma^2_{i,\tau}(\omega) = \ll \left(\widehat{\varphi}(\omega, m; \tau) - \varepsilon_{i,\tau}(\omega)\right)^2 \gg_{\omega,m,\tau} \tag{12}$$

$$\psi_{i,\tau} = \frac{1}{\Omega \mathscr{M}} \sum_{\omega,m,\tau} v_{i,\tau}(\omega, m) \tag{13}$$

Where Ω represents the total number of frequency bins and $\mathscr{M}$ represents the total number of time bins of spatial spectrogram. In addition to estimating the model parameters, the probabilistic IPD mask for the source s_i is given by marginalizing $v_{i,\tau}$ over delay τ:

$$M_{iIPD} = \sum_{\tau} v_{i,\tau} \tag{14}$$

b. ILD Cues Processing

In STEM, the TD3 agent is used for generating the ILD mask (M_{1ILD}) of the target source only. The trained RL agent takes the observations (the ILDs of a single time bin) from the ILD spectrogram one by one and generates an action vector for each of them. Let the ILD spectrogram be assumed as a matrix of observations given as:

$$ILDspectrogram = [O_1, O_2, O_3, \cdots, O_m, \cdots, O_{\mathscr{M}}] \tag{15}$$

where m(lies in the range $(1, \mathscr{M})$) is the time-bin index of the spectrogram. Each observation $O_m = \left[o_{-\Omega/2}, \cdots, o_0, \cdots, o_{+\Omega/2}\right]^T$ is a column vector containing ILDs of a single time bin m. o_0 is the ILD observation at DC, while $o_{-\Omega/2}$ and $o_{+\Omega/2}$ respectively are observations at the maximum negative and positive frequencies obtained from the Ω point STFT spectrograms by following the eqs (1) to (3). The size of O_m is therefore Ω x 1.

For each observation O_m, the action generated by the RL agent is a column vector given as $A_m = \left[a_{-\Omega/2}, \cdots, a_0, \right]^T$. The size of A_m is therefore $\left(\frac{\Omega}{2} + 1\right)$ x 1, containing the probabilistic mask for TF units lying on the negative side of the frequency spectrum including DC. As the magnitude spectrogram of any audio signal is symmetric around DC, there is no need to generate the mask for both positive and negative frequencies [63]. The actions produced by the TD3 agent are flipped around DC (shown in Fig. 1 by the dotted action flipping line) to generate the mask for the positive spectrum. This helps in reducing the size of the action space that improves the stability of the RL agent. The extended action vector A_{em} (after flipping around DC) is given as $A_{em} = \left[a_{-\Omega/2}, \cdots, a_0, \cdots, a_{+\Omega/2}\right]^T$. The ILD mask of the target source s_1 is obtained by concatenating the A_{em} s of all observation vectors O_m s of the ILD spectrogram (in eq. (15)) in a single matrix given as:

$$M_{1ILD} = [A_{e1}, A_{e2}, A_{e3}, \cdots, A_{em}, \cdots, A_{e\mathscr{M}}] \tag{16}$$

The probabilistic ILD mask (M_{2ILD}) of the interfering source s_2 is obtained by applying the probability rule of complementary events by subtracting the ILD mask of s_1 from a unit matrix $\mathscr{J}_{\Omega,\mathscr{M}}$ (matrix of all ones) as:

$$M_{2ILD} = \mathscr{J}_{\Omega,\mathscr{M}} - M_{1ILD} \tag{17}$$

c. Product Mask for Source Retrieval

The ILD and the IPD masks for each source are multiplied element-wise (Hadamard product) to obtain a product mask for each source, as shown in Fig. 1. These masks are later multiplied by the STFT matrices of the left and right mixtures (also by Hadamard product) and the inverse STFT (ISTFT) is applied to convert the signal from the TF domain to the time domain. For source s_1, the process is given mathematically as:

$$s_1 = ISTFT(M_{1ILD}. \times M_{1IPD}. \times X_L(\omega, m)) + ISTFT(M_{1ILD}. \times M_{1IPD}. \times X_R(\omega, m)) \tag{18}$$

where $.\times$ is the Hadamard multiplier and $+$ is the vector addition operator. Using its respective ILD and IPD masks, the same procedure is used for the s_2 retrieval. These sources are then stored as WAV files for evaluation.

4. Experimental settings

4.1. Model training

The EM algorithm does not require any training prior to its use in STEM during the testing phase. However, the TD3 agent must be trained before it can be used in our proposed BSS system, shown in Fig. 1.

a. TD3 Training

The training algorithm of the MATLAB TD3 agent, used in the STEM BSS system is explained in [64] and its Simulink model is shown in Fig. 2.

During training, the RL agent observes the state (or Observation) S_j and takes action A_j, according to its policy π. The agent interacts with the environment in multiple steps $j = \{1, 2, \cdots\cdots N\}$, where N is the number of total steps in a single episode. Due to the randomness of the action chosen by an agent in each step, the step reward R_j and the episode reward are variable. The goal of an agent is to obtain the maximum cumulative reward over several episodes. All mixtures of training data are first concatenated to form a long audio file with left and right channels and then converted to an ILD spectrogram by the steps mentioned in the section 'problem formulation'. The resulting ILD matrices are transposed. The time step information is then appended as the first column with this transposed ILD matrix. Each row of this matrix acts as a single observation for the agent during the training. Thus the state space (also called the observation (Obs) space) of a TD3 agent used for STEM consists of ILD cues of a single time bin. The magnitude matrix of the STFT of only the left channel of the training audio is transposed and given to input # 2, while the magnitude and phase matrices of the STFT of the clean target (s_1 of training data) are separately stored in two MAT files after being transposed and given respectively at the input # 3 and 4. All four training matrices are according to the format used for input in model [65] and they are utilized to calculate the 'step reward' to be explained shortly.

As the probabilistic mask results in better speech quality of the retrieved source than the one produced by the binary mask [66], the action space of the TD3 agent of STEM is kept continuous. The lower and upper limits of the action space are set to 0 and 1 respectively. At each

Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

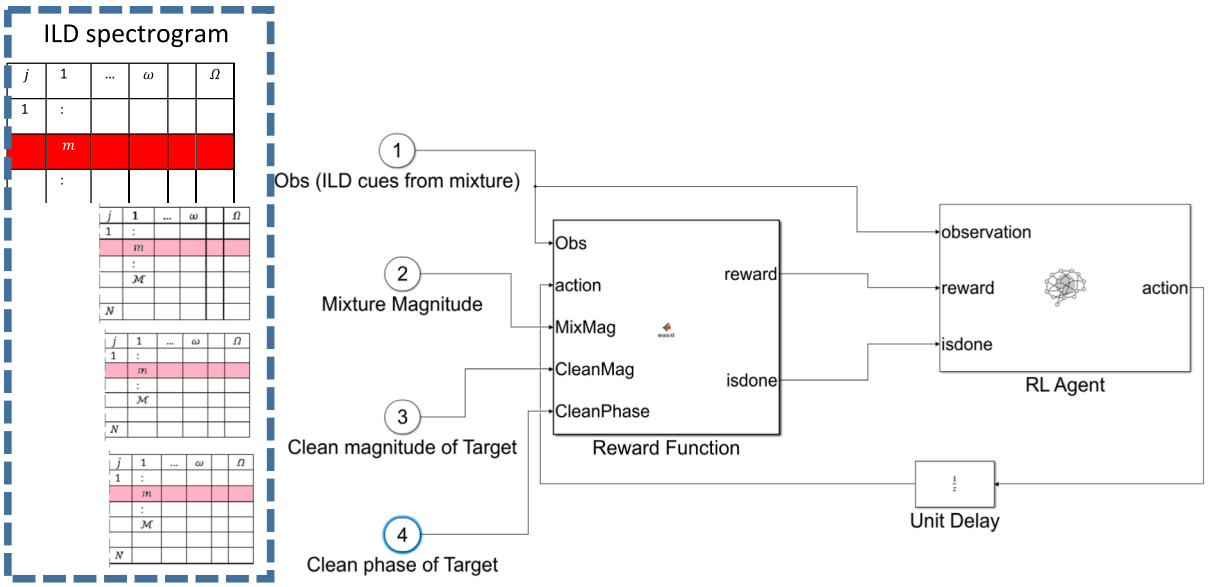

Fig. 2. Environment interacting the TD3 agent during the training phase.

sample time j, the agent observes the state S_j and performs an action A_j on the environment, which in turn calculates the step reward R_j.

b. Step Reward

Although it is tempting to use the perceptual metric (e.g. PESQ) directly as a reward function, it is inadvisable as it is affected both by the internal performance of the RL network and by the external conditions e.g. signal-to-noise ratio (SNR) [31]. So, inspired by the gaming model [67], the step reward function usually uses the difference between the perceptual scores. This relative score calculation for step reward has proved to generate better speech quality in RL-based SE models (e.g. [31;47]).

The step reward function of the TD3 agent for our proposed BSS model subtracts the PESQ score of the dirty target speech Z^{Dirty} from the PESQ score of the masked target's speech Z^{MSK} estimated from the TF units belonging to a single time bin. Mathematically the step reward is given as:

$$R_j = Z^{Dirty} - Z^{MSK} \tag{19}$$

The dirty speech is composed by combining the clean phases of the target (input through port # 4 in Fig. 2) with the magnitudes of the STFT of mixture's speech collected at the left microphone (input through port # 2). On the other hand, the masked speech is generated by the same procedure but after applying the mask (action of the agent in the previous time step) over the magnitudes of the mixture's speech (input through port # 2). The mask must be flipped around DC to make its length equal to the mixture's magnitude vector before they are multiplied together. This mask is fed back to the reward function after a unit delay. This delay is unavoidable due to the system's design constraints. As the dirty and the masked speech are using the same phase, the reward would remain negative until the mask is good enough to increase the PESQ score of the masked speech above the PESQ score of the dirty speech. The PESQ score calculation requires clean target speech as a reference, which is generated by combining the clean target magnitude (input port # 3) with the clean target phase (input port # 4).

c. Is-done Condition

The termination or is-done signal is set to zero in the TD3 agent of STEM, indicating no early termination condition. This is done intentionally so that each episode runs through the maximum number of predefined steps N. Also, the probability of early termination is set to zero by setting a very high value of the cumulative reward.

d. Reset Function

After an episode ends, the new episode starts with the environment reset conditions given in the reset function. According to the reset function defined for our TD3 agent, the agent extracts a mini-batch of M experiences with a random starting point from the experience buffer. However, it is ensured that only the magnitudes and phases of the mixture and target signal corresponding to that mini-batch are transferred from the base workspace to the Simulink environment to correctly calculate the step reward there.

The reset function for the test phase simply takes the time bins of the input ILD spectrogram sequentially, appends the time step information with them, and generates a probabilistic mask for each bin. These masks are then saved as columns of a matrix (ILD mask), which is then stored in a MAT file.

e. Actor and Critic Architectures

The TD3 agent is an advanced extension of a DDPG agent (an off-policy actor-critic method for environments with a continuous action space), which addresses its stability issues by introducing twin critic networks, delayed policy updates, and target policy smoothing [68]. It also features a target actor, two target critics, and an experience buffer. Both the actor and the critics must have the same structure and parameterization as their target counterparts. However, the two critics can have different structures, but for the best performance, it is recommended that they both have similar structures but different initial parameter values [64].

The neural network architectures of actor and critic networks of the TD3 agent (and their target counterparts) used for processing the ILD cues in our proposed BSS model are depicted in Fig. 3. The total learnable parameters of this agent is 404.7 K. Its observation space is a vector of size 64, while the action space is a vector of size 33 (negative side of the spectrum including DC) when the STFT is carried according to the parameters listed in the subsection "Short time Fourier transform (STFT) and expectation maximization (EM) parameters" of experimental settings.

f. TD3 Training Options

Before being used in STEM, the TD3 agents need to be trained. The training options of the TD3 agent are summarized in Table 1. Except for these, all other options of the TD3 agent are left unchanged at their default values defined in MATLAB 2024a.

4.2. Room Layout

As shown in Fig. 1, the source s_1 (target) is placed in front of the

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

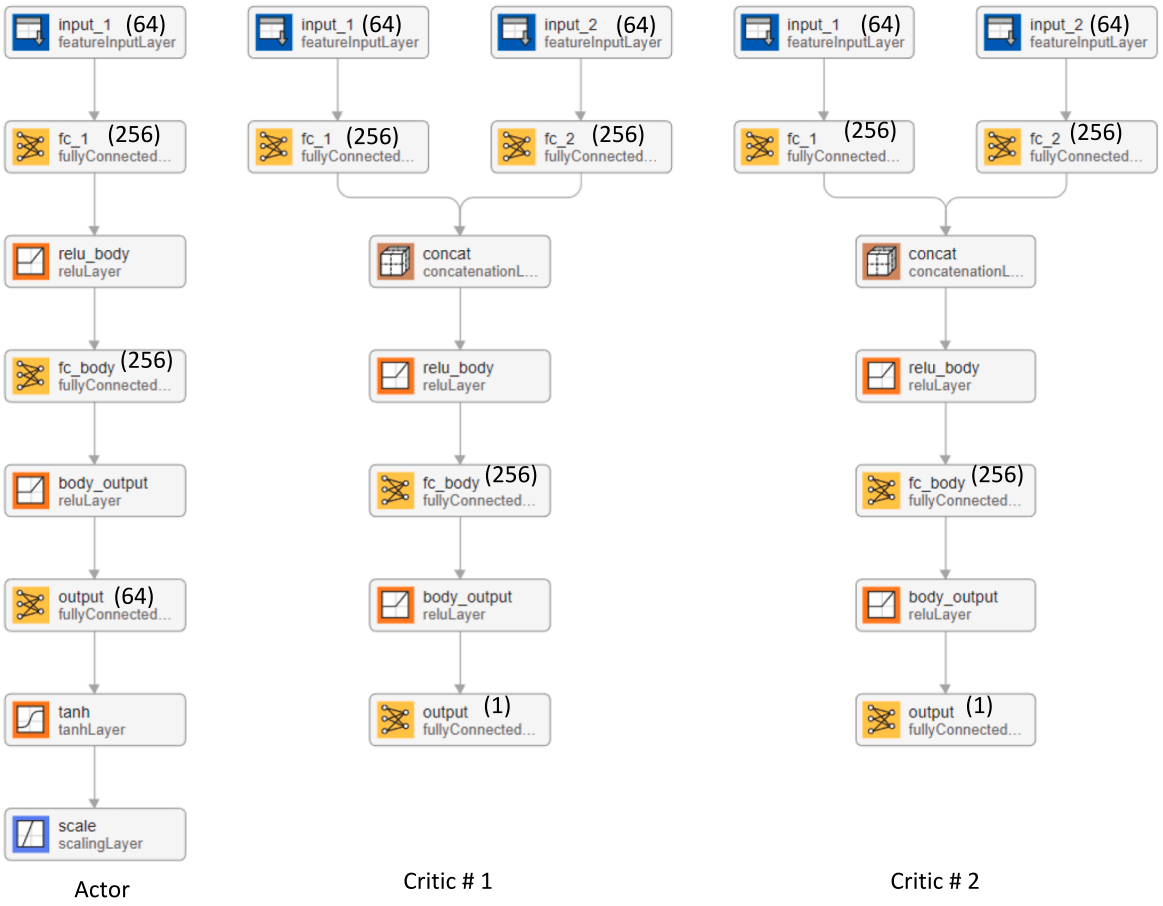

Fig. 3. Actor-Critic network architecture. The number (inside the brackets) within each block shows the number of neurons in each layer.

Table 1
TD3 training options.

	Parameter	Value
TD3 options	Mini batch size	128
	Experience buffer length	1e + 06
	Sample time	0.016
Actor and critic optimizer options	Optimizer	Adam
	Learn rate	5e-5
	Gradient threshold	10
RL training options	Maximum episodes	250
	Maximum steps per episode	50
	Score averaging window length	5
	Stop training criteria	Average reward
	Stop training value	200

binaural setup at the same elevation as that of the setup. The second source s_2 is also at the same elevation but separated from s_1 by an angle Θ that varies in the set (−90°:30°:90°) excluding 0°. The binaural room impulse responses (BRIRs) of Room X [69] are used for simulating the anechoic chamber, where the radial distance d between the sources and the binaural setup is 1.5 m.

4.3. Training and testing datasets

The dataset used for training STEM is Libri2mix [70]. This dataset consists of two subsets 1) train-360, having 212 h of training audios, and 2) train-100, with 58 h of audios. The sampling frequency of this dataset is 8 kHz. The EM block does not require any training. The TD3 agent is trained only on 1.5 h of data from the train-100 subset of Libri2mix and tested on test set with 1200 mixtures.

For STEM testing, two datasets are used: 1) Libri2mix (matched dataset; i.e. the same as used for its training) and 2) TIMIT [71] (the mismatched dataset). TIMIT speech corpus [71] is used to check the proposed system's generalization with other datasets. The sampling frequency of its audio files is reduced from 16 kHz to 8 kHz. Ten clean speech samples each of two seconds are mixed with ten different clean samples to form 10 speech mixtures that are used for testing the system's generalization for mismatched speakers at each separation angle θ.

4.4. Data pre-processing

Instead of using the available mixtures, the new mixtures are prepared for the training and testing phase from the clean sources themselves. The training mixture is prepared by first concatenating all s_1 and s_2 audio files separately to form two WAV files of 1.5 h each and then normalizing them and convolving them with the BRIRs according to the positions of s_1 and s_2 inside the room. Six such mixtures are generated with s_1 placed at 0° and s_2 at one of the six angular positions from the set {+90°, +60°, +30°, −30°, −60°, −90°}.

As the audio files given in the test set of Libri2mix are of random duration, each file is first clipped to two seconds. 1200 such pairs are first created which are divided into six subsets. Then 200 mixtures are created from each subset for each target-interferer separation. These mixtures are then converted to the ILD spectrograms and each spectrogram is stored as a separate MAT file. As opposed to the training data, there is no need to transpose or append the time step information with these matrices as the reset function for the test phase will do the job. The mixtures used for STEM testing are different from those given in the original dataset [70]. In contrast to already available mixtures in the test

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

set [70], which have a mean TIR of 0 dB and a standard deviation of 4.1 dB, our mixtures have a fixed TIR of 0 dB. The testing mixtures of the TIMIT dataset are also generated at 0 dB TIR from the clean TIMIT sources after their normalization and convolution with the BRIRs according to their location.

4.5. Evaluation metrics

The evaluation metrics used for ablation studies and comparison with the baseline algorithms are signal-to-distortion ratio (SDR) [72], scale-invariant source-to-noise ratio (SI-SNR (a variant of SDR, also called SI-SDR [73])) [74], perceptual evaluation of speech quality (PESQ) [75], and short-time objective intelligibility (STOI) [76]. SDR and SI-SNR are expressed in decibels (dBs). The PESQ score lies in the range $(-0.5, 4.5)$ while the STOI in (0, 1). Either these metrics or their deltas (Δ) are used, depending on the other baseline algorithms used for comparison in this paper, as their results are reported directly from their respective papers. The superscript '*mix*' on Δ (Δ^{mix}) shows the change in metrics over the raw mixture while the superscript '$+90'$ on Δ (Δ^{+90}) shows the change with respect to the results of testing, when the two sources are separated by 90° (s_1 placed at 0° and s_2 at + 90°). For all metrics, higher is better. For computational cost comparison, the number of parameters (in millions), multiply-accumulate operations (MACs) (in billions), required training data (in hours), training duration (in hours), separation time (in seconds), and real-time factor (RTF) are used. The RTF is given as $\frac{processingtime}{speechduration}$ [81]. However, as there are differences in the processors of all algorithms, the RTFs given in the papers of baseline algorithms are adjusted according to the processing speed of our GPU. For all these metrics, lower is better.

4.6. Short time Fourier transform (STFT) and expectation maximization (EM) parameters

The input to both modules (TD3 agent and EM) of STEM must be in the form of spatial spectrograms. For this purpose, short time Fourier transform (STFT) is required. Also, the maximum number of iterations for the EM algorithm needs to be set a priori, otherwise, the algorithm continues switching back and forth between the E and the M steps, until the parameter estimation has converged [77], which may take a random amount of time for each mixture and separation angle θ. The STFT parameters used for converting the time domain signal to the TF domain and the EM parameters used for generating the IPD mask are listed in Table 2.

The number of DFT points is restricted to 64 to avoid the agent crash during training, which occurs if the observation space is very large.

4.7. Baseline algorithms

Our proposed model is compared to the four recently proposed SOTA BSS models. These models are chosen as they are trained and tested on the same datasets used for STEM. The first model used for comparison is ESEDNet [78]. ESEDNet is an end-to-end time domain speech separation model that accommodates short-encoded sequences by using an encoder with multiple convolutions and down-sampling operations to reduce the length of high-resolution sequences. The decoder of ESEDNet reconstructs the high-quality target speech by using the encoded features provided through skip connections directly from the encoder side. ESEDNet overcomes the limitations of previously presented encoder-decoder frameworks which required a small convolutional kernel size responsible for their increased complexity and computational cost. ESEDNet combines the encoder-decoder framework with a multi-temporal resolution transformer separation network for extracting the contextual information, exhibiting outstanding separation accuracy with a very small number of parameters and reduced computational cost.

The second model used for comparison is [79], a single-channel diffusion-based source separation model. The neural network is first trained to approximate the score function of the marginal probabilities of the diffusion-mixing process. This network is then used to solve the reverse time stochastic differential equation that progressively separates the sources. The model uses the network architecture of Flow++ [80].

The third model [81] is also a diffusion-based model, which utilizes the forward process of the diffusion model and the reconstruction objective of the discriminative model. This model also uses the architecture of Flow++ with slight modifications.

The fourth model [82] is based on a frozen self-supervised pre-trained model, followed by two lightweight modules for the downstream tasks of speaker verification and target speech extraction. Although many SSL models have been tested, the best performance is achieved using the WaveLM pre-trained model [83].

Table 2
STFT and EM parameters.

Parameters	Values
Window Shape	Hamming
Window length	64 points
Number of discrete Fourier transform (DFT) points	64 samples
Overlap length	50 % (32 samples)
Sampling frequency	8 KHz
Training data duration	1.5 h
Testing data duration	1200 clips
EM iterations	16
Number of speech sources	2

5. Experiments and results

5.1. Ablation studies

For all ablation studies, the testing is carried out on ten speech mixtures. The sources used for generating these mixtures belong to the two speakers taken from the test dataset. For all experiments, the agent is trained for 250 episodes with the target placed at 0° and the interferer at +90° relative to the binaural setup, until stated otherwise.

5.2. Effect of agent

MATLAB 2024 has four off-policy RL agents with continuous action space. These are 1) twin-delayed deep deterministic (TD3) policy gradient, 2) deep deterministic policy gradient (DDPG), 3) soft actor-critic (SAC), and 4) model-based policy optimization (MBPO). The MBPO agent is trained with the TD3 agent as its base agent. All of them are compared in Table 3 to decide the most suitable agent for designing STEM. The best results are boldfaced.

Although the improvement offered by different agents over the unprocessed data is almost similar, the minimum training duration favours using TD3 or SAC. So, TD3 is selected for STEM to process the ILD cues.

5.3. Mask generation method

In this study, the combination of methods from the set {TD3, EM},

Table 3
Performance comparison of different RL agents.

Agent	Δ^{mix} SDR (dB)	Δ^{mix} SI-SNR (dB)	Δ^{mix} PESQ	Δ^{mix} STOI	Training duration (minutes)
TD3	9.41	4.1	0.73	**0.15**	**32**
DDPG	9.16	3.87	0.70	0.14	87
SAC	9.11	**4.3**	**0.76**	0.14	33
MBPO (TD3)	**9.92**	4.27	0.65	0.12	493

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

which can provide better clustering for the ILD and the IPD cues of each source is investigated and the results are listed in Table 4.

As clear from Table 4, the masks generated by the combination of TD3 and EM (for ILD and IPD cues respectively) have generated the maximum gain over unprocessed data when compared to all other combinations. So, for comparison to other baseline algorithms, this combination is used.

5.4. Effect of number of episodes

For this study, the TD3 agent is trained for: 1) 250, and 2) 500 episodes. The performance is compared in Table 5.

As can be seen in Table 5, doubling the number of episodes results in almost 8 times increase in training duration without any noticeable improvement in performance. So, in all future experiments the number of episodes is set to 250.

5.5. Effect of target perturbations

As the binaural cues are location-specific, the agent trained for a specific separation is tested for slight perturbation of the target position to the left and right of its original training position. Here the change Δ in different metrics is given relative to the results of testing without any perturbation, i.e. the target placed at 0° and interferer at 90°. The results are listed in Table 6.

As is clear from Table 6, except for $\pm 05^0$, the performance of the system sharply declines for all types of perturbations towards the left or right of the original position of training. At $+05^0$, the performance is almost similar to the performance without any perturbation. However, the case is not the same for perturbation by -05^0, where there is an improvement in SDR by 3.26 dB, and a slight decline in other metrics, but not to the extent of perturbations of higher angular values. The rise in SDR may be due to the increased separation between the target and the masker (from 90^0 to 95^0), increasing the Euclidean distance between the spatial cues of both sources and making it easier generally for all spatial-cue based separation networks to discriminate them [35, 37, and 42]. At higher perturbation angles, there is a severe deterioration of performance over all metrics. This is not astonishing because the system is spatial-cue-based, and perturbations in target position result in changing these cues, which in turn result in a decline in the system's performance. Spatial cue-based systems assume the sources to be stationary, which limits their use in real situations where the sources are in motion [84]. This is because once trained, the model parameters are not updated according to the changes in the source's location [85]. In future, tracking the moving target source by using spatial filtering (beamforming) in each time instance according to its new position on the azimuth [86] or embedding the video cues with the audio cues as done in the BSS model [87] and rotating the mannequin head accordingly may cease the decline in performance as the target would always be on the midsagittal plane.

Table 4
Methods used for generating ILD and IPD masks.

ILD mask	IPD mask	Δ^{mix} SDR (dB)	Δ^{mix} SI-SNR (dB)	Δ^{mix} PESQ	Δ^{mix} STOI
TD3	None	3.8	1.96	0.19	0.06
EM	None	5.39	−8.0	−0.52	−0.21
None	EM	0.1	−6.5	−0.43	−0.4
None	TD3	1.7	−0.04	0.1	0.04
EM	TD3	1.66	−0.04	0.09	0.03
TD3	TD3	1.55	0.24	0.1	0.04
EM	EM	7.4	−17.82	−0.53	−0.44
TD3	EM	**9.1**	**2.4**	**0.73**	**0.13**

Table 5
Effect of varying the number of episodes.

Total episodes	Δ^{mix} SDR (dB)	Δ^{mix} SI-SNR (dB)	Δ^{mix} PESQ	Δ^{mix} STOI	Training duration (minutes)
250	**9.4**	4.1	**0.72**	**0.14**	**32**
500	9.3	**4.3**	0.71	**0.14**	264

Table 6
Effect of perturbations to the left and right of the original position of target.

Perturbation Angle	Towards	Δ^{+90} SDR (dB)	Δ^{+90} SI-SNR (dB)	Δ^{+90} PESQ	Δ^{+90} STOI
$+05^0$	Right	−0.27	0.0	−0.08	0.0
$+10^o$	Right	−22.3	−20.4	−1.1	−0.63
$+15^o$	Right	−22.9	−22.4	−1.1	−0.60
-05^0	Left	3.26	−7.3	−0.8	−0.4
-10^o	Left	−21.8	−31	−1.2	−0.67
-15^o	Left	−21.8	−28	−1.2	−0.68

6. Required number of trained agents

In this study, two experiments are conducted. In the first experiment, six agents are trained one for each of the interferer positions from the set {+90°, +60°, +30°, −30°, −60°, −90°}. For all these agents, the target position is however fixed at 0°. Such customized networks are tested with the target and interferer placed at the same positions as were kept during the training of each network. In the second case, only one network is trained with the target at 0° and the interferer at + 90°, and this network is tested at all six positions of the interferer. This solution would be called the 'generalized' solution. The gains provided over the unprocessed mixtures by both types of solutions are listed in Table 7.

When compared with the customized solution, although there is a trivial difference in the output sound quality, the time-saving achieved by training a single network favours the use of the generalized solution. So, only the generalized pretrained network is used for testing at all interferer positions when the comparison is made to other baseline algorithms. It is also observed that the performance in the generalized solution for the target source is not declined by varying the interferer position as was witnessed in the previous ablation study 'effect of target perturbations'. It is because although the interferer position (during testing) mismatches with its position during training, the target position remains fixed as it was during the training of the generalized network. Due to the W-disjoint orthogonality phenomenon for speech [88], the probability of each TF unit of an audio mixture being occupied by the ILD cues of a single source is very high. Even when the interferer position changes, there is no change in the target position and hence no

Table 7
Using customized and generalized networks for testing.

Target/ Interferer placement during testing	Target/ Interferer placement during training	Solution	Δ^{mix} SDR (dB)	Δ^{mix} SI-SNR (dB)	Δ^{mix} PESQ	Δ^{mix} STOI
$0^o/+90^o$	$0^o/+90^o$	Customised	9.22	4.4	0.78	0.14
	$0^o/+90^o$	Generalized	9.22	4.4	0.78	0.14
$0^o/+60^o$	$0^o/+60^o$	Customised	9.98	4.05	0.91	0.16
	$0^o/+90^o$	Generalized	9.97	4.36	0.92	0.16
$0^o/+30^o$	$0^o/+30^o$	Customised	9.77	2.34	0.98	0.15
	$0^o/+90^o$	Generalized	9.56	1.91	0.99	0.14
$0^o/-30^o$	$0^o/-30^o$	Customised	8.05	3.66	0.98	0.11
	$0^o/+90^o$	Generalized	7.85	2.66	0.86	0.12
$0^o/-60^o$	$0^o/-60^o$	Customised	7.5	1.8	0.78	0.12
	$0^o/+90^o$	Generalized	7.49	1.79	0.78	0.12
$0^o/-90^o$	$0^o/-90^o$	Customised	9.28	4.6	1.0	0.33
	$0^o/+90^o$	Generalized	9.45	4.2	0.99	0.33

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

change in its ILD cues. The agent in the generalized solution is trained for these cues and would not face any difficulty in recognizing them in the presence of changes in the interferer cues (due to its position change during testing) resulting in almost similar performance, as would be provided by the customized solution.

7. Generalization for other datasets

In our final ablation study, STEM trained with target-interferer separation of 90° is tested with 10 mixtures generated from the TIMIT speech corpus with the same target-interferer settings as were there during the training phase. The results are given in Table 8.

Except for SI-SNR, the gains for all other metrics on the TIMIT mixtures are slightly better than those of the matched corpus (i.e. Libri2mix). As already discussed most recent BSS models e.g. [25–29] fail to generalize well with the mismatched speech speakers but as STEM is based entirely on spatial cues, it is minimally affected by the changes in the spectral contents of speech.

7.1. Comparison to baseline algorithms

For comparison with other algorithms, the pretrained network with target-interferer separation of 90° (during training) is used for testing at all the six interferer locations on our full test dataset of Libri2mix corpus (1200 mixtures). The results are listed in Table 9.

Although the state-of-the-art (SOTA) models used for comparison with STEM are trained over much larger datasets, it was observed that increasing the dataset size for STEM has resulted in a substantial increase in training time, but has not further improved the results. The exact reason requires further investigation, but it appears that it is due to STEM being based on an offline RL agent. Contrary to the supervised deep learning models, where the generalization is generally improved by increasing the amount of training dataset, data augmentation, transfer learning, regularization strategies, and the number of training epochs, in an offline RL, the errors accumulate over each iteration of training, resulting in the problem called 'unlearning/ overfitting effect' [89]. Increasing the size of the dataset does not help avoid this problem. As the Q-function is trained over longer durations, the target values become more erroneous, resulting in the degradation of the entire Q-function. Secondly, if all the states and actions fed to the Q-function for target-value calculation are in-distribution to the training dataset, the errors do not accumulate in the Q-function, and generalization is achieved [89]. In case of STEM, the presence of twin critic networks and delayed policy updates in the TD3 agent ensures policy smoothing [68], and reduction in the accumulated errors due to out-of-distribution inputs, which results in better generalization [89] as evident from our results on the mismatched dataset (the dataset which does not belong to the speech corpus used for training).

As is clear from Table 9, our proposed model outperforms all other baseline systems in terms of compactness (least number of trainable parameters), required data and time for training, slightly improved RTF, and smallest MACs. This makes it an ideal choice for handheld, edge, and Internet of Things (IoT) devices [90]. Our PESQ score is better than the SSL model [82] and almost similar to the diffusion-based models [79] and [81], but our STOI slightly lags behind that obtained from the model [82]. However, when compared to others, the SDR and SI-SNR of STEM are much lower. Particularly for STEM, the reason for the lower values of SDR and SI-SDR is explained in the next section, but it should be noted that generally for any source separation system, the SDR score shows a lower correlation with the perceptual quality than the PESQ score, and this correlation is further reduced in the case of SI-SDR [91]. Therefore, the STEM's unsatisfactory performance over these metrics should not be of much concern when its PESQ score is comparable to those of other baseline systems. In addition, it takes a little longer to process a test sample by STEM than by the overall best model [78]. This may be due to the differences in the processors used by the two models. Although it is difficult to attribute the lesser separation time of the BSS model [78] to its implementation in Python until both STEM and [78] are simulated in Python and Matlab simultaneously and all resulting models are run over the same processor, yet for simpler machine learning models (STEM and [78] both are lightweight systems) python has an advantage over Matlab in execution speed [92]. The STEM offers the lowest RTF when compared to other systems, when their RTF values (mentioned in their respective papers) are adjusted according to our processor speed.

The spectrogram and time domain representation of clean and estimated speech signals of target and interferer sources and their audio mixture is depicted in Fig. 4.

Table 8
Performance of STEM for mismatched corpus.

Corpus	Δ^{mix} SDR (dB)	Δ^{mix} SI-SNR (dB)	Δ^{mix} PESQ	Δ^{mix} STOI
Libri2mix	9.41	4.07	0.72	0.15
TIMIT	9.87	3.43	0.83	0.23

8. Discussion and future work

There seem to be two main reasons for the lower gains in SDR and SI-SNR of STEM compared to other baseline algorithms. The first is that, unlike our proposed model which uses relative PESQ score as its step reward, the models [78,79], and [82] use the SI-SNR as their objective function during training, resulting in exceptionally high values of SI-SNR and its root metric i.e. SDR itself.

Secondly, other algorithms use the already available mixtures of Libri2mix corpus, while the speech mixtures are generated from scratch for STEM by first normalizing the sources. The normalization process reduces their loudness units full scale (LUFS). The sources used for Libri2mix have LUFS that are uniformly distributed in the range (−25, −33) [70]. On the other hand, it is found that LUFS of the sources used for STEM mixtures are distributed in the range (−39, −43) resulting in lower performance of STEM compared to other algorithms. The LUFS distribution of sources in Libri2mix and STEM is depicted in Fig. 5.

It is not possible to use the already available mixtures for STEM as the sources in Libri2mix [70] are mixed instantaneously without convolving with the BRIRs required for STEM. The STEM mixtures more closely depict the real-world scenarios as sources reaching the listener are usually separated physically and convolve with BRIRs before being mixed in his ears, making STEM a more suitable candidate for hearing aids.

In this paper, the audio mixtures containing only two clean fully overlapping sources are considered. However, in real life, the mixtures are often contaminated by reverberations and background noise. Also, there may be more than two simultaneous speakers with varying intensities. In the future, for universal separation, this system must be tested for other audio mixtures containing more than two speech or non-speech sources e.g. music, animal sounds, environmental sounds, and industrial sounds. Also, the effect of reverberations and varying background noise in the case of moving sources on the system's performance is required to be investigated.

9. Conclusion

In this paper, a novel BSS algorithm based on reinforcement learning is presented, which offers comparable performance to the other baseline algorithms in terms of PESQ and STOI at much reduced computational cost and required training dataset. The reduced number of trainable parameters makes it a favorable choice for mobile devices with limited computational resources. In the future, fusing other cues (e.g. video [2] or text [93]) and integrating other more efficient machine learning algorithms with RL agents may further enhance its performance and support moving sources.

Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

Table 9
Comparison with baseline algorithms.

Algo.	NVIDIA GPUs used	Δ^{mix} SDR (dB)↑	Δ^{mix} SI-SNR (dB)↑	PESQ ↑	STOI ↑	Paras (M)	Training dataset duration (hours)	Training epochs/ duration (hours)	Separation time (sec)	RTF ↓ (adjusted according to our processor speed)	MACs (G)↓
[78]	1 RTX 3090	14.08	**13.24**	**2.81**		2.31	58	120/18	**34**	31.05	7.13
[79]			9.6	2.58		31.4	38.6	1000/–			
[82]			11	2.04	**0.89**	94.7	58	200/14			
[81]	8 TESLA V100 32 GB		9.9	1.8		32	220	512/–			
STEM	**1 RTX 3050**	10.7	4	2.35	0.84	**0.4**	**1.5**	–-/**0.5**	60	**30**	**0.8**

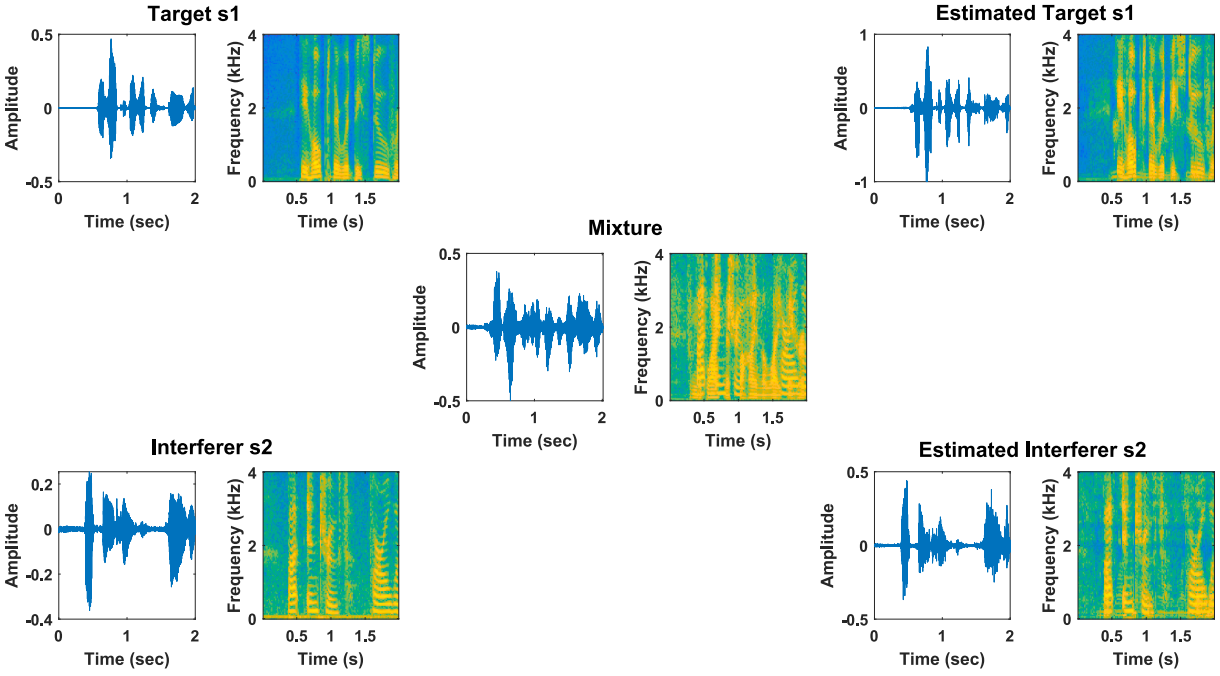

Fig. 4. Time domain and spectrogram representation of clean and estimated target and interferer signals and their mixture.

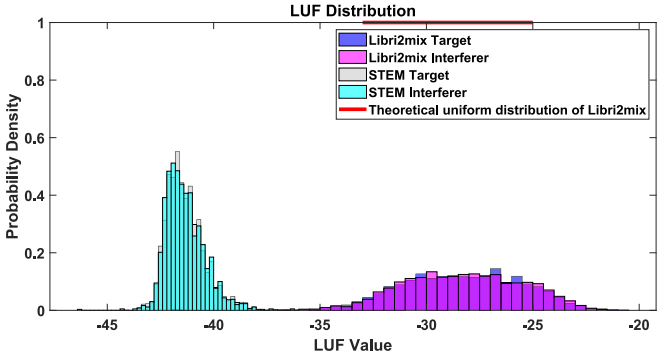

Fig. 5. LUFS distribution of Libri2mix and STEM sources.

CRediT authorship contribution statement

Muhammad Salman Khan: Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Sania Gul:** Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Open access dataset

References

[1] Cherry EC. Some experiments on the recognition of speech, with one and two ears. J Acoustic Soc Am 1953;25:975–9.
[2] Tao R, Qian X, Jiang Y, Li J, Wang J, Li H. Audio-visual target speaker extraction with selective auditory attention. IEEE Trans Audio Speech Language Process 2025.
[3] Haykin S, Chen Z. The cocktail party problem. Neural Comput 2005;17(9): 1875–902.
[4] Bronkhorst AW. The cocktail-party problem revisited: early processing and selection of multi-talker speech. Atten Percept Psychophys 2015 Jul;77(5): 1465–87.
[5] Essaid B, Kheddar H, Batel N, Chowdhury ME, Lakas A. Artificial intelligence for cochlear implants: review of strategies, challenges, and perspectives. IEEE Access 2024.
[6] Pal M, Roy R, Basu J, Bepari MS. Blind source separation: A review and analysis. In 2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/ CASLRE) 2013 Nov 25 (pp. 1-5). IEEE.

iranpaper Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

[7] Ansari S, Alatrany AS, Alnajjar KA, Khater T, Mahmoud S, Al-Jumeily D, et al. A survey of artificial intelligence approaches in blind source separation. Neurocomputing 2023;7(561):126895.
[8] Wan D, Su P, Kong Q. Research on a dual-element array sound source signal separation method based on compressed sensing theory. In International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024) 2024 Jun 13 (Vol. 13180, pp. 672-678). SPIE.
[9] Fraś M, Kowalczyk K. Reverberant Source Separation Using NTF With Delayed Subsources and Spatial Priors. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2024 Mar 6.
[10] Xu M, Li K, Chen G, Hu X. TIGER: Time-frequency Interleaved Gain Extraction and Reconstruction for Efficient Speech Separation. arXiv preprint arXiv:2410.01469. 2024 Oct 2.
[11] Drude L, Hasenklever D, Haeb-Umbach R. Integration of neural networks and probabilistic spatial models for acoustic blind source separation. IEEE J Select Top Signal Process 2019;13(4).
[12] Luo Y, Chen Z, Hershey JR, Le Roux J, Mesgarani N. Deep clustering and conventional networks for music separation: Stronger together. In 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP) 2017 Mar 5 (pp. 61-65). IEEE.
[13] Latif S, Cuayáhuitl H, Pervez F, Shamshad F, Ali HS, Cambria E. A survey on deep reinforcement learning for audio-based applications. Artif Intell Rev 2023 Mar;56 (3):2193–240.
[14] Doya K. Reinforcement learning: Computational theory and biological mechanisms. HFSP J 2007;1(1):30.
[15] Sutton RS. Reinforcement learning architectures. Proceedings ISKIT 1992;92.
[16] Barto AG. Reinforcement learning. In Neural systems for control 1997 Jan 1 (pp. 7-30). Academic Press.
[17] Mahadevan S. Average reward reinforcement learning: Foundations, algorithms, and empirical results. Mach Learn 1996;22(1):159–95.
[18] Glorennec PY. Reinforcement learning: An overview. In Proceedings European Symposium on Intelligent Techniques (ESIT-00), Aachen, Germany 2000 Sep 14 (pp. 14-15).
[19] Qiang W, Zhongli Z. Reinforcement learning model, algorithms and its application. In 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC) 2011 Aug 19 (pp. 1143-1146). IEEE.
[20] Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. Deep reinforcement learning: a brief survey. IEEE Signal Process Mag 2017 Nov 9;34(6):26–38.
[21] Prudencio RF, Maximo MR, Colombini EL. A survey on offline reinforcement learning: taxonomy, review, and open problems. IEEE Trans Neural Networks Learn Syst 2023.
[22] Reinforcement Learning Agents: https://www.mathworks.com/help/reinforcement-learning/ug/create-agents-for-reinforcement-learning.html.
[23] Yu D, Kolbæk M, Tan ZH, Jensen J. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2017 Mar 5 (pp. 241-245). IEEE.
[24] Zeghidour N, Grangier D. Wavesplit: End-to-end speech separation by speaker clustering. IEEE/ACM Trans Audio Speech Lang Process 2021 Jul;26(29):2840–9.
[25] Luo Y, Mesgarani N. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. IEEE/ACM Trans Audio Speech Lang Process 2019; 27(8):1256–66.
[26] Chen J, Mao Q, Liu D. Dual-path transformer network: direct context-aware modeling for end-to-end monaural speech separation. In Proc Interspeech 2020: 2642–6.
[27] Subakan C, Ravanelli M, Cornell S, Bronzi M, Zhong J. Attention is all you need in speech separation. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2021 Jun 6 (pp. 21-25). IEEE.
[28] Wang ZQ, Cornell S, Choi S, Lee Y, Kim BY, Watanabe S. TF-GridNet: Integrating full-and sub-band modeling for speech separation. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2023 Aug 11.
[29] Zhao S, Ma B. Mossformer: Pushing the performance limit of monaural speech separation using gated single-head transformer with convolution-augmented joint self-attentions. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2023 Jun 4 (pp. 1-5). IEEE.
[30] Wang H, Villalba J, Moro-Velazquez L, Hai J, Thebaud T, Dehak N. Noise-robust Speech Separation with Fast Generative Correction. arXiv preprint arXiv: 2406.07461. 2024 Jun 11.
[31] Koizumi Y, Niwa K, Hioka Y, Kobayashi K, Haneda Y. DNN-based source enhancement self-optimized by reinforcement learning using sound quality measurements. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2017 Mar 5 (pp. 81-85). IEEE.
[32] Arons B. A review of the cocktail party effect. J Am Voice I/O Soc 1992;12(7): 35–50.
[33] Snapp HA, Ausili SA. Hearing with one ear: consequences and treatments for profound unilateral hearing loss. J Clin Med 2020;9:1010. https://doi.org/10.3390/jcm9041010.
[34] Wei S, Zhang R. Underdetermined blind source separation based on spatial estimation and compressed sensing. Circuits Syst Signal Process 2024;43:2428–53.
[35] Mandel MI, Weiss RJ, Ellis DPW. Model-based expectation-maximization source separation and localization. IEEE Trans Audio Speech Lang Process 2010;18(2): 382–94.
[36] Hambrook DA, Ilievski M, Mosadeghzad M, Tata M. A Bayesian computational basis for auditory selective attention using head rotation and the interaural time-difference cue. PLoS One 2017;12(10):e0186104.
[37] Gul S, Fulaly MS, Khan MS, Shah SW. Clustering of spatial cues by semantic segmentation for anechoic binaural source separation. Appl Acoust 2021;1(171): 107566.
[38] Wang R, Fujimura T, Toda T. Target speaker extraction under noisy underdetermined conditions using conditional variational autoencoder, global style token, and neural postfilter. APSIPA Trans Signal Inf Process 2025;14(1).
[39] Tan K, Xu B, Kumar A, Nachmani E, Adi Y. SAGRNN: Self-attentive gated RNN for binaural speaker separation with interaural cue preservation. IEEE Signal Process Lett 2020 Dec;11(28):26–30.
[40] Wu Y, Li C, Yang S, Wu Z, Qian Y. Audio-visual multi-talker speech recognition in a cocktail party. In Interspeech 2021:3021–5.
[41] Lutati S, Nachmani E, Wolf L. Separate and diffuse: using a pretrained diffusion model for better source separation. In The Twelfth International Conference on Learning Representations. 2024.
[42] Gul S, Khan MS, Shah SW. Integration of deep learning with expectation maximization for spatial cue-based speech separation in reverberant conditions. Appl Acoust 2021 Aug;1(179):108048.
[43] Drude L, Hasenklever D, Haeb-Umbach R. Unsupervised training of a deep clustering model for multichannel blind source separation. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2019 May 12 (pp. 695-699). IEEE.
[44] Nugraha AA, Liutkus A, Member EV. Multichannel audio source separation with deep neural networks. IEEE/ACM Trans Audio Speech Lang Process 2016;24(9): 1652–64.
[45] Fakoor R, He X, Tashev I, Zarar S. Reinforcement learning to adapt speech enhancement to instantaneous input signal quality. arXiv preprint arXiv: 1711.10791. 2017 Nov 29.
[46] Alamdari N, Lobarinas E, Kehtarnavaz N. Personalization of hearing aid compression by human-in-the-loop deep reinforcement learning. IEEE Access 2020; 3(8):203503–15.
[47] Hao X, Xu C, Xie L, Li H. Optimizing the perceptual quality of time domain speech enhancement with reinforcement learning. Tsinghua Sci Technol 2022;27(6): 939–47.
[48] Zhou W, Ji R, Lai J. MetaRL-SE: a few-shot speech enhancement method based on meta-reinforcement learning. Multimed Tools Appl 2023;82(28):43903–22.
[49] Shen YL, Huang CY, Wang SS, Tsao Y, Wang HM, Chi TS. Reinforcement learning based speech enhancement for robust speech recognition. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2019 May 12 (pp. 6750-6754). IEEE.
[50] Bui HD, Chong NY. Autonomous speech volume control for social robots in a noisy environment using deep reinforcement learning. In 2019 IEEE international conference on robotics and biomimetics (ROBIO) 2019 Dec 6 (pp. 1263-1268). IEEE.
[51] Shen YL, Huang CY, Wang SS, Tsao Y, Wang HM, Chi TS. Reinforcement learning based speech enhancement for robust speech recognition. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2019 May 12 (pp. 6750-6754). IEEE.
[52] Subramanian AS, Wang X, Baskar MK, Watanabe S, Taniguchi T, Tran D, Fujita Y. Speech enhancement using end-to-end speech recognition objectives. In 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) 2019 Oct 20 (pp. 234-238). IEEE.
[53] Kumar A, Perrault A, Williamson DS. Using RLHF to align speech enhancement approaches to mean-opinion quality scores. In ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2025 Apr 6 (pp. 1-5). IEEE.
[54] Chu SC, Wu CH. Multi-step quality-oriented training for cross-dataset offline iterative speech enhancement. IEEE Access 2025.
[55] Liu G, Shi J, Chen X, Xu J, Xu B. Improving speech separation with adversarial network and reinforcement learning. In 2018 International Joint Conference on Neural Networks (IJCNN) 2018 Jul 8 (pp. 1-7). IEEE.
[56] Majumder S, Al-Halah Z, Grauman K. Move2hear: Active audio-visual source separation. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2021 (pp. 275-285).
[57] Majumder S, Grauman K. Active audio-visual separation of dynamic sound sources. In European Conference on Computer Vision 2022 Oct 23 (pp. 551-569). Cham: Springer Nature Switzerland.
[58] Remaggi L, Jackson PJ, Wang W. Modeling the comb filter effect and interaural coherence for binaural source separation. IEEE/ACM Trans Audio Speech Lang Process 2019;27(12):2263–77.
[59] Spoorthi GE, Gorthi S, Gorthi RK. PhaseNet: a deep convolutional neural network for two-dimensional phase unwrapping. IEEE Signal Process Lett 2018;26(1):54–8.
[60] Nustede EJ, Anemüller J. On the Generalization Ability of Complex-Valued Variational U-Networks for Single-Channel Speech Enhancement. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2024 Aug 15.
[61] Borman S. The expectation maximization algorithm: A short tutorial. 2004. URL http://www. seanborman. com/publications/EM_algorithm. pdf. 2012 Aug;32:68.
[62] Aarabi P. Self-localizing dynamic microphone arrays. IEEE Trans SystMan Cybernet—Part C: Appl Rev 2002;32(4).
[63] Gul S, Khan MS, Fazeel M. Single-channel speech enhancement using colored spectrograms. Comput Speech Lang 2024;1(86):101626.
[64] Twin-Delayed Deep Deterministic (TD3) Policy Gradient Agent: https://www.mathworks.com/help/reinforcement-learning/ug/td3-agents.html.
[65] Du Y, Li F, Kurte K, Munk J, Zandi H. Demonstration of intelligent HVAC load management with deep reinforcement learning: real-world experience of machine learning in demand control. IEEE Power Energ Mag 2022;20(3):42–53.

Downloaded from https://iranpaper.ir
https://www.tarjomano.com ترجمه تخصصی این مقاله مورد تایید ایران پیپر

[66] Chen P, Nguyen BT, Iwai K, Nishiura T. Threshold-based combination of ideal binary mask and ideal ratio mask for single-channel speech separation. Information 2024 Oct 4;15(10):608.
[67] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S. Mastering the game of Go with deep neural networks and tree search. nature. 2016 Jan;529(7587):484-9.
[68] Shen X. Comparison of DDPG and TD3 Algorithms in a Walker2D Scenario. In 2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023) 2024 Feb 14 (pp. 148-155). Atlantis Press.
[69] Michael I. Mandel. Binaural model-based source separation and localization. Ph.D. thesis. Columbia University, February 2010.
[70] Cosentino J, Pariente M, Cornell S, Deleforge A, Vincent E. Librimix: An open-source dataset for generalizable speech separation. arXiv preprint arXiv: 2005.11262. 2020 May 22.
[71] Lyons JW. DARPA TIMIT acoustic-phonetic continuous speech corpus. National Institute of Standards and Technology; 1993. http://www.ldc.upenn.edu/Catalog/LDC93S1.html.
[72] Vincent E, Gribonval R, Févotte C. Performance measurement in blind audio source separation. IEEE Trans Audio Speech Lang Process 2006;14:1462–9.
[73] Le Roux J, Wisdom S, Erdogan H, Hershey JR. SDR–half-baked or well done?. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2019 May 12 (pp. 626-630). IEEE.
[74] Isik Y, Roux JL, Chen Z, Watanabe S, Hershey JR. Single-channel multi-speaker separation using deep clustering. arXiv preprint arXiv:1607.02173. 2016 Jul 7.
[75] ITU (2007). Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs. ITU-T Recommendation P.862.2.
[76] Cees H. Taal, Richard C. Hendriks, Richard Heusdens, Jesper Jensen. A short time objective intelligibility measure for time-frequency weighted noisy speech. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Dallas, TX, USA; 2010.
[77] McLachlan GJ, Krishnan T. The EM algorithm and extensions. John Wiley & Sons; 2008.
[78] Liu D, Zhang T, Christensen MG, Ma B, Deng P. Efficient time-domain speech separation using short encoded sequence network. Speech Comm 2025;1(166): 103150.
[79] Scheibler R, Ji Y, Chung SW, Byun J, Choe S, Choi MS. Diffusion-based generative speech source separation. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2023 Jun 4 (pp. 1-5). IEEE.
[80] Ho J, Chen X, Srinivas A, Duan Y, Abbeel P. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In International conference on machine learning 2019 May 24 (pp. 2722-2730). PMLR.
[81] Zhang L, Qian Y, Yu L, Wang H, Yang H, Liu S, Zhou L, Qian Y. DDTSE: Discriminative diffusion model for target speech extraction. In 2024 IEEE Spoken Language Technology Workshop (SLT) 2024 Dec 2 (pp. 294-301). IEEE.
[82] Peng J, Delcroix M, Ochiai T, Plchot O, Ashihara T, Araki S, Černocký J. Probing self-supervised learning models with target speech extraction. In 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) 2024 Apr 14 (pp. 535-539). IEEE.
[83] Chen S, Wang C, Chen Z, Wu Y, Liu S, Chen Z, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE J Sel Top Signal Process 2022;16(6):1505–18.
[84] Han C. Automatic speech separation for brain-controlled hearing technologies. Columbia University; 2024.
[85] Abdipour R, Akbari A, Rahmani M, Nasersharif B. Binaural source separation based on spatial cues and maximum likelihood model adaptation. Digital Signal Process 2015;1(36):174–83.
[86] Nikunen J, Diment A, Virtanen T. Separation of moving sound sources using multichannel NMF and acoustic tracking. IEEE/ACM Trans Audio Speech Lang Process 2017;26(2):281–95.
[87] Phokhinanan W, Obin N, Argentieri S. Binaural Sound Localization in Noisy Environments Using Frequency-Based Audio Vision Transformer (FAViT). In INTERSPEECH 2023 Aug 20 (pp. 3704-3708). ISCA.
[88] Scott R. The DUET blind source separation algorithm. Dublin (Springer): University College; 2007. p. 217–41.
[89] Levine S, Kumar A, Tucker G, Fu J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643. 2020 May 4.
[90] Wang CH, Huang KY, Yao Y, Chen JC, Shuai HH, Cheng WH. Lightweight deep learning: an overview. IEEE Consum Electron Mag 2022;13(4):51–64.
[91] Torcoli M, Kastner T, Herre J. Objective measures of perceptual audio quality reviewed: an evaluation of their application domain dependence. IEEE/ACM Trans Audio Speech Lang Process 2021 Mar;29(29):1530–41.
[92] Ghanem AA. Machine Learning in Practice MATLAB vs. Italy, Oct: The University of Genoa's; 2024. Python. Master's thesis.
[93] Liu X, Kong Q, Zhao Y, Liu H, Yuan Y, Liu Y, Xia R, Wang Y, Plumbley MD, Wang W. Separate anything you describe. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2024 Dec 31.