Lightweight Attentive ConvNeXt-TCN for Causal Target Sound Extraction

MinJie Xiang, Ruiyu Liang, *Member, IEEE*, Ye Ni, *Student Member, IEEE*, Li Zhao, and Björn W. Schuller, *Fellow, IEEE*

***Abstract*—Target sound extraction (TSE) aims to isolate specific sounds from complex acoustic mixtures. While various causal TSE models have been developed for real-time processing, most existing causal models operate in the time domain and do not effectively leverage frequency-domain information. This paper proposes a novel time-frequency domain model, ACN-TCN, which integrates the ConvNeXt design paradigm into the temporal convolutional network (TCN) to jointly model temporal and spectral information. Additionally, a convolutional self-attention mechanism is introduced to improve feature selection. Since shallow information is easily lost in a deep network, we incorporate a feature enhancement module to effectively integrate shallow features. Experimental results demonstrate that the proposed model improves signal-to-noise ratio (SNR) by 2.5 dB and scale-invariant signal-to-noise ratio (SI-SNR) by 3.4 dB compared to the state-of-the-art causal TSE method in single-target extraction task. Furthermore, ACN-TCN reduces the number of parameters by approximately 40% compared to previous models.**

***Index Terms*—Target sound extraction (TSE), deep learning, causal model, attention, efficient.**

I. Introduction

IN MODERN acoustic environments, various acoustic events (AEs) occur simultaneously, forming complex sound mixtures [1]. However, humans possess the remarkable ability to selectively perceive sounds of interest despite background noise and competing sources [2]. Inspired by this perceptual phenomenon, recent studies [3], [4], [5], [6], [7] have focused on target sound extraction (TSE), which aims to accurately isolate specific target sounds from complex audio mixtures containing interfering noise or competing sources. TSE is crucial for enhancing the robustness of human-computer interaction systems, restoring audio, and creating immersive audio experiences by filtering out irrelevant sounds.

The TSE system takes a mixture of audio signals and condition information about the target sound as input and outputs the extracted target sound [8]. Typically, the category labels of target sounds serve as conditioning information and can be represented as 1D vector. In real-world applications, real-time audio processing is often essential, which requires models to be causal. Waveformer [9] is the first causal real-time TSE model that employs dilated causal convolution (DCC) layers as the encoder and a transformer layer as the decoder. Another approach – CATSE [10] – uses a temporal convolutional network (TCN) [11], [12] as the separator and employs a convolutional layer for encoding and decoding time-domain audio. Although non-causal models often outperform causal models, they typically suffer significant performance degradation when transformed to their causal counterparts. To address this issue, a knowledge distillation method is proposed in [13], leveraging a non-causal model as a teacher to guide the learning of a causal model. Both the causal and non-causal models adopt the Conv-TasNet [14] architecture.

The three aforementioned causal models are time-domain models. Many studies [15], [16], [17], [18], [19], [20] on speech-related tasks, such as speech enhancement, speech separation, and neural vocoder, have demonstrated that incorporating information from both the time and frequency domains can significantly improve performance compared to pure time-domain models. Among them, TIGER [18] exemplifies how time–frequency domain modeling can lead to both computational efficiency and superior separation quality. In this letter, we propose Attentive ConvNeXt-TCN (ACN-TCN), a causal TSE method that operates in the time-frequency domain. ACN-TCN integrates the design paradigm of convolution in ConvNeXt [21] into TCN to jointly model temporal and spectral information. Additionally, a convolutional self-attention module is incorporated to enhance feature selection, while a feature enhancement module is introduced to preserve shallow features. Experimental results demonstrate that ACN-TCN outperforms Waveformer and CATSE in both single-target and multi-target extraction tasks. Given the computational demands of the self-attention mechanism, we conduct an ablation study to evaluate the performance of the model without the convolutional self-attention layer. Even in its absence, the proposed time-frequency domain

Received 6 April 2025; revised 14 October 2025; accepted 20 October 2025. Date of publication 24 October 2025; date of current version 11 November 2025. The work was supported in part by the National Natural Science Foundation of China under Grant 61871213 and in part by the Project of China Disabled Persons Federation under Grant 2023CDPFHS-02. The associate editor coordinating the review of this article and approving it for publication was Prof. Zhizheng Wu. *(Corresponding author: Ruiyu Liang.)*

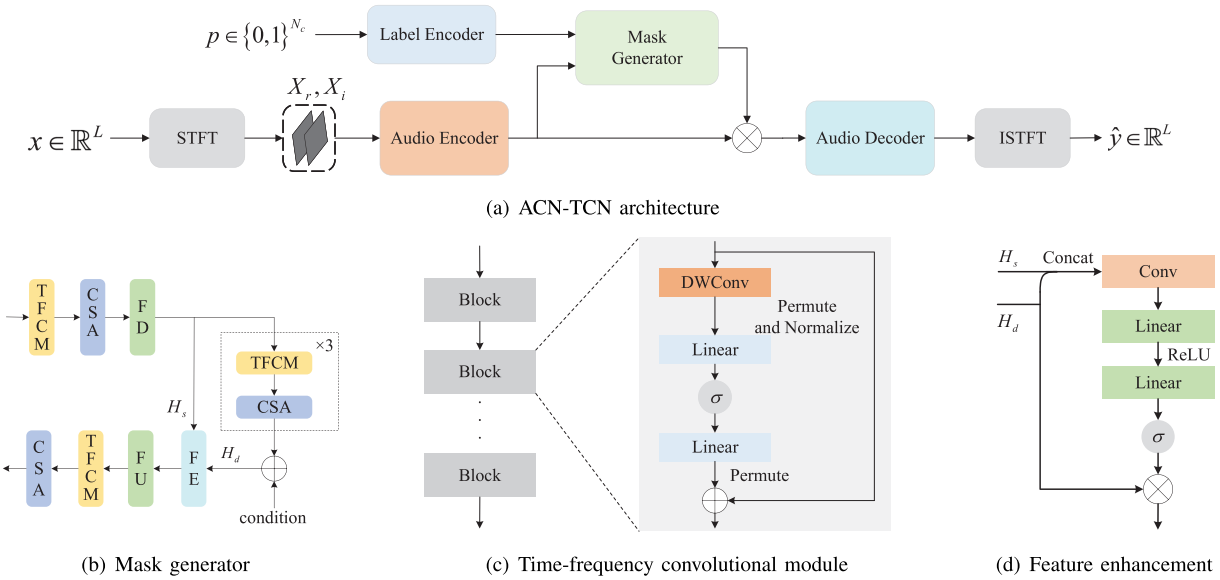
MinJie Xiang, Ye Ni, and Li Zhao are with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: 220230881@seu.edu.cn; niye@seu.edu.cn; zhaoli@seu.edu.cn).

Ruiyu Liang is with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China, and also with the School of Information and Communication Engineering, Nanjing Institute of Technology, Nanjing 211167, China (e-mail: liangry@njit.edu.cn).

Björn W. Schuller is with the Chair of Health Informatics (CHI), Technical University of Munich University Hospital, 81675 Munich, Germany, and also with the Group on Language, Audio, and Music (GLAM), Imperial College London, SW7 2AZ London, U.K. (e-mail: bjoern.schuller@imperial.ac.uk).

We provide code and audio samples: https://github.com/Xiang-M-J/ACN-TCN.

Digital Object Identifier 10.1109/LSP.2025.3625128



(a) ACN-TCN architecture

(b) Mask generator

(c) Time-frequency convolutional module

(d) Feature enhancement

Fig. 1. An overview of the proposed method.

model ACN-TCN achieves considerable improvements over previous time-domain approaches.

II. Methodology

For a time-domain audio mixture $x \in \mathbb{R}^L$ consisting of background noise n and several sounds s_i indexed by i, we can express the mixture as:

$$x = n + \sum_i s_i. \tag{1}$$

TSE aims to extract the target sound from the given mixture x. To specify which sounds should be retained, a 1D vector $p \in \{0,1\}^{N_c}$ is used as conditional information. In this vector, positions corresponding to the target category to be preserved are set to 1, while all others are set to 0. Here, N_c represents the number of sound categories. The model $\mathcal{M}$, parameterized by θ, predicts the target sound from the audio mixture and the condition vector.

$$\hat{y} = \mathcal{M}(x, p; \theta) \tag{2}$$

A. System Overview

The overall framework of ACN-TCN is shown in Fig. 1(a). Firstly, the short-time Fourier transform (STFT) is performed on the sound mixture to get the complex spectrum. The real and imaginary components of the complex spectrum are then stacked along the channel dimension, forming the input tensor $X \in \mathbb{R}^{2\times T\times F}$, where T and F represent the size of the time dimension and the frequency dimension, respectively. Both the audio encoder and audio decoder consist of a 2D convolutional layer with a kernel size of 3×3. To maintain the causality in the time dimension, zero-padding of size 2 is applied to the beginning of the input sequence before convolution. The audio encoder maps X into a high-dimensional tensor $H \in \mathbb{R}^{C\times T\times F}$, while the audio decoder restores the size of the channel dimension back to 2.

For the conditional input $p \in \{0,1\}^{N_c}$, a label encoder is employed to convert p into the conditional embedding $c \in \mathbb{R}^{C\times 1\times F_h}$. The label encoder consists of five layers; the first four layers are fully connected layers with layer normalization and rectified linear unit (ReLU) activation, each having an output dimension of 512. The final layer is also a fully connected layer with layer normalization, but its output dimension is $\mathbb{R}^{C\times F_h}$, where F_h is half of the frequency dimension size after STFT.

To facilitate model prediction, a mask generator is employed instead of directly estimating the target spectrum, as shown in Fig. 1(b). It first learns representations of various sound sources in mixed audio. These deep representations are then together with the conditional embedding c to obtain the features related to the target sound. Finally, the target-related features are decoded to predict the mask. The element-wise multiplication between the predicted mask and the audio encoder output produces a spectral representation, which is subsequently processed by the audio decoder to obtain the estimated target sound spectrum. Finally, the inverse short-time Fourier transformation (ISTFT) is applied to reconstruct the estimated target audio $\hat{y}$ in the time domain.

B. Time-Frequency Convolutional Module

The structure of the time-frequency convolutional module (TFCM) is shown in Fig. 1(c). Unlike recent time–frequency domain separation models [18], [22] that process the time and frequency domains independently, and previous time-domain TSE methods that perform modeling solely in the time domain, the TFCM module is designed to jointly model both domains within a unified convolutional framework, enabling more effective cross-domain interaction and representation learning. TFCM stacks N convolutional blocks, where $N = 6$ in the experiment. The convolutional block adopts an architecture similar to ConvNeXt [21], which utilizes depthwise convolutions [23], [24] with enlarged kernels for efficient contextual modeling

and integrates feed-forward layers to strengthen the nonlinear representation capacity. DWConv is a depthwise convolution layer with a kernel size of 2×7. The dilation size in the time dimension increases exponentially with the block level $(2^0, 2, \ldots, 2^{N-1})$, whereas the frequency-domain dilation size remains fixed to 1.

After DWConv, the shape of the feature is permuted from $\mathbb{R}^{C\times T\times F}$ to $\mathbb{R}^{T\times F\times C}$, and the feature is normalized along the channel dimension using layer normalization. The normalized feature is then processed by a feedforward network (FFN) consisting of two fully connected layers, with a Gaussian error linear units (GELU) activation function in between. The output dimension of the first fully connected layer in FFN is four times that of the input dimension. The output of the FFN is permuted back to the original shape, and a residual connection is applied by adding the block input to the output.

C. Convolutional Self-Attention

The design of Convolutional Self-Attention (CSA) differs from the vanilla attention mechanism in a common Transformer [25]. Based on the implementation in TF-Gridnet [22], the layers for mapping query Q, key K, value V, and output in CSA are replaced by a 2D convolutional layer with a kernel size of 1×1. After each convolutional layer, ReLU activation and layer normalization along the channel and frequency dimensions are applied. The tensors Q, K and V are reshaped from $\mathbb{R}^{C\times T\times F}$ to $\mathbb{R}^{N\times T\times (F*H)}$, where N is the number of heads and $H = C/N$ is the dimension of the heads. The attention matrix is obtained by performing a matrix multiplication between Q and the transposed tensor of K. To ensure causality, the right half of the attention matrix will be masked by setting the values in that part to negative infinity. After scaling and applying softmax to the masked attention matrix, it is multiplied by V to obtain the output tensor O with added attention

$$O = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{F \times H}}\right) V, \tag{3}$$

where $K^{\top} \in \mathbb{R}^{N\times (F*H)\times T}$.The tensor O is reshaped back to original dimensions and then passed through a transformation layer before being added to the input, forming the final output of the CSA.

D. Frequency Down-Sampling and Up-Sampling

To mitigate computational overhead while leveraging the rich information in the time-frequency domain, we adopted a strategy of first compressing the frequency dimension and then restoring it. Frequency down (FD) consists of a 2D convolutional layer followed by a parametric rectified linear unit (PReLU) activation function. The convolution kernel size is 1×3, and the stride along the frequency dimension is 2.

Frequency up (FU) consists of two convolutional blocks, with settings similar to those in FD. The first block uses a 2D convolutional layer with a kernel size of 1×3, while the second block employs a 2D transpose convolutional layer with a stride of 2 and the same kernel size as the first block, restoring the frequency dimension to its original size.

E. Feature Enhancement

The feature enhancement (FE) block contains a convolutional layer with a kernel size of 1×3 and FFN, as illustrated in Fig. 1(d). To integrate features of different resolutions, we concatenate the shallow features' H_s output by FD with the deep features H_d containing the embedded conditional information along the channel dimension. The concatenated tensor then passes through the convolutional layer with a kernel size of 1×3 to reduce the channel dimension of the concatenated tensor to match that of H_d.

The FFN in FE consists of two fully connected layers with ReLU activation in between. The input and output sizes of the feed-forward layer are the size of the frequency dimension, and the intermediate dimension is set to half of the input dimension. Finally, a weight is obtained using a Sigmoid activation. The weight and H_d are element-wise multiplied to enhance the feature.

F. Loss Function

Let $y \in \mathbb{R}^L$ denote the clean target sound, and let $\hat{y} \in \mathbb{R}^L$ denote the estimated target sound. The loss function is the weighted sum of the signal-to-noise-ratio (SNR) loss and the scale-invariant signal-to-noise ratio (SI-SNR) loss.

$$L_{snr} = 10\log_{10}\frac{\|y\|_2}{\|y-\hat{y}\|_2} \tag{4}$$

$$L_{si-snr} = 10\log_{10}\frac{\|\alpha y\|_2}{\|\alpha y-\hat{y}\|_2}, \alpha = \frac{\langle y, \hat{y}\rangle}{\|y\|_2} \tag{5}$$

$$L = -\left(0.9 * L_{snr} + 0.1 * L_{si-snr}\right) \tag{6}$$

III. Experiments and Analysis

A. Experiment Settings

The dataset setup follows the configuration in Waveformer [9]. Each sound mixture contains a background noise and 3–5 foreground sounds from different categories. The foreground sounds are sourced from the FSDKaggle2018 dataset [26], while the background noise comes from the TAU Urban Acoustic Scenes 2019 dataset [27]. The FSDKaggle2018 dataset contains 41 categories of sounds and a total of 11,073 audio clips. The scaper toolkit [28] is used to generate mixtures, where the SNR of each foreground sound relative to the background noise is randomly distributed between 15–25 dB. The duration of the foreground sounds is cropped to 3–5 seconds. Each sound mixture lasts 6 seconds, with a sampling rate of 16,000 Hz. The dataset includes 50 k training clips, 5 k validation clips, and 10 k testing clips. For single-target extraction, one foreground sound is randomly selected as the target. In the multi-target extraction model, up to three sounds are randomly selected as targets.

The STFT implementation is based on the Asteroid [29] package, with a filter size and kernel size of 256, a stride of 128, and a Hann window. The output channel size C of the audio encoder in the model is set to 32, and the channel dimension remains unchanged until the audio decoder reduces it to 2. The

TABLE I
SINGLE-TARGET SOUND EXTRACTION RESULTS, ΔSNR AND ΔSI-SNR REFER TO THE INCREASE OF SNR AND SI-SNR RELATIVE TO MIXTURES

Models	FLOPs (G)	Params (M)	Δ SNR (dB)	Δ SI-SNR (dB)
Conv-TasNet[14]	2.42	9.74	12.08	8.70
Waveformer[9]	2.79	3.62	12.68	9.09
CATSE[10]	-	3.52	13.28	9.53
ACN-TCN (proposed)	3.24	2.16	**15.77**	**12.99**

TABLE II
MULTI-TARGET SOUND EXTRACTION RESULTS. THE SETTINGS OF ALL MODELS IN THIS TABLE ARE IDENTICAL TO THOSE USED FOR THE SINGLE-TARGET SOUND EXTRACTION TASK BUT HAS BEEN RE-TRAINED TO PERFORM THE MULTI-TARGET TASK

Models	ΔSNR (dB) / ΔSI-SNR (dB)		
	1 target	2 targets	3 targets
Waveformer[9]	12.77/9.39	6.33/4.65	2.05/1.30
CATSE[10]	13.02/10.07	6.44/4.97	2.84/2.26
ACN-TCN (proposed)	**15.88/13.40**	**8.77/7.52**	**4.60/4.14**

model is trained using the Adam optimizer with a learning rate of 5e-4, and a batch size of 4. The single-target extraction model is trained for 20 epochs, while the multi-target model is trained for 30 epochs.

To evaluate the proposed model, we compare it with two previous causal TSE models Waveformer [9] and CATSE [10]. Additionally, Conv-TasNet [14], originally designed for speech separation, is adapted for TSE task.

B. Single-Target TSE

Table I compares the single-target extraction performance of three previous causal models and the proposed model in this letter. Because of the effective utilization of time-frequency domain information, ACN-TCN achieves considerable improvements in both SNR and SI-SNR metrics. Compared to CATSE, ACN-TCN improves the SNR by 2.5 dB and the SI-SNR by 3.4 dB. The number of parameters of ACN-TCN is reduced by 40% compared to Waveformer and CATSE, while the flops have increased by 16% relative to Waveformer.

C. Multi-Target TSE

Table II presents the results of the three models for the multi-target extraction task. The columns labeled with 1, 2, and 3 targets indicate cases where 1, 2, or 3 target sounds are randomly selected during testing, respectively. A comparison with single-target TSE results reveals that multi-target TSE task provides some improvement over single-target extraction. As the number of selected targets increases, the model's improvement diminishes. This is likely because selecting more target sounds increases the SNR of the target audio relative to the original mixture, which in turn limits the potential improvement of the model. However, ACN-TCN still demonstrates an obvious advancement, especially in the extraction of the 3-target extraction scenario. Compared with CATSE, SNR improves by 60% and by 83% , highlighting the effectiveness of the proposed approach.

TABLE III
ABLATION RESULTS UNDER THE SETTING OF SINGLE-TARGET EXTRACTION

	Frequency dilation size	ΔSNR (dB)	Δ SI-SNR (dB)
w/o FE	$2^i, i = 0, 1, ...$	15.16	11.98
w/o FE	1	6.97	12.00
w/o CSA	1	14.66	11.52
ACN-TCN (proposed)	$2^i, i = 0, 1, ...$	15.44	12.22
ACN-TCN (proposed)	1	15.77	12.99

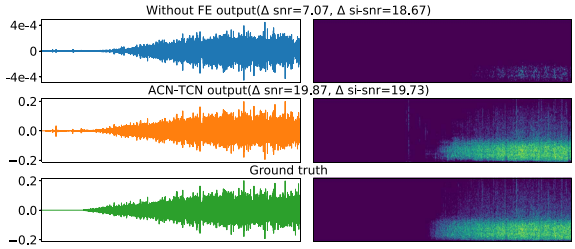


Fig. 2. Visualization of output waveforms of single target extraction with ACN-TCN without FE, ACN-TCN, and ground-truth.

D. Ablation Study

Table III demonstrates that applying identical dilation configurations to both the time and frequency dimensions results in degraded model performance. When the dilation size in the frequency dimension matches that of the time dimension, FE yields only a slight improvement in SNR and SI-SNR. However, when the frequency dilation size is always 1, FE can considerably improve the SNR. This phenomenon can be explained as follows: the output of the FD contains shallow information related to the amplitude of the signal. As features pass through multiple layers, this amplitude-related information is projected into a higher-dimensional space. When the dilation size of the frequency is set to 1, the model struggles to capture this shallow information, but can still extract deep information, such as the shape of the signal. This may lead to a situation where the SNR is low but the SI-SNR is high.

Fig. 2 illustrates that the output signal shape of the model without FE is similar to the ground truth, but the amplitude is considerably lower. Since the TFCM in ACN-TCN is designed to learn simultaneously in the time-frequency domain, even after removing CSA, it still outperforms time-domain models such as Waveformer and CATSE.

IV. CONCLUSION

In this letter, we proposed a causal TSE model that operates in the time-frequency domain. ACN-TCN incorporates a novel convolutional kernel design into TCN, allowing it to better capture information in both the time and frequency domains while preserving causality. This model leverages the TFCM and CSA to learn masks for specific target audio, and utilizes FD to reduce the number of parameters. Consequently, it achieves superior performance compared to previous time-domain methods while maintaining a lightweight parameter configuration. In the future, we will focus on exploring solutions for unknown target sound labels and the integration of multimodal conditional inputs to further enhance model performance and adaptability.

REFERENCES

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

[2] H. Wang, J. Hai, Y.-J. Lu, K. Thakkar, M. Elhilali, and N. Dehak, "Soloaudio: Target sound extraction with language-oriented audio diffusion transformer," in *Proc. 2025 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2025, pp. 1–5.

[3] E. Karamatlı, A. T. Cemgil, and S. Kırbız, "Audio source separation using variational autoencoders and weak class supervision," *IEEE Signal Process. Lett.*, vol. 26, no. 9, pp. 1349–1353, Sep. 2019.

[4] M. Delcroix, J. B. Vázquez, T. Ochiai, K. Kinoshita, and S. Araki, "Few-shot learning of new sound classes for target sound extraction," in *Proc. Interspeech 2021*, 2021, pp. 3500–3504.

[5] M. Delcroix, J. B. Vázquez, T. Ochiai, K. Kinoshita, Y. Ohishi, and S. Araki, "SoundBeam: Target sound extraction conditioned on sound-class labels and enrollment clues for increased performance and continuous learning," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 121–136, 2023.

[6] J. Hai, H. Wang, D. Yang, K. Thakkar, N. Dehak, and M. Elhilali, "DPM-TSE: A diffusion probabilistic model for target sound extraction," in *Proc. 2024 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 1196–12000.

[7] X. Liu et al., "Separate anything you describe," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 33, pp. 458–471, 2025.

[8] C. Hernandez-Olivan et al., "Soundbeam meets M2D: Target sound extraction with audio foundation model," in *Proc. 2025 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2025, pp. 1–5.

[9] B. Veluri, J. Chan, M. Itani, T. Chen, T. Yoshioka, and S. Gollakota, "Real-time target sound extraction," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[10] S. Baligar, M. Kegler, B. Irvin, M. Stamenovic, and S. Newsam, "CATSE: A context-aware framework for causal target sound extraction," in *Proc. 32nd Eur. Signal Process. Conf.*, 2024, pp. 401–405.

[11] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 47–54.

[12] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[13] K. Wakayama et al., "Online target sound extraction with knowledge distillation from partially non-causal teacher," in *Proc. 2024 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 561–565.

[14] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[15] D. Lee and J.-W. Choi, "DeFT-AN: Dense frequency-time attentive network for multichannel speech enhancement," *IEEE Signal Process. Lett.*, vol. 30, pp. 155–159, 2023.

[16] R. Yu, Z. Zhao, and Z. Ye, "PFRNet: Dual-branch progressive fusion rectification network for monaural speech enhancement," *IEEE Signal Process. Lett.*, vol. 29, pp. 2358–2362, 2022.

[17] H. Kim and J. W. Shin, "On training speech separation models with various numbers of speakers," *IEEE Signal Process. Lett.*, vol. 30, pp. 1202–1206, 2023.

[18] M. Xu, K. Li, G. Chen, and X. Hu, "Tiger: Time-frequency interleaved gain extraction and reconstruction for efficient speech separation," in *Proc. 13th Int. Conf. Learn. Representations*, vol. 2025, pp. 70205–70222.

[19] T. Kaneko, K. Tanaka, H. Kameoka, and S. Seki, "iSTFTNet: Fast and lightweight mel-spectrogram vocoder incorporating inverse short-time Fourier transform," in *Proc. 2022 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 6207–6211.

[20] H. Siuzdak, "Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 1–15.

[21] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.

[22] 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 *Proc. 2023 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[24] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[25] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.

[26] E. Fonseca et al., "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Proc. Scenes Events 2018 Workshop (DCASE2018)*, 2018, pp. 69–73.

[27] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. Scenes Events 2018 Workshop (DCASE2018)*, 2018, pp. 9–13.

[28] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. 2017 IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2017, pp. 344–348.

[29] M. Pariente et al., "Asteroid: The pytorch-based audio source separation toolkit for researchers," in *Proc. Interspeech*, 2020, pp. 2637–2641.