

Lightweight Dual-channel Target Speaker Separation for Mobile Voice Communication

Yuanyuan Bao^{*,1}, Yanze Xu^{*,1}, Na Xu[†], Wenjing Yang[†], Hongfeng Li[†], Shicong Li[†], Yongtao Jia[†],
Fei Xiang[†], Jincheng He^{*} and Ming Li^{*,2},

^{*} Data Science Research Center, Duke Kunshan University, Kunshan, China

[†] Xiaomi, Beijing, China

Abstract—Nowadays, there is a strong need to deploy the target speaker separation (TSS) model on mobile devices. While the existing open-source datasets, single-channel or dual-channel simulated by simple addition from single ones, are not suitable for the mobile communication scenario, and the available computational resource is constrained strictly in mobile devices. To better perform TSS for mobile voice communication, we first make a dual-channel dataset based on a specific scenario with two people chatting on the phone, LibriPhone. Specifically, LibriPhone is made by simultaneously replaying pairs of utterances from LibriSpeech by two professional artificial heads and recording by two built-in microphones of the mobile. Then, to fit the restriction of computational resource on mobile devices, we propose a lightweight time-frequency domain separation model, LSTMFormer, which is based on the LSTM module with source-to-noise ratio (SI-SNR) loss. For the experiments on the re-recorded dataset, we explore the dual-channel LSTMFormer model and a single-channel version by a random single channel of LibriPhone. Experimental result shows that both versions work well on LibriPhone, and the dual-channel LSTMFormer outperforms the single-channel LSTMFormer with relative 25% improvement. This work provides a feasible solution for the TSS task on the mobile devices, playing back and recording multiple data sources in real application scenarios for getting dual-channel real data can assist the lightweight model to achieve higher performance.

Index Terms—Target speaker separation, dual-channel, lightweight, LibriPhone

I. INTRODUCTION

In mobile voice communication scenario, the speech quality is always degraded because of the existence of interference signals, such as background noise and people talking around. There are a lot of studies on integrating the speech enhancement algorithms into mobile devices [1]–[3] to suppress the regular noise, while the voice of unwanted people could not always be suppressed that well, especially when the volume level of interpreted voice is close to the clean speech. That calls for a technique to separate the human utterance properly.

Speech separation is such a technique that decomposes the given mixed utterance into individual sources. Thanks to the development of deep learning, these years have witnessed great progress in this area, such as deep attractor network (DANet) [4], [5], permutation invariant training (PIT) [6], [7], time-domain audio source separation (Tas-Net) [8]–[11]. While these methods all require the number of speakers to be known

in advance, which limits the usage of speech separation in real-world applications. In recent years, a new technique called target speaker separation (TSS) [12], [13] or speaker extraction [14]–[17] is brought up. It could separate the target speaker’s voice from the mixed utterance when the reference of the target speaker is available. TSS avoids the permutation and output dimension problems in traditional speech separation method simultaneously, making it more suitable in the mobile voice communication scenario.

Studies on TSS technique have achieved impressive and heuristic performance. Single-channel work like the time-frequency domain model VoiceFilter [13] and time-domain model SpEx+ [16] use the speaker embedding extracted from the reference utterance to help the model learn whom to separate. Multi-channel work like [18] makes use of the location information of the target speaker to get the utterance wanted. In the multi-modal domain, researchers incorporate visual information into the speech separation system for better separation performance [19].

The mentioned studies do achieve state-of-the-art results on public speech separation datasets. However, there are still some limitations when it comes to real-world applications. First, due to the restriction of computational resources on mobile phones, the model size, and computing complexity are often constrained strictly. And the real constriction is on the separation module rather than the speaker verification module. Specifically, the target speaker’s utterance is usually enrolled in advance, thus the speaker embedding could be extracted and saved beforehand. So only the parameters and multiply-accumulate operations (MACs) cost of the separation module are considered and reported. However, to improve the separation accuracy, speech separation modules always have a large number of parameters and deep network design. For instance, the MACs costs of the well-known separation models, TasNet [8]–[11] and even its light version [22], are of the order of G. The Atss-Net [20] do decrease the number of parameters significantly compared to VoiceFilter, while the computational complexity is even higher. Another issue is the need for causal and real time processing which many attention based models can not satisfy. The third difficulty for the application on mobile is that there is a domain mismatch between the real scenario of mobile voice communication and that of the simulated available public corpus. Moreover, the dual-channel data used for dual-channel separation is

¹Equal Contribution.

²Corresponding author: Ming Li, ming.li369@dukekunshan.edu.cn

always simulated from the single-channel public corpus [21] by simple addition. The domain gaps caused by both the simulation and the difference of the scenarios may lead to the degradation of the real-case inference performance.

To overcome the aforementioned problems for the application on mobile devices, we first record a dual-channel dataset called LibriPhone to mimic the mobile voice communication scenario. Specifically, instead of simulating mixed data by direct addition, LibriPhone is collected in a relatively real setting in which two random utterances from LibriSpeech are simultaneously played by two professional artificial heads and recorded by two built-in microphones in a cell phone.

Based on our previous work in Atss-Net [20], we design a lightweight time-frequency domain separation model and share the same speaker verification module with Atss-Net. The reason for choosing the time-frequency domain model is that it has a smaller input dimension than the time-domain model, which means smaller model size. Inspired by the effort made in lightweight model development in speech separation domain [22], [23] and the introduction of the SI-SNR loss [8], we design a lightweight target speaker separation model, LSTMFormer, based on the LSTM network and incorporate SI-SNR loss for better performance. Nonetheless, re-recording will inevitably introduce delays between the original clean utterance sent to artificial mouth and the one received by the microphones caused by the recording equipment. This may cause the non-convergence problem for training. The reason is that the computation of the SI-SNR loss is not robust to the time delay. Therefore, we perform generalized cross-correlation [24] based alignment as a training strategy to address this problem.

For the experiments on LibriPhone, we explore both the dual-channel and single-channel versions of LSTMFormer. Specifically, both single- and dual-channel versions work well on LibriPhone, and the MACs costs of both versions are less than 200M. Moreover, we discover that the dual-channel model outperforms the single-channel one significantly when they have similar model size. It shows that the dual-channel data is easier for the lightweight model to learn and to achieve better performance, which provides a feasible solution when the available computational resource is constrained on some mobile devices.

II. DATASET DESCRIPTION

The LibriPhone is a dual-channel dataset re-recorded from LibriSpeech for mobile voice communication scenario. There are 25 hours 16 kHz sampled mixed utterances in total and split with 15 hours for training, 5 hours for validating, and 5 hours for testing. The subsets for training and validating is made from train-clean-100, the test subset is made from dev-clean. In each subset, we respectively playback and record the mixed utterance and target utterance. Specifically, for each mixed utterance, we select a pair of utterances, the target speaker utterance, and the other speaker utterance, with non-repetition of the target speaker identity. And then, two artificial heads separately replay two clean utterances simultaneously. A

cell phone with two microphones is used to record the mixed sound. Likewise, we also record the target speaker utterances as the ground truth for supervised learning with the similar setup but only one artificial head replaying the target speaker utterance.

The detailed setup is shown in Figure 1. The recording equipment is an Android phone placed in the center of the room. The room is 3.3 meters wide and 3.5 meters long with a height of 2.3 meters. Two artificial heads are used to mimic people talking. The one closer to the phone is the target speaker, the other is the interference. The specific locations of both artificial heads are illustrated in Figure 1. The mouths of both artificial heads are placed at 1.5m height, the screen of the phone is horizontal and orients towards the target speaker. By measuring the acoustic parameters of the recording room, the background noise level is 25 dBA, the reverberation time T_{60} is 0.45s to 0.55s.

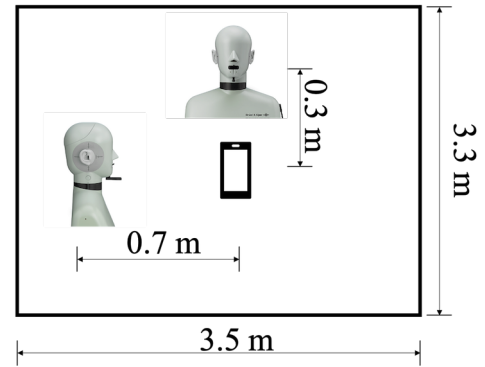


Fig. 1. The recording setup of our proposed LibriPhone database.

III. MODEL DESIGNING

Our TSS framework shares the same speaker verification module with our previous work in [20]. Due to the high computational complexity of the Transformer-encoder-based separation module, when designing LSTMFormer, we replace the self-attention in the Transformer encoder with a single direction LSTM and keep the idea of skip connection and layernorm.

In this work, we want to explore whether the real-collected dual-channel data in our simulated acoustic studio can achieve superior separation performance. Hence, we first designed a dual-channel version LSTMFormer on the LibriPhone. And then, for comparison, we also design a single-channel version LSTMFormer trained by using the random single channel of LibriPhone. The two versions of the LSTMFormer are respectively introduced in section III-A and section III-B.

A. Dual-channel LSTMFormer

As shown in Figure 2, an STFT Conv1D layer is used to transform the mixed dual-channel waveform $\mathbf{x} \in \mathbb{R}^{2 \times L}$ into magnitude and phase spectrogram, where L is the length of sample points, and \mathbf{x} is a linear combination of C sources $s_1(t), \dots, s_c(t)$. The mixed dual-channel utterance is first

split into two channels and passed through the Conv1D layer separately.

$$M_i, P_i = \text{Conv1D}(\text{channel}_i) \quad i = 1, 2 \quad (1)$$

where $M_i, P_i \in \mathbb{R}^{T \times F}$ stand for magnitude and phase spectrogram respectively. T and F are the dimension of frames and spectrogram bin axes.

As the yellow block in Figure 2 shows, the two magnitude spectrogram are firstly concatenated along the feature dimension and then compressed using fully-connected layer for feature extraction. The speaker embedding is repeated for

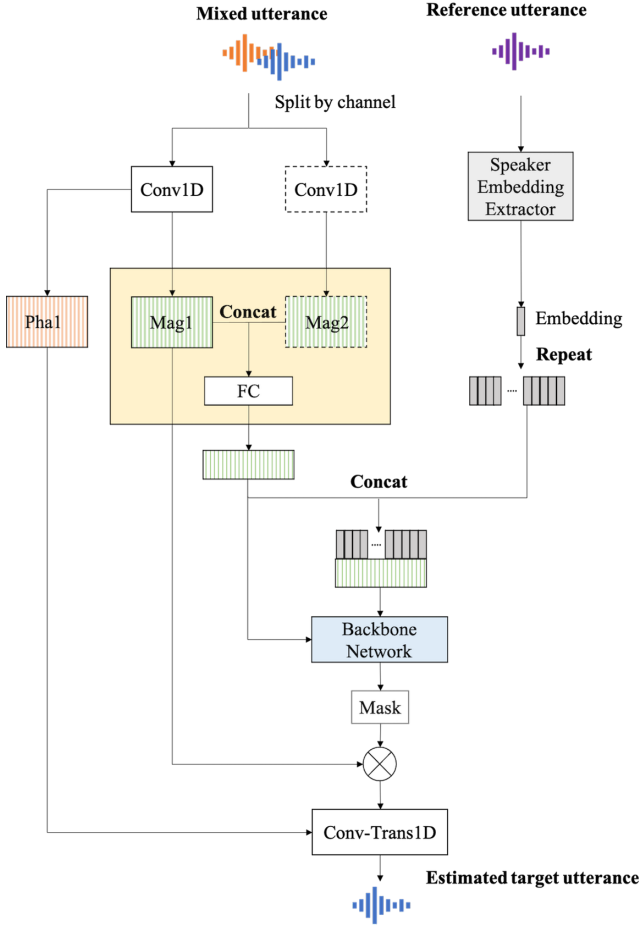


Fig. 2. Solid part + Dashed part: Model architecture of Dual-channel LSTMFormer. Solid part only: Model architecture of Single-channel LSTMFormer. Mag1 and Mag2 represent the magnitude spectrogram of the split two channels of mixed utterance. Pha1 is the phase spectrogram of the first channel used to reconstruct the target utterance. The box in yellow is used to compress Mag1 and Mag2.

T times for the ease of concatenation. The extracted feature map and the repeated speaker embedding are then concatenated along the feature dimension and fed into the following LSTMFormer. The backbone network of our proposed LSTMFormer model is shown in Figure 3. Motivated by the idea of residual learning [25], there is a skip connection between

the compressed magnitude spectrogram and the Layer Norm output. This design makes our network better remember the information of compressed magnitude spectrogram. The fully-connected layers (FC) are used for deeper feature extraction as well as the flexible dimension changes between different modules of the whole model. An extra FC layer in the end serves as the compress layer to reduce the dimension to match with the time-frequency mask for a single reference channel. Specifically, we choose the first channel as the reference channel. The estimated magnitude spectrogram is calculated by performing the element-wise product between the mixed magnitude spectrogram of the first channel M_1 and the estimated mask $R \in \mathbb{R}^{T \times F}$. With the estimated magnitude spectrogram and phase spectrogram of the first channel, the iSTFT Conv-Trans1D outputs the estimated target speaker's utterance $\hat{s} \in \mathbb{R}^{1 \times L}$.

$$\hat{s} = \text{Conv-Trans1D}(M_1 \odot R, P_1) \quad (2)$$

where \odot denotes the element-wise product operation.

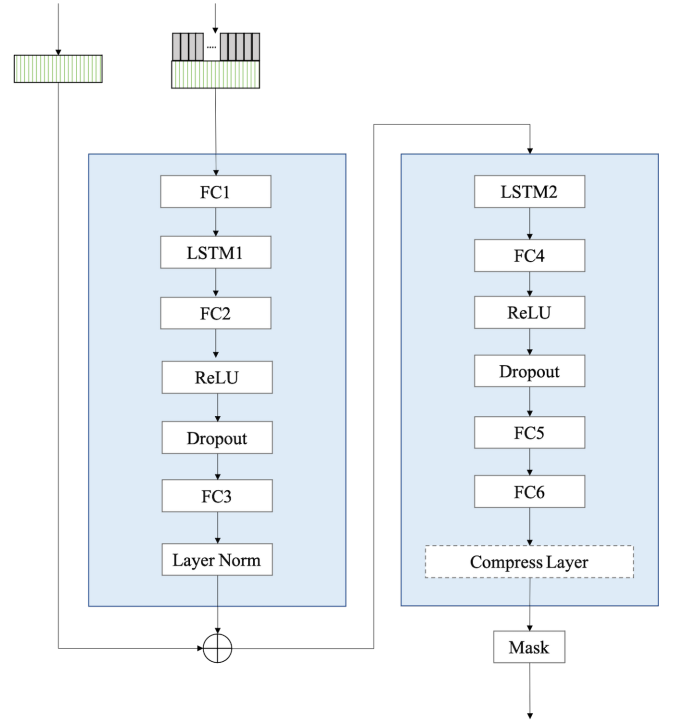


Fig. 3. The backbone network of the LSTMFormer model.

As the input and output of this whole separation module are both waveform, we choose scale-invariant source-to-noise ratio (SI-SNR) as our training target, to directly optimize the separation performance [8]. The SI-SNR is defined as follows:

$$s_{\text{target}} = \frac{\langle \hat{s}, s \rangle s}{\|s\|^2} \quad (3)$$

$$e_{\text{noise}} = \hat{s} - s \quad (4)$$

TABLE I

THE MODEL CONFIGURATIONS OF SINGLE- AND DUAL- CHANNEL LSTMFORMER MODELS. SINGLE-CHANNEL (HALF) AND SINGLE-CHANNEL (EQUAL) REFER TO HALF AND EQUAL MODEL PARAMETERS IN BACKBONE NETWORK OF THE DUAL-CHANNEL MODEL, RESPECTIVELY.

Layer	Dual-channel	Single-channel (half)	Single-channel (equal)
Conv1D	$[514 \times 400, 2]$	$[514 \times 400, 1]$	$[514 \times 400, 1]$
FC	$[1 \times 1, 256]$	$[1 \times 1, 128]$	$[1 \times 1, 256]$
FC1	$[1 \times 1, 256]$	$[1 \times 1, 128]$	$[1 \times 1, 256]$
LSTM1	$hidden_state_{num} = 256$	$hidden_state_{num} = 128$	$hidden_state_{num} = 256$
FC2	$[1 \times 1, 200]$	$[1 \times 1, 100]$	$[1 \times 1, 200]$
FC3	$[1 \times 1, 256]$	$[1 \times 1, 128]$	$[1 \times 1, 256]$
LSTM2	$hidden_state_{num} = 256$	$hidden_state_{num} = 128$	$hidden_state_{num} = 256$
FC4	$[1 \times 1, 180]$	$[1 \times 1, 90]$	$[1 \times 1, 180]$
FC5	$[1 \times 1, 256]$	$[1 \times 1, 128]$	$[1 \times 1, 256]$
FC6	$[1 \times 1, 514]$	$[1 \times 1, 257]$	$[1 \times 1, 514]$
Compress Layer	$[1 \times 1, 257]$	N/A	N/A
Conv-Trans1D	$[514 \times 400, 1]$	$[514 \times 400, 1]$	$[514 \times 400, 1]$

$$SI-SNR := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{noise}}\|^2} \quad (5)$$

where $s \in \mathbb{R}^{1 \times L}$ is the ground truth target speaker's utterance.

B. Single-channel LSTMFormer

For the single-channel version, as shown in Figure 2, removing modules with the dashed line, including one STFT Conv1D layer and one compress layer of the dual-channel version. There are two kinds of single-channel LSTMFormer with the same architectures but different model size. One of the model size of single-channel LSTMFormer is decreased to half by halving the hidden numbers of the LSTM and the output dimension of the FC layer. Another one's model size is close to the dual-channel LSTMFormer.

C. Model Training

All of our systems are implemented using PyTorch [26]. Adam serves as the optimizer during training [27] with an initial learning rate of 0.0001. All models are trained with the batch size of 64 and stopped till the model performance didn't improve on the validation set any longer. The model configurations of all the models related to LSTMFormer are shown in table I. The single-channel model shares the same architecture with the dual-channel one using half of the parameters in the dual-channel LSTMFormer module. To find out the impact of increasing model parameters and validate the advantage of dual-channel complementary information in the TSS task, we trained another single-channel model with as much parameters as the dual-channel model. During the training of the single-channel model, we randomly chose one channel of the mixed utterance. Signal-to-distortion ratio (SDR) is used as the evaluation metric for the TSS task.

D. Alignment

The unstable time delay between the target and mixed utterances in our LibriPhone dataset is found to be a big barrier

during the training stage. To overcome it, we compute the time delay of each target and mixed utterance pairs based on the generalized cross-correlation method implemented in Matlab¹ for better alignment.

IV. RESULTS AND DISCUSSION

For the performance of different models on the LibriPhone dataset, Table II shows the before-separated, after-separated SDR values and the relative improvement, the information on the model size and computational complexity are also reported.

From Table II, we can see that both dual- and single- LSTMFormer achieve an over 4.8dB SDR improvement, which means alignment can well address the problem that the SI-SNR loss can not converge because of the delay introduced by the re-recording. We can also find that the MACs cost² of any model with 1 second input speech is very low, which requires less computation resources of the application platforms.

TABLE II
PERFORMANCE OF DIFFERENT MODELS ON THE LIBRIPHONE TEST SET.
THE MODEL SIZE AND MACs COST ON 1-SECOND INPUTS OF DIFFERENT MODELS ARE ALSO REPORTED.

Model	Params	MACs	Mean SDR		
			Before	After	Improved
Single-channel (half)	0.37M	36.32M	3.12	7.97	4.85
Single-channel (equal)	1.31M	128.45M	3.22	8.34	5.12
Dual-channel	1.57M	154.29M	2.69	10.40	7.71

Specifically, the improvement of single-channel (half) with 36.32M MACs is 4.85dB, and the improvement of single-channel(equal) with 128.45M MACs is 5.12dB. The increase of the model size for single-channel LSTMFormer only improves 0.27dB SDR. However, the dual-channel LSTMFormer with 154.29M MACs achieves a 7.71dB improvement, a

¹<https://www.mathworks.com/help/phased/ref/gccphat.html>

²Calculated by open-source toolbox: <https://github.com/Lyken17/pytorch-OpCounter>

relative 25% improvement compared with the single-channel model. It shows that dual-channel data can help the lightweight model LSTMFormer to achieve better performance. This motivates us to explore whether the multi-channel data can improve the lightweight model further than the dual-channel data. In a similar way we performed, we can re-record multi-channel data to mimic more complex application scenarios. Without explicitly using the location and direction information, our work provides a feasible solution to the dual-channel TSS task.

V. CONCLUSIONS

In this paper, to reduce the domain gap of dual-channel TSS systems for mobile voice communication, we first design and collect a dual-channel dataset, LibriPhone. Specifically, to better mimic the real application scenario, instead of directly mixing the single-channel dataset, LibriPhone is made by simultaneously replaying pairs of utterances from LibriSpeech by two artificial heads and recording by two microphones of the mobile device. And then, we propose a lightweight time-frequency domain separation model, LSTMFormer for the experiments on LibriPhone. We explore both the dual-channel and the single-channel LSTMFormer model. Experimental result shows that the dual-channel model outperforms the single-channel significantly when they have the similar model size, and when the former is twice bigger than the latter, showing that the dual-channel data are complementary and easier for the lightweight model to learn. Our proposed lightweight model, and corresponding training strategies demonstrate the promising prospects of deep-learning-based TSS models being applied on real-world low-resource platforms.

REFERENCES

- [1] J.-M. Valin, "A hybrid dsp/deep learning approach to real-time full-band speech enhancement," in *MMSP*, pp. 1–5, IEEE, 2018.
- [2] J.-M. Valin, U. Isik, N. Phansalkar, R. Giri, K. Helwani, and A. Krishnaswamy, "A perceptually-motivated approach for low-complexity, real-time enhancement of fullband speech," *arXiv preprint arXiv:2008.04259*, 2020.
- [3] K. Tan, X. Zhang, and D. Wang, "Real-time speech enhancement for mobile communication based on dual-channel complex spectral mapping," in *ICASSP*, IEEE, 2021, pp. 6134–6138.
- [4] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *ICASSP*, IEEE, 2017, pp. 246–250.
- [5] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM TASLP*, vol. 26, no. 4, pp. 787–796, 2018.
- [6] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *ICASSP*, IEEE, 2017, pp. 241–245.
- [7] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM TASLP*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [8] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *ICASSP*, IEEE, 2018, pp. 696–700.
- [9] Y. Luo and N. Mesgarani, "Real-time single-channel dereverberation and separation with time-domain audio separation network," in *Interspeech*, IEEE, 2018, pp. 342–346.
- [10] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM TASLP*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [11] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP*, IEEE, 2020, pp. 46–50.
- [12] J. Du, Y. Tu, Y. Xu, L. Dai, and C. H. Lee, "Speech separation of a target speaker based on deep neural networks," in *ICSP*, IEEE, 2014, pp. 473–477.
- [13] 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," in *Interspeech*, IEEE, 2019, pp. 2728–2732.
- [14] C. Xu, W. Rao, E. S. Chng, and H. Li, "Time-domain speaker extraction network," in *ASRU*, IEEE, 2019, pp. 327–334.
- [15] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *IEEE/ACM TASLP*, vol. 28, pp. 1370–1384, 2020.
- [16] 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*, vol. 28, IEEE, 2020, pp. 1406–1410.
- [17] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "Multi-stage speaker extraction with utterance and frame-level reference signals," in *ICASSP*, IEEE, 2021, pp. 6109–6113.
- [18] G. Li, S. Liang, S. Nie, W. Liu, M. Yu, L. Chen, S. Peng, and C. Li, "Direction-aware speaker beam for multi-channel speaker extraction," in *Interspeech*, IEEE, 2019, pp. 2713–2717.
- [19] R. Gu, S. Zhang, Y. Xu, L. Chen, Y. Zou, and D. Yu, "Multi-modal multi-channel target speech separation," *IEEE JSTSP*, vol. 14, no. 3, pp. 530–541, 2020.
- [20] T. Li, Q. Lin, Y. Bao, and M. Li, "Atss-net: Target speaker separation via attention-based neural network," in *Proc. Interspeech*, 2020, pp. 1411–1415.
- [21] C. Li, J. Xu, N. Mesgarani, and B. Xu, "Speaker and direction inferred dual-channel speech separation," in *ICASSP*, IEEE, 2021, pp. 5779–5783.
- [22] Y. Luo, C. Han, and N. Mesgarani, "Ultra-lightweight speech separation via group communication," in *ICASSP*, IEEE, 2021, pp. 16–20.
- [23] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo rm-rf: Efficient networks for universal audio source separation," in *MLSP*, IEEE, 2020, pp. 1–6.
- [24] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE TASLP*, vol. 24, no. 4, pp. 320–327, 1976.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.
- [27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.