Interspeech 2024
1-5 September 2024, Kos, Greece

WeSep: A Scalable and Flexible Toolkit Towards Generalizable Target Speaker Extraction

Shuai Wang[1,2,5], *Ke Zhang*[1], *Shaoxiong Lin*[3], *Junjie Li*[1], *Xuefei Wang*[1], *Meng Ge*[3]
Jianwei Yu[4], *Yanmin Qian*[3], *Haizhou Li*[2,1]

[1]Shenzhen Research Institute of Big Data, [2]School of Data Science,
The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), Guangdong, China
[3]Auditory Cognition and Computational Acoustics Lab, Shanghai Jiao Tong University, Shanghai, China
[4] Tencent AI Lab, Shenzhen,Guangdong, China, [5] WeNet Open Source Community

wangshuai@cuhk.edu.cn

Abstract

Target speaker extraction (TSE) focuses on isolating the speech of a specific target speaker from overlapped multi-talker speech, which is a typical setup in the cocktail party problem. In recent years, TSE draws increasing attention due to its potential for various applications such as user-customized interfaces and hearing aids, or as a crutial front-end processing technologies for subsequential tasks such as speech recognition and speaker recongtion. However, there are currently few open-source toolkits or available pre-trained models for off-the-shelf usage. In this work, we introduce WeSep, a toolkit designed for research and practical applications in TSE. WeSep is featured with flexible target speaker modeling, scalable data management, effective on-the-fly data simulation, structured recipes and deployment support. The toolkit will be publicly avaliable at `https://github.com/wenet-e2e/WeSep`.

Index Terms: target speaker extraction, speaker embedding, cocktail-party problem

1. Introduction

Daily communication environments are often complex, with various audio sources and voices intertwining. Interestingly, humans seem to possess a natural ability: in such complicated backgound, they can effectively focus their attention on the voice of the person they want to listen to. This phonomenon is often termed as "Selective Attentive Mechnism" [1, 2, 3]. Target speaker extraction (TSE) aims to enable a similar process. Unlike blind source separation (BSS), TSE typically relies on additional cue information that directly indicates the identity of the target speaker, thereby circumventing the permutation problem, leading to more flexible and applicable systems. In the current era of large-scale models, it is critical to take advantage of the abundant online media resources. However, before utilizing them for tasks like speech synthesis, it is necessary to process and filter these resources. TSE can play an important role in such pipelines [4].

TSE has gained significant attention in academia and industry. However, the availability of related open-source tools is relatively limited. This scarcity can be attributed to two main factors. Firstly, most TSE research is conducted on synthetic datasets, which may not generalize well to real speech. Secondly, improving the generalization performance for unknown speakers requires advanced speaker modeling techniques. To address these limitations, we aim to provide an accessable open-source toolkit called "WeSep", focusing on TSE.

The key features of the WeSep toolkit are as follows,

- To the best of our knowledge, WeSep is the first toolkit focusing on target speaker extraction task, implementing current mainstream models with plans to incorporate more powerful models in the future.
- WeSep has achieved seamless integration with the open-source speaker modeling toolkit Wespeaker [5], allowing for flexible integration with powerful pre-trained models and predefined network architectures for joint training.
- Following the design of WeNet and WeSpeaker, WeSep offers a flexible and efficient data management mechenism called Unified IO (UIO). This mechanism enables WeSep to easily handle large-scale datasets, ensuring scalability and efficiency in data processing.
- WeSep implements the on-the-fly data simulation pipeline, which allows users to leverage mono-speaker audios prepared for speech recognition or speaker recognition without the need for pre-mixing, thereby enabling model training to scale up and achieve better performance with large datasets.
- Lastly, models in WeSep can be easily exported by torch Just In Time (JIT) or as the ONNX format, which can be easily adopted in the deployment environment. Pretrained models and sample deployment codes in C++ are also provided.

2. Related Work

2.1. Target Speaker Extraction

A typical TSE system is depicted in Figure 1. Assume the mixture signal m containing K speakers is composed of the target speaker x_s and other $K-1$ interfere speakers, as demonstrated by

$$m = x_s + \sum_{k \neq s}^{K} x_k + \epsilon \tag{1}$$

where ϵ represents the residual signals capturing noise and reverberation.

A TSE system aims to reconstruct the x_s from the mixture waveform m, given the cue C_s. The optimization goal of TSE model $\mathcal{M}_{\text{TSE}}$ parameterized by θ^{TSE} is to minimize the training loss $\mathcal{L}(\cdot)$, which measures how close estimated target speech $\hat{x}_s$ is to the target source signal x_s.

$$\theta^{\text{TSE}} = \arg\min_{\theta} \mathcal{L}\left(x_s, \hat{x}_s\right) \tag{2}$$

$$\hat{x}_s = \mathcal{M}_{\text{TSE}}\left(m, C_s; \theta\right) \tag{3}$$

For audio-based target speaker extraction, the cue C_s typically refers to a pre-enrolled utterance from the target speaker. In the case of visual-based TSE[1], C_s can be represented by a sequence of image frames capturing the lip movements of the target speaker.

[1] The visual cue based TSE will be supported in next release

10.21437/Interspeech.2024-1840

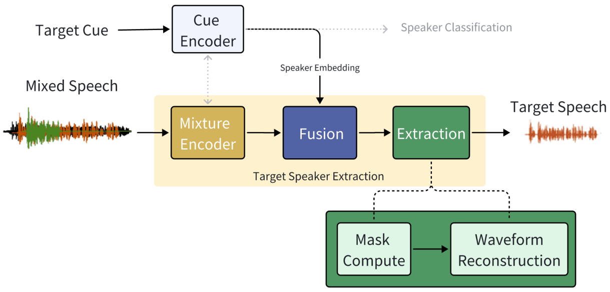

Figure 1: *Architecture of a typical TSE system, the cue encoder can be jointly trained or pretrained, an additional speaker classification loss is usually added in the joint-training mode. The parameters of the cue encoder can be shared (or partially shared) with the mixture encoder.*

2.2. Related Open-Source Projects

Deep learning-based TSE systems have gained popularity in recent years. Although some notable works, such as Spex+[2] [6] and SpeakerBeam[3] [7], have made their source code publicly available, there is currently no comprehensive toolkit specifically dedicated to this task. Unlike some general-purpose tools like [8] provide simple TSE recipes, WeSep features a simple code structure that focuses on TSE. In addition to defining speaker model structures within WeSep, users can directly access various state-of-the-art models and pre-trained models from WeSpeaker.

3. WeSep

3.1. Unified I/O for Local Data Management

To effectively handle both experimental data and production-scale datasets that encompass tens of thousands of hours of speech, often fragmented into a multitude of small files, we have implemented the Unified Input/Output (UIO) framework [9] within WeSep. This mechanism has also been integrated into WeNet and WeSpeaker.

3.2. On-the-Fly Data Simulation

For datasets like Libri2Mix [10], researchers typically use preprocessed data and standardized setups to ensure fair comparisons. However, to develop functional systems for real-world applications, it is necessary to train on a substantial amount of data. Preprocessing data and storing it on a hard drive is not an optimal solution. Instead, we propose employing an online data simulation approach as shown in Figure 2. This method not only conserves storage resources but also allows for the creation of a more diverse set of training data in a flexible manner, thereby enhancing the robustness of the model.

3.2.1. Online Noise and Reverb Generation

WeSep supports online noise addition and reverberation generation. In line with the approach implemented in WeSpeaker [5], we draw additive noises from a designated noise database, such as MUSAN [11] and AudioSet [12]. However, when it comes to reverberation, WeSep not only offers standard sampling from a Room Impulse Response (RIR) dataset [13] but also incorporates the fast random approximation of RIR signals, as introduced in the work by Luo et al. [14]. This enhancement allows for more dynamic and customizable reverberation effects tailored to various acoustic environments.

3.2.2. Dynamic Speaker Mixing Strategy

Dynamic Speaker Mixing (DSM) [15]involves generating the mixture waveform in real-time during the training process. In contrast to traditional static mixing methods, DSM enhances the model's robustness and generalization ability by introducing greater data diversity and complexity. In WeSep, the DSM algorithm implemented follows Algorithm 1.

Algorithm 1: Dynamic Speaker Mixing Strategy

```
Data:
$n_{\text{speaker}}$: Number of speakers for the mixed speech
$L_{\text{Buffer}}$: Buffer list containing training utterances
$L_{\text{wavs}}$: List of wavs to mix
$\text{SNR}_{\text{min}}$: Min value of SNR (interfere v.s. target speaker)
$\text{SNR}_{\text{max}}$: Max value of SNR (interfere v.s. target speaker)
$L_{\text{wavs}} \leftarrow [\,]$;
for $i \leftarrow 0$ to $n_{speaker}$ do
    if $i == 0$ then
        // Select the utterance for target
           speaker
        $s_t \leftarrow random\_sample(L_{\text{Buffer}})$;
        $L_{\text{wavs}}.\text{append}(s_t)$;
    else
        // Select the utterance for
           interfere speaker and scale with
           random snr
        $snr \leftarrow random.uniform(\text{SNR}_{\text{min}}, \text{SNR}_{\text{max}})$;
        $s_i \leftarrow random\_sample(L_{\text{Buffer}})$;
        while $same\_speaker(s_t, s_i)$ do
            $s_i \leftarrow random\_sample(L_{\text{Buffer}})$;
        end
        $L_{\text{wavs}}.append(rescale(s_i, snr))$;
    end
    $s_m = add\_and\_rescale(L_{\text{wavs}})$
end
Output: $s_m$, $L_{\text{wavs}}$
```

3.3. Backbone Support

- **ConvTasNet**: Proposed in [16], ConvTasNet is a pioneering deep learning model for single-channel audio source separation that operates directly in the time domain, utilizing convolutional neural networks to learn and estimate masks for separating target sources from mixtures. Based on Conv-TasNet, WeSep supports its most famous variant talored for the TSE task, Spex+ [6].
- **BSRNN**: Initially proposed in [17] for music source separation, Band-split Recurrent Neural Network (BSRNN) explicitly divides the spectrogram into different frequency bands and performs fine-grained modelling. [18] adapts BSRNN for the task of personal speech enhancement (PSE) by incorporating an additional speaker embedding, which inspires the implementation of the BSRNN for TSE in WeSep.
- **DPCCN**: The Densely-connected Pyramid Complex Convolutional Network (DPCCN) [19] is a novel architecture inspired by DenseUNet, incorporating features from Temporal Convolutional Networks (TCNs) and DenseNet to improve separation performance.
- **TF-GridNet**: Proposed in [20], TF-GridNet operates in the T-F domain and stacks several multi-path blocks to leverage

[2] https://github.com/gemengtju/SpEx_Plus

[3] https://github.com/BUTSpeechFIT/speakerbeam

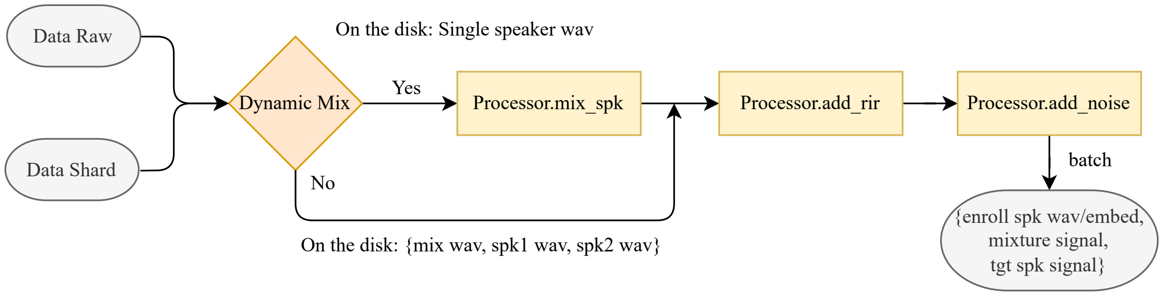

Figure 2: *The online data preparation pipeline in WeSep, the case of 2 speakers is demonstrated*

local and global spectro-temporal information, representing the State-of-the-art model for speech seperation. In WeSep, speaker embeddings are integrated prior to each multi-path block to specifically tailor it for the TSE task.

3.4. Target Speaker Modeling

To guide the extraction of target speaker's speech, a cue C_s is provided, for the audio based TSE, the cue C_s^a is often represented by a fixed-dimensional speaker embedding, extracted from a speaker encoder which is pretrained for the speaker recognition task or jointly trained within the TSE model.

3.4.1. Speaker Encoders

Besides specific design in well-known architectures, such as the ResNet based speaker encoder in Spex+ [6], WeSep offers seamless integration with various speaker models that are predefined in WeSpeaker [5]. It provides support for both "pretrained" and "joint training" modes. The "pretrained" mode involves loading the weights released by WeSpeaker[4], while the "joint training" mode only requires the model definition to be loaded, with the weights being optimized jointly with the targeted speech enhancement (TSE) task.

```
# psudo-codes for integrating wespeaker models
from wespeaker import get_speaker_model
s = get_speaker_model(spk_model_name)(**spk_args)
m = BSRNN(**sep_args) # Or other backbones
m.speaker_model = s
```

3.4.2. Fusion methods

Considering a speaker embedding $\mathbf{e}_s$ derived from the cue C_s and the intermediate outputs $\mathbf{H} = \mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T$ encoded from the mixed signal m, WeSep supports the following fusion methods, both for the pretraining mode and joint training mode.

- **Concat**: Directly replicate $\mathbf{e}_s$ for T times and concatenate it to $\mathbf{H}$, as used in VoiceFilter [21] and Spex series [22, 6].
- **Add**: $\mathbf{e}_s$ is first projected to the same dimension with $\mathbf{h}_t$ and do sample-wise addition.
- **Multiply**: $\mathbf{e}_s$ is first projected to the same dimension with $\mathbf{h}_t$ and do sample-wise multiplication. This is adopted mainly in the SpeakerBeam series [23, 24].
- **FiLM**: Feature-wise linear modulation (FiLM) [25, 26] applies a transformation to $\mathbf{H}$ by a learned affine transformation, represented by $\mathbf{h}'_t = \gamma(\mathbf{e}_s) \odot \mathbf{h}_t + \beta(\mathbf{e}_s)$, where γ and β are functions of the speaker embedding $\mathbf{e}_s$, and $\odot$ denotes element-wise multiplication.

3.5. Training Strategies

3.5.1. Joint Training with Speaker Encoders

Despite directly leveraging the pretrained speaker encoders for target cue extraction, WeSep also facilitates the joint optimization of the speaker encoder along with other components. An optional speaker classification loss can be easily configured to help contrain the learned speake embedding space.

3.5.2. Online Sampling of the Enrollment

To improve the model's resilience to varying enrollment conditions, WeSep maintains a correspondence mapping of spk2utt for all training data. This allows for the random selection of an enrollment utterance belonging to the target speaker for each sample, with the optional corruption of noise addition or reverberation effects to simulate more challenging conditions.

3.5.3. Training Objectives

WeSep follows common TSE research by using negative scale-invariant signal-to-noise ratio (SI-SNR) [27] as the default training objective. For flexibility, we integrated loss functions from Auraloss[5]. Additionally, we implemented GAN-based loss to offer potential enhancement in perceptual quality.

3.6. Deployment

Models in WeSep can be effortlessly exported to ONNX or PyTorch's Just-In-Time (JIT) format. We provide sample code to facilitate deployment. Additionally, we offer command-line interfaces (CLI) that are accessible through a straightforward "*pip install*" process. Users have access to off-the-shelf pretrained models which can be easily used as a standalone tool or for integration into custom pipelines.

4. Recipes and Results

WeSep provides recipes for the standard datasets such as Libri2Mix [10][6], following their respective split and pre-mixing strategies. Additionally, WeSep utilizes the VoxCeleb dataset to showcase the construction of a more generalizable TSE system using single-speaker data collected from real-world scenarios. However, due to space limitations, we will focus on a detailed comparison using the Libri2Mix dataset and highlight the generalization capabilities using VoxCeleb. For comprehensive results on other datasets, please refer to the online repository.

[4] https://github.com/wenet-e2e/wespeaker/blob/master/docs/pretrained.md

[5] https://github.com/csteinmetz1/auraloss

[6] Recipes for standard WSJ0-2Mix [28] and AISHELL-2Mix [19] datasets are also provided, but not presented in this paper

4.1. Libri2Mix

The performance of different models on the Libri2Mix-Eval dataset are showcased in Table 1. In line with the approach detailed in [19, 6], we have implemented DPCCN and Spex+ with a default joint training of the speaker encoder. For the remaining models, the ECAPA-TDNN [29] pretrained on the VoxCeleb2 [30] Dev set by WeSpeaker is utilized.

Table 1: *SI-SDR (dB) comparison of different models*

Backbone	SpkEnc	Training Data	
		train-100	train-360
BSRNN	Pretrain	13.32	16.57
TF-GridNet	Pretrain	12.09	15.79
DPCCN	Joint Train	11.45	13.80
Spex+	Joint Train	12.64	14.57

In the sections below, we will provide a detailed analysis of the impact of fusion strategy, speaker model architecture, and the pretrain/joint-train paradigm. Unless otherwise specified, the experiments utilize BSRNN as the default backbone, the pretrained ECAPA-TDNN as the speaker model, multiplication as the default method, and train-100 as the training dataset.

4.1.1. Impact of the Fusion Strategy

To incorporate the encoded speaker representation into the TSE system, a fusion mechanism is employed. Four fusion methods are compared in Table 2, and it is observed that the simple multiplication achieves the best performance, followed by FiLM. Concatenation and addition methods show similar results.

Table 2: *Performance comparison of different fusion methods*

Fusion Method	Concat	Add	Multiply	FiLM
Libri2Mix	12.84	13.15	13.25	**13.32**
AISHELL2Mix	4.61	5.15	4.76	**5.54**

4.1.2. Impact of the Speaker Model

To assess the compatibility of various pretrained speaker encoders, we present the results of the BSRNN system utilizing different pretrained embeddings in Table 3. When comparing architectures trained on the same dataset (VoxCeleb2-Dev, 5994 speakers), achieving superior results on the speaker verification task (VoxCeleb1-O) does not necessarily lead to enhanced performance on the TSE task. However, training on a more extensive dataset can lead to improved TSE performance. For instance, the CAM++ model [31] developed by Alibaba[7], trained on a dataset of 200,000 Chinese speakers, demonstrates this improvement, despite its poor performance on VoxCeleb1-O, which may be due to the language mismatch.

Table 3: *Performance comparison using different pretrained speaker encoders*

SpkEnc Type	Train Data	SI-SDR (dB)	EER (%)
TDNN [32]	VoxCeleb2	12.41	1.721
ResNet34 [33]	VoxCeleb2	13.18	0.937
ECAPA-TDNN [29]	VoxCeleb2	13.32	1.072
CAM++ [31]	VoxCeleb2	12.29	0.845
CAM++	Ali 200k	14.50	6.225

4.1.3. Impact of Joint Training

WeSep facilitates the joint training of the speaker encoder alongside the backbone model. In Table 4, we present some preliminary results of various training paradigms, illustrating that joint training typically yields superior performance[8]. However, we did not observe the anticipated additional performance gain from the inclusion of the speaker classification loss, as suggested in [6, 24].

Table 4: *Joint training v.s. pretrained speaker encoder*

SpkEnc Type	Joint Training	Multitask	SI-SDR (dB)
ResNet34	×	×	13.18
	✓	×	13.96
	✓	✓	13.97
ECAPA-TDNN	×	×	13.32
	✓	×	13.87
	✓	✓	13.85

4.2. VoxCeleb1

To privide a TSE model with enhanced applicability and to exemplify the training process of such a system utilizing large-scale data, we have offered a Recipe on VoxCeleb1 [34].

Table 5: *Generalization on out-of-domain dataset*

Training Dataset	SI-SDR (dB)	
	Libri2Mix	AISHELL2Mix
Libri2Mix-train-100	13.32	5.54
Libri2Mix-train-360	16.57	8.17
VoxCeleb1	16.18	10.18

As demonstrated in Table 5, the system trained on VoxCeleb1 yields results on Libri2Mix that are comparable to those obtained by the system trained on in-domain data. Moreover, it exhibits significantly better generalization capabilities on AISHELL2Mix.

5. Conclusion and Future work

In this paper, we present WeSep, an open-source project focused on Target Speaker Extraction. WeSep is designed with versatile speaker modeling capabilities, enables online data simulation, and offers scalability to large-scale datasets. Looking ahead, WeSep will continually integrate state-of-the-art (SOTA) models, audio-visual recipes, and will expand its capabilities to include blind speech separation tasks within a unified framework.

6. Acknowledgement

This work is supported by Internal Project of Shenzhen Research Institute of Big Data under grant No. T00120220002 and No.J00220230014. Thanks to the help from the WeNet Open Source Community, especially Chengdong Liang.

[7] `https://modelscope.cn/models/iic/speech_campplus_sv_zh-cn_16k-common`

[8] This doesn't always holds, for instance, joint trained system with CAM++ can not beat the pretrained model with Ali 200k data, further investgation needs to be carried out

7. References

[1] K. Zmolikova, M. Delcroix, T. Ochiai, K. Kinoshita, J. Černockỳ, and D. Yu, "Neural target speech extraction: An overview," *IEEE Signal Processing Magazine*, vol. 40, no. 3, pp. 8–29, 2023.

[2] A. W. Bronkhorst, "The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions," *Acta acustica united with acustica*, vol. 86, no. 1, pp. 117–128, 2000.

[3] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[4] J. Yu, H. Chen, Y. Bian, X. Li, Y. Luo, J. Tian, M. Liu, J. Jiang, and S. Wang, "Autoprep: An automatic preprocessing framework for in-the-wild speech data," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1–5.

[5] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[6] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "SpEx+: A Complete Time Domain Speaker Extraction Network," in *Proc. Interspeech 2020*, 2020, pp. 1406–1410.

[7] 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.

[8] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.

[9] B. Zhang, D. Wu, Z. Peng, X. Song, Z. Yao, H. Lv, L. Xie, C. Yang, F. Pan, and J. Niu, "WeNet 2.0: More Productive End-to-End Speech Recognition Toolkit," in *Proc. Interspeech 2022*, 2022, pp. 1661–1665.

[10] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Librimix: An open-source dataset for generalizable speech separation," *arXiv preprint arXiv:2005.11262*, 2020.

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

[12] 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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[13] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas *et al.*, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[14] Y. Luo and J. Yu, "FRA-RIR: Fast random approximation of the image-source method," in *Interspeech*, 2023, pp. 3884–3888.

[15] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.

[16] Y. Luo 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.

[17] Y. Luo and J. Yu, "Music source separation with band-split rnn," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[18] J. Yu, H. Chen, Y. Luo, R. Gu, W. Li, and C. Weng, "Tspeech-ai system description to the 5th deep noise suppression (dns) challenge," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–2.

[19] J. Han, Y. Long, L. Burget, and J. Černockỳ, "Dpccn: Densely-connected pyramid complex convolutional network for robust speech separation and extraction," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7292–7296.

[20] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "Tf-gridnet: Making time-frequency domain models great again for monaural speaker separation," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[21] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," in *Interspeech*, 2018.

[22] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 28, pp. 1370–1384, 2020.

[23] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018, pp. 5554–5558.

[24] M. Delcroix, T. Ochiai, K. Zmolikova, K. Kinoshita, N. Tawara, T. Nakatani, and S. Araki, "Improving speaker discrimination of target speech extraction with time-domain speakerbeam," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 691–695.

[25] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[26] S. Cornell, Z.-Q. Wang, Y. Masuyama, S. Watanabe, M. Pariente, and N. Ono, "Multi-channel target speaker extraction with refinement: The wavlab submission to the second clarity enhancement challenge," *arXiv preprint arXiv:2302.07928*, 2023.

[27] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[28] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2016, pp. 31–35.

[29] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[30] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[31] H. Wang, S. Zheng, Y. Chen, L. Cheng, and Q. Chen, "CAM++: A Fast and Efficient Network for Speaker Verification Using Context-Aware Masking," in *Proc. INTERSPEECH 2023*, 2023, pp. 5301–5305.

[32] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[33] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[34] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *Telephony*, vol. 3, pp. 33–039, 2017.