Interspeech 2022
18-22 September 2022, Incheon, Korea

Improving Target Sound Extraction with Timestamp Information

Helin Wang[1†], *Dongchao Yang*[1†], *Chao Weng*[2], *Jianwei Yu*[2], *Yuexian Zou*[1,*]

[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]Tencent AI Lab, Shenzhen, China

zouyx@pku.edu.cn

Abstract

Target sound extraction (TSE) aims to extract the sound part of a target sound event class from a mixture audio with multiple sound events. The previous works mainly focus on the problems of weakly-labelled data, jointly learning and new classes, however, no one cares about the onset and offset times of the target sound event, which has been emphasized in the auditory scene analysis. In this paper, we study to utilize such timestamp information to help extract the target sound via a target sound detection network and a target-weighted time-frequency loss function. More specifically, we use the detection result of a target sound detection (TSD) network as the additional information to guide the learning of target sound extraction network. We also find that the result of TSE can further improve the performance of the TSD network, so that a mutual learning framework of the target sound detection and extraction is proposed. In addition, a target-weighted time-frequency loss function is designed to pay more attention to the temporal regions of the target sound during training. Experimental results on the synthesized data generated from the Freesound Datasets show that our proposed method can significantly improve the performance of TSE.

Index Terms: target sound extraction, target sound detection, timestamp information, mutual learning

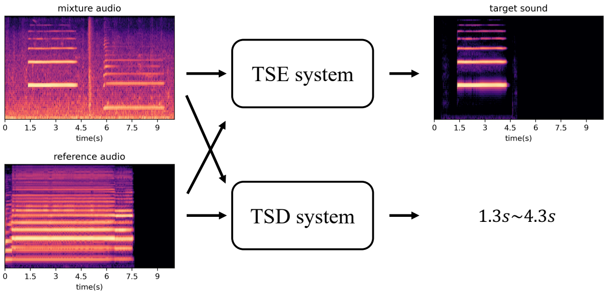

Figure 1: *Example of TSE and TSD tasks. Both input the mixture audio and a reference audio. TSE aims to directly output the target sound while TSD aims to output the timestamps that the target sound happen within the mixture audio.*

1. Introduction

A wide variety of sounds exist in the world, which provide critical information about our surroundings, such as the violin sound in a concert and klaxons on the street. Our daily lives can be greatly improved if we could develop hearing devices that have the ability of selecting the sound we are interested in and removing the sound we are not interested in. Such target sound extraction and removal applications have been studied these years in machine hearing, aiming at target speaker [1, 2, 3], music instruments [4, 5], and sound events [6, 7, 8, 9, 10]. In this paper, we focus on the target sound extraction problem, which is particularly challenging due to the large variety of sounds it covers (*e.g.* baby cry, telephone call, animal sounds, etc).

Target sound extraction (TSE) consists of extracting the sound of a target sound event class from a mixture audio, which uses an extraction network that estimates the target sound given the sound mixture and an embedding vector that represents the characteristics of the target sound. The embedding vector can be obtained using an embedding encoder that receives either an enrollment audio sample [8] or a one-hot vector that represents the target sound event class [10, 11]. The extraction neural network and the embedding encoders can be jointly trained. Several researchers further explored the problems of new classes and weakly-labelled data. For example, Delcroix *et al.* [7] combined 1-hot-based and enrollment-based target sound extraction to improve the performance of new classes. Pishdadian *et al.* [9] proposed a method to solve the weakly-supervised sound separation with a sound classification network, and Gfeller *et al.* [8] solved this problem by randomly mixing two sounds and using one as both the reference audio and the target sound. Although these previous works show good performance, no one cares about the timestamp information of the target sound, *i.e.* the onset and offset times. Such information is quite important in the auditory scene analysis, which has been widely studied like sound event detection [12, 13] and voice activity detection [14, 15], and can be explored to help guide the localization and extraction of target sound.

In this paper, we explicitly study to utilize the timestamp information for TSE, and propose to use an additional detection network along with a target-weighted time-frequency loss function. To be more specific, we can obtain the result of target sound detection (TSD) [16] given the mixture audio and the embedding vector, and this result is applied as the additional timestamp information. Then, the TSE network inputs the mixture audio, the embedding vector and the detection result, and outputs the extracted result. Furthermore, we find that the result of TSE can also help TSD, by using it as another input audio reference. In this way, a mutual learning framework is proposed to jointly improve the performance of target sound extraction and detection. In addition, in order to enhance the model's ability to deal with the overlapping parts of the target sound and others, we combine the time-domain and frequency-domain loss function and giving more weights to the overlapping timestamp when calculating the loss function, which is called a target-weighted time-frequency loss function. Experiments are carried out on the simulated data generated from sound events taken from the Freesound Datasets (FSDs) [17, 18]. Our proposed method can obtain an overall SI-SDR improvement of 11.81 dB and a SI-SDR improvement of 6.14 dB for the temporal regions

[†] Indicates equal contribution.
[*] Corresponding Author.

10.21437/Interspeech.2022-676

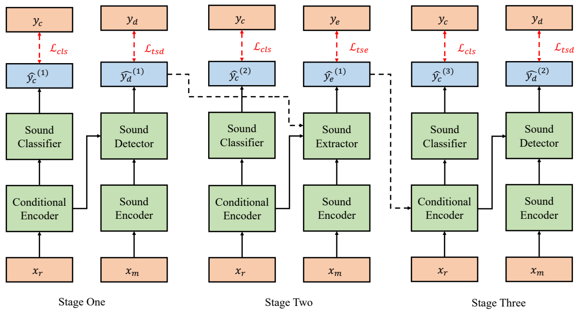

Figure 2: *Overall architecture of Tim-TSENet. In Stage One, we train a target sound detection network which outputs the predicted detection result. In Stage Two, we train a target sound extraction network using the predicted detection result from Stage One. Then in Stage Three, the target sound detection network is updated using the extracted result from Stage Two.*

in which the target sound happens, which provides 15.2% and 19.5% improvement over our baseline method, respectively.

2. Proposed Method

In this section, we will introduce our proposed method that uses the timestamp information for the target sound extraction, which is called Tim-TSENet. The architecture of Tim-TSENet is shown in Figure 2, which shows three stages for the training of target sound extraction and detection networks. Let's begin with a brief review of the baseline system. We then introduce the loss function and the mutual learning framework.

2.1. Target sound extraction and detection

Figure 1 shows an example of TSE and TSD tasks. Given a reference audio that represents the target sound event, TSE aims to extract the target sound from a mixture audio (a regression task), while TSD aims to get the detection result of the target sound (a classification task). Both need a conditional encoder to obtain the embedding vector of the reference audio and a sound encoder to extract acoustic features from the mixture audio. The difference is that TSE needs a sound extractor to output the estimated sound but TSD needs a sound detector to output the predicted scores. More specifically, given the reference audio $\boldsymbol{x_r} \in \mathcal{R}^{N_r}$ and the mixture audio $\boldsymbol{x_m} \in \mathcal{R}^{N_m}$, a conditional encoder $f_{con}(\cdot)$ is applied to get the embedding vector $\boldsymbol{e} \in \mathcal{R}^{N_e}$.

$$\boldsymbol{e} = f_{con}(\boldsymbol{x_r}; \theta_{con}) \tag{1}$$

where θ_{con} denotes the parameters of the conditional encoder. N_r, N_m and N_e are the dimension of the reference audio, the mixture audio and the embedding vector respectively. Then a sound encoder $f_{enc}(\cdot)$ is applied to get the acoustic features $\boldsymbol{a} \in \mathcal{R}^{N_t \times N_f}$.

$$\boldsymbol{a} = f_{enc}(\boldsymbol{x_m}; \theta_{enc}) \tag{2}$$

where θ_{enc} denotes the parameters of the sound encoder, N_t denotes the number of frames for the acoustic features and N_f denotes the dimension of each frame. For the TSE task, a sound extractor $f_{ext}(\cdot)$ is finally employed to obtain the estimated target sound $\boldsymbol{\hat{y_e}} \in \mathcal{R}^{N_m}$.

$$\boldsymbol{\hat{y_e}} = f_{ext}(\boldsymbol{a}, \boldsymbol{e}; \theta_{ext}) \tag{3}$$

Here, θ_{ext} denotes the parameters of the sound extractor. Note that the dimension of $\boldsymbol{\hat{y_e}}$ is the same as $\boldsymbol{x_m}$. For the TSD task, a sound detector $f_{det}(\cdot)$ is applied to obtain the predicted scores of all the frames $\boldsymbol{\hat{y_d}} \in \mathcal{R}^{N_t}$.

$$\boldsymbol{\hat{y_d}} = f_{det}(\boldsymbol{a}, \boldsymbol{e}; \theta_{det}) \tag{4}$$

where θ_{det} denotes the parameters of the sound detector.

2.2. Loss function

For the sound separation tasks (*e.g.* speech separation [19, 20, 21, 22] and speech enhancement [23, 24]), there are two popular loss functions. One is the frequency-domain loss function, which starts with calculating the short-time Fourier transform (STFT) to create a time-frequency (T-F) representation of the mixture audio. The T-F bins that correspond to the target sound are then separated, and are used to synthesize the source waveform using inverse STFT. In this case, the loss function is formulated by the mean square error (MSE) between the T-F bins estimated target sound $\boldsymbol{\hat{s}} \in \mathcal{R}^{T \times F}$ and the corresponding ground truth $\boldsymbol{s} \in \mathcal{R}^{T \times F}$.

$$\mathcal{L}_{fmse} = \frac{1}{T \times F} \|\boldsymbol{s} - \boldsymbol{\hat{s}}\|^2 \tag{5}$$

where $\|\cdot\|^2$ denotes the l_2 norm, T denotes the number of frames and F denotes the number of frequency bins for each frame. Such frequency-domain is friendly to the computational complexity and shows good performance in many applications. However, the phase information is sometimes not totally considered [19], and suitable window size and hop size of STFT need to be designed. Another way is calculating the loss directly on the time domain, including the time-domain MSE loss and the time-domain scale-invariant source-to-distortion ratio (SI-SDR) loss [25, 26]. Given the estimated waveform $\boldsymbol{\hat{y_e}} \in \mathcal{R}^{N_m}$ and the corresponding ground truth $\boldsymbol{y_e} \in \mathcal{R}^{N_m}$, the time-domain MSE loss is formulated by:

$$\mathcal{L}_{tmse} = \frac{1}{N_m} \|\boldsymbol{y_e} - \boldsymbol{\hat{y_e}}\|^2 \tag{6}$$

Here, N_m denotes the dimension of the mixture audio. This time-domain MSE loss is sensitive to the variation of the energy of the waveform and can be used to limit the energy of the estimated waveform to the ground truth. In addition, the SI-SDR loss is defined as:

$$\mathcal{L}_{sisdr} = 10 \log_{10} \left(\frac{\left\| \frac{\langle \boldsymbol{\hat{y_e}}, \boldsymbol{y_e} \rangle}{\langle \boldsymbol{y_e}, \boldsymbol{y_e} \rangle} \boldsymbol{y_e} \right\|^2}{\left\| \frac{\langle \boldsymbol{\hat{y_e}}, \boldsymbol{y_e} \rangle}{\langle \boldsymbol{y_e}, \boldsymbol{y_e} \rangle} \boldsymbol{y_e} - \boldsymbol{\hat{y_e}} \right\|^2} \right) \tag{7}$$

where $\langle \cdot, \cdot \rangle$ is the inner product. To ensure scale invariance, the signals $\boldsymbol{\hat{y_e}}$ and $\boldsymbol{y_e}$ are normalized to zero-mean prior to the SI-SDR calculation. This loss is designed to directly optimize the evaluation metric (*i.e.* source-to-distortion ratio [27]) and has widely used in speech separation. Based on the complementary of these three loss functions, we propose to combine them, which forms a time-frequency loss functions for the TSE task.

$$\mathcal{L}_{tse} = \mathcal{L}_{fmse} + \alpha \mathcal{L}_{tmse} + \beta \mathcal{L}_{sisdr} \tag{8}$$

where α and β are the hyper-parameters to balance the loss function. Such loss function treats the whole audio equally, however, for a TSE task, we focus more on the temporal regions in which the target sound happens with in the whole audio clip. In this paper, in order to enhance the model's ability to deal with such regions, we propose a target-weighted time-frequency loss function that gives more weights to the overlapping timestamps when calculating the loss. Assume that the target sound happens in the timestamp $[N_{m_1}, N_{m_2}]$ of the mixture audio where $N_{m_2} \leq N_m$, we can get the losses $\mathcal{L}_{fmse-t}$ (target-region frequency-domain MSE loss), $\mathcal{L}_{tmse-t}$ (target-region time-domain MSE loss) and $\mathcal{L}_{sisdr-t}$ (target-region time-domain SI-SDR loss) according to equations (5-7). The target-weighted time-frequency loss function is then calculated by:

$$\mathcal{L}_{tse-t} = \mathcal{L}_{fmse-t} + \alpha\mathcal{L}_{tmse-t} + \beta\mathcal{L}_{sisdr-t} \tag{9}$$

$$\mathcal{L}_{tse-w} = \mathcal{L}_{tse} + \tau\mathcal{L}_{tse-t} \tag{10}$$

Here, τ is another hyper-parameter to control the ratio of the weighted part. For the TSD task, given the predicted score $\hat{\boldsymbol{y_d}} \in \mathcal{R}^{N_t}$ and the frame-level ground truth label $\boldsymbol{y_d} \in \mathcal{R}^{N_t}$, the TSD network can be optimized by minimize the binary cross-entropy (BCE) loss function:

$$\mathcal{L}_{tsd} = \sum_{i=1}^{N_t}(-y_d^i \log \hat{y_d}^i - (1 - y_d^i)\log(1 - \hat{y_d}^i)) \tag{11}$$

where $y_d^i \in \{0, 1\}$, $\hat{y_d}^i \in [0, 1]$ and N_t denotes the number of frames. Both TSD and TSE can be jointly optimized by a sound classification task to improve the quality of the embedding vector. As shown in Figure 2, a sound classifier $f_{cls}(\cdot)$ is applied to get the predicted classification result $\hat{\boldsymbol{y_c}} \in \mathcal{R}^C$.

$$\hat{\boldsymbol{y_c}} = f_{cls}(\boldsymbol{e}; \theta_{cls}) \tag{12}$$

where θ_{cls} denotes the parameters of the sound classifier and C denotes the number of classes. Given the clip-level ground truth label, $\boldsymbol{y_c} \in \mathcal{R}^C$, the cross-entropy loss function [28] is then employed.

$$\mathcal{L}_{cls} = \sum_{i=1}^{C}(-y_c^i \log \hat{y_c}^i - (1 - y_c^i)\log(1 - \hat{y_c}^i)) \tag{13}$$

Note that $y_c^i \in \{0, 1\}$ and $\hat{y_c}^i \in [0, 1]$. The final loss function for the TSE and TSD with jointly learning is:

$$\mathcal{L}_{tse-j} = \mathcal{L}_{tse-w} + \mathcal{L}_{cls} \tag{14}$$

$$\mathcal{L}_{tsd-j} = \mathcal{L}_{tsd} + \mathcal{L}_{cls} \tag{15}$$

2.3. Mutual learning framework

TSE and TSD tasks provide different views of the target sound within the mixture audio, where TSE focuses on the separation and TSD focuses on the timestamp information. Motivated by this, we propose a mutual learning framework to improve the two tasks. The TSE network can utilize the timestamp information from the TSD result to guide the separation of target sound, and the TSD network can use the extracted target sound from the TSE result as an additional reference audio to enhance the reference information. Figure 2 shows an example of three-stage training architecture. In Stage One, a TSD network is trained with equation (15) as the loss function. Then in Stage Two, this TSD network is fixed and we then train the TSE network by loss function (14). Here, we use the result of the TSD network $\hat{\boldsymbol{y_d}}^{(1)}$ as the additional timestamp information, and concatenate it with the embedding vector $\boldsymbol{e}$. So the equation (3) is modified by:

$$\hat{\boldsymbol{y_e}}^{(1)} = f_{ext}(\boldsymbol{a}, \boldsymbol{e}, \hat{\boldsymbol{y_d}}^{(1)}; \theta_{ext}) \tag{16}$$

Note that we upsample or downsample $\hat{\boldsymbol{y_d}}^{(1)}$ to the same number of frames as $\boldsymbol{a}$. Finally, for the Stage Three, we fix the TSE network and update the TSD network. Except the original reference audio $\boldsymbol{x_r}$, the result from the TSE network $\hat{\boldsymbol{y_e}}^{(1)}$ is also used in this stage. We can get two embeddings from the two audios, and we use the average value as the new embedding.

$$\boldsymbol{e}' = f_{con}(\hat{\boldsymbol{y_e}}^{(1)}; \theta_{con}) \tag{17}$$

$$\hat{\boldsymbol{y_d}}^{(2)} = f_{det}(\boldsymbol{a}, \frac{\boldsymbol{e} + \boldsymbol{e}'}{2}; \theta_{det}) \tag{18}$$

In this way, TSE and TSD can help each other and we argue that this training framework can be recycled to further improve the performance. In this work, considering the cost of training time, we only attempt three stages.

3. Experiments

3.1. Datasets

Following [11], we create datasets of simulated sound event mixtures based on the Freesound Dataset Kaggle 2018 corpus (FSD) [17, 18], which contains audio clips from 41 diverse sound event classes, such as human sounds, object sounds, musical instruments. The duration of the audio samples ranges from 0.3 to 30 seconds. We generate 10 seconds mixtures by randomly choosing one target sound and 1-3 interference sound clips from the FSD corpus and adding them to random time-positions on top of the 10 seconds background sound. We choose the background sound from the acoustic scene classification task of Detection and Classification of Acoustic Scenes and Events (DCASE) 2019 Challenge [29]. Furthermore, the signal-noise ratio (snr) is set randomly between -5 and 10 dB for each foreground sound. If the duration of the foreground sound is longer than 10 seconds, we only keep the first 10 seconds of it. In addition, we found that some foreground sounds contain silent zones, and we use a pre-processing algorithm [30] to remove these zones. We also record the onset and offset times of the target sound according to the time-positions and the duration information. The onset and offset information can then be used to train the target sound detection task. Note that the onset and offset information is only used in the training stage. For each mixture audio, we randomly choose a reference audio that has the same class with the target sound. We also pad or cut the reference audio to 10 seconds. In the experiments, we down-sample the audios to 16 kHz to reduce the computational and memory costs. The training, validation and test sets consist of 47356, 16000 and 16000 samples respectively.

3.2. Metrics

For the TSE task, we use the mostly-used scale-invariant signal-to-distortion ratio improvement (SI-SDRi). In addition, in order to evaluate the model's ability to extract the target sound in the temporal regions where it happens, we apply a novel metric called scale-invariant signal-to-distortion ratio improvement in the target sound regions (SI-SDRi-t). For the TSD task, the segment-based F-measure (segment-F1) and event-based F-measure (event-F1) [16] are employed.

3.3. Implementation details

For the sound extractor in Tim-TSENet, we use the Conv-tasnet with the original settings [26] due to its effectiveness and efficiency. For the sound detector, we use the same CRNN structure

Table 1: *Performance of different models. Here, 'f', 't', 's' and 'w' represents the frequency-domain MSE loss, the time-domain MSE loss, the time-domain SI-SDR loss and the target-weighted time-frequency loss respectively. All the experiments of Tim-TSENet use the jointly learning of TSE and a classification task. Note that the results of [7] and [11] are reproduced under the experimental setups with ours.*

Model	Loss function				SI-SDRi	SI-SDRi-t
	f	t	s	w		
Tim-TSENet (ours)	√	×	×	×	7.61	2.90
	×	√	×	×	4.39	2.31
	×	×	√	×	3.63	1.86
	√	√	×	×	7.44	3.14
	√	×	√	×	10.22	5.06
	×	√	√	×	9.22	4.68
	√	√	√	×	10.25	5.14
	√	√	√	0.2	10.88	5.65
	√	√	√	0.5	10.68	5.52
	√	√	√	1.0	10.77	5.65
	√	√	√	1.2	10.86	5.77
	√	√	√	1.5	**10.91**	**5.83**
	√	√	√	1.8	10.87	5.76
	√	√	√	2.0	10.79	5.63
[7]	×				9.87	4.69
[11]	×				10.02	4.74

as [16]. Besides, following [16], a four-block convolutional network [31] pretrained on Audioset [32] is applied as the conditional encoder for both TSE and TSD. The dimension of the embedding vector is 128. We use the Adam optimizer [33] with an initial learning rate of 1×10^{-3} and a weight decay of 1×10^{-5}. The whole training epochs are 60 for TSE and 100 for TSD. We carry out the experiments three times and report the mean value.

3.4. Experimental results

Table 1 reports the performance of our proposed Tim-TSENet and two state-of-the-art methods [7, 11]. We firstly test Tim-TSENet with different settings of loss function, *i.e.* the frequency-domain MSE loss, the time-domain MSE loss, the time-domain SI-SDR loss and the target-weighted time-frequency loss. For a single one among the first three loss, we can see that the time-domain SI-SDR loss preforms the worst. An interesting thing is that for the speech-related tasks [26], the time-domain SI-SDR loss is always the state-of-the-art method. We argue that the separation of speech and sound event has a obvious gap because of different sampling rate, different frequency distribution and different duration. By combining the time-domain and frequency-domain loss, we can get our baseline performance of 10.25 for SI-SDRi and 5.14 for SI-SDRi-t. Comparing with [7] and [11], our baseline method has already surpassed them as they only use the time-domain loss. In addition, by applying the target-weighted time-frequency loss, we can obtain better results for both the overlapping regions and the whole audio clip. With the ratio of τ increasing, the performance of TSE first raises then decreases, and the best performance can be achieved when τ is 1.5. If the ratio is too large, the model will focus too much on the regions that target sound happens, so that the suppression of other regions do not work well. Furthermore, we report the results of three-stage mutual learning framework for TSE and TSD tasks on Table 2. We find that each stage of mutual learning procedure can lead to the performance gain, especially for the first stage. After three stages, TSE could get a SI-SDRi of 11.81 and a SI-SDRi-t of 6.14, while TSD could get a segment-based F score of 87.05% and an event-based F score of 51.08%. Figure 3 gives an example of a test audio, where Figure 3(c) shows the spectrogram of the extracted result from our baseline model (without the target-weighted loss and mutual learning) and Figure 3(d) shows shows the spectrogram of the extracted result from Tim-TSENet. We can see that Tim-TSENet preforms much better than the baseline, especially for the overlapping regions. More demos can be found here.[1]

Table 2: *The influence of mutual learning for TSE and TSD tasks.*

Stage	TSE		TSD	
	SI-SDRi	SI-SDRi-t	segment-F1	event-F1
0	10.91	5.83	85.11%	43.61%
1	11.46	5.92	85.29%	49.26%
2	11.67	6.10	85.93%	50.31%
3	**11.81**	**6.14**	**87.05%**	**51.08%**

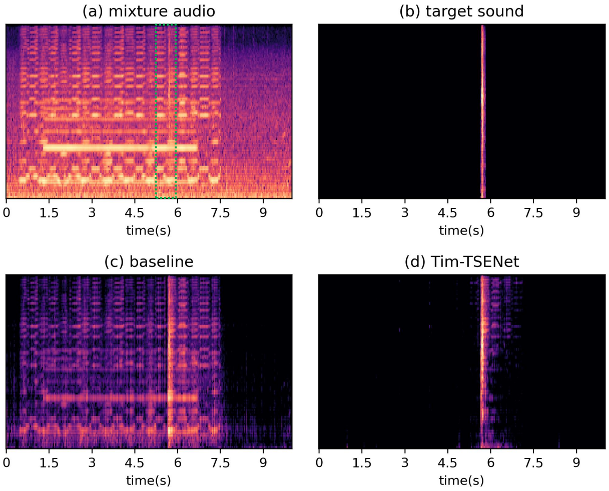

Figure 3: *Visualization of the spectrogram of a test example. (a) The input mixture audio. (b) The target sound. (c) The result of our baseline model. (d) The result of Tim-TSENet.*

4. Conclusions

This paper presented a novel network for target sound extraction task, which utilize the timestamp information from a detection network and apply a target-weighted time-frequency loss function to focus more on the target sound. We have shown the possibility of mutual learning for an extraction task and a detection task. A future work is to further improve the SI-SDRi-t, and we think designing new sound extractors is a potential solution.

5. Acknowledgements

This paper was partially supported by Shenzhen Science & Technology Research Program (No:GXWD20201231165807007-20200814115301001; No: JSGG20191129105421211) and NSFC (No: 62176008).

[1] http://dongchaoyang.top/Tim-TSENet-demo/

6. References

[1] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černockỳ, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[2] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," *Proc. Interspeech*, pp. 2728–2732, 2019.

[3] Q. Lin, L. Yang, X. Wang, L. Xie, C. Jia, and J. Wang, "Sparsely overlapped speech training in the time domain: Joint learning of target speech separation and personal VAD benefits," *arXiv preprint arXiv:2106.14371*, 2021.

[4] O. Slizovskaia, G. Haro, and E. Gómez, "Conditioned source separation for musical instrument performances," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2083–2095, 2021.

[5] J. H. Lee, H. Choi, and K. Lee, "Audio query-based music source separation," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 878–885.

[6] M. Borsdorf, C. Xu, H. Li, and T. Schultz, "Universal speaker extraction in the presence and absence of target speakers for speech of one and two talkers," in *Proc. Interspeech*, 2021.

[7] M. Delcroix, J. B. Vázquez, T. Ochiai, K. Kinoshita, and S. Araki, "Few-shot learning of new sound classes for target sound extraction," *arXiv preprint arXiv:2106.07144*, 2021.

[8] B. Gfeller, D. Roblek, and M. Tagliasacchi, "One-shot conditional audio filtering of arbitrary sounds," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 501–505.

[9] F. Pishdadian, G. Wichern, and J. Le Roux, "Learning to separate sounds from weakly labeled scenes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 91–95.

[10] Q. Kong, Y. Wang, X. Song, Y. Cao, W. Wang, and M. D. Plumbley, "Source separation with weakly labelled data: An approach to computational auditory scene analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 101–105.

[11] T. Ochiai, M. Delcroix, Y. Koizumi, H. Ito, K. Kinoshita, and S. Araki, "Listen to what you want: Neural network-based universal sound selector," *Proc. Interspeech*, pp. 1441–1445, 2020.

[12] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[13] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 6440–6444.

[14] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[15] X. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, 2012.

[16] D. Yang, H. Wang, Y. Zou, and C. Weng, "Detect what you want: Target sound detection," *arXiv preprint arXiv:2112.10153*, 2021.

[17] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Proceedings of the 18th ISMIR Conference*, 2017.

[18] E. Fonseca, M. Plakal, F. Font, D. P. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Scenes and Events 2018 Workshop (DCASE2018)*, p. 69.

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

[20] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 175–179.

[21] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.

[22] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," *Interspeech*, pp. 545–549, 2016.

[23] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2013.

[24] J. Benesty, S. Makino, and J. Chen, *Speech enhancement*. Springer Science & Business Media, 2006.

[25] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 696–700.

[26] L. Yi and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[27] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 246–250.

[28] H. Wang, Y. Zou, D. Chong, and W. Wang, "Environmental sound classification with parallel temporal-spectral attention," *Proc. Interspeech*, pp. 821–825, 2020.

[29] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," *arXiv preprint arXiv:1807.09840*, 2018.

[30] H. Wang, D. Chong, D. Huang, and Y. Zou, "What affects the performance of convolutional neural networks for audio event classification," in *International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, 2019, pp. 140–146.

[31] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[32] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[33] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.