INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

A Neural State-Space Model Approach to Efficient Speech Separation

Chen Chen[1], *Chao-Han Huck Yang*[2], *Kai Li*[3], *Yuchen Hu*[1], *Pin-Jui Ku*[3], *Eng Siong Chng*[1]

[1]Nanyang Technological University, Singapore
[2]Georgia Institute of Technology, USA
[3]Tsinghua University, China

chen1436@e.ntu.edu.sg

Abstract

In this work, we introduce S4M, a new efficient speech separation framework based on neural state-space models (SSM). Motivated by linear time-invariant systems for sequence modeling, our SSM-based approach can efficiently model input signals into a format of linear ordinary differential equations (ODEs) for representation learning. To extend the SSM technique into speech separation tasks, we first decompose the input mixture into multi-scale representations with different resolutions. This mechanism enables S4M to learn globally coherent separation and reconstruction. The experimental results show that S4M performs comparably to other separation backbones in terms of SI-SDRi, while having a much lower model complexity with significantly fewer trainable parameters. In addition, our S4M-tiny model (1.8M parameters) even surpasses attention-based Sepformer (26.0M parameters) in noisy conditions with only 9.2% of multiply-accumulate operation (MACs).

Index Terms: Speech separation, state-space model, ordinary differential equations

1. Introduction

Speech separation (SS) aims to separate target speech from overlapping speech signal sources [1], also known as *cocktail party problem*. SS widely serve as a pre-processor for speech applications [2, 3], e.g., automatic speech recognition [4, 5] and speaker verification [6]. Recently, SS has gained remarkable progress driven by the power of deep learning [7, 8, 9], where the clean speech of individual speakers serves as ground truth to supervise the training of the neural network [10].

Developing an efficient SS architecture with low model complexity is challenging due to the high-dimensional input of speech signals, which contains tens of thousands of time steps per second and exhibits long-range behaviors at multiple timescales. In order to handle this challenge, previous deep learning-based attempts have tailored standard sequence modeling approaches like CNNs [11], RNNs [12, 13], and Transformers [14] to predict clean speech from a mixture. However, these works have different limitations.CNNs are constrained by the size of the receptive field, making it difficult to achieve global coherence [15]. RNNs lack computational efficiency because they cannot be parallelized during training. While Transformers-based [16] architectures achieve impressive performance on a public dataset, their vast network size (e.g., Sepformer with 26.0M parameters [14]) results in high computational costs for training and inference, hampering the application of the trained model in practical scenarios.

The computational work for this article was fully/partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg)

To improve the efficiency for SS, we are inspired by the recent advances in neural state-space model (SSM) [17], which have shown outstanding performance in high-rate audio generation tasks [15]. The globally coherent generation of SSM is similar to self-attention mechanism in Transformers, but with significantly fewer trainable parameters are required in SSM. Consequently, we believe that SSM offers a solution to reduce the model complexity of SS, thus improving the separation efficiency for both training and inference.

In this paper, we introduce an efficient SS method called S4M (speech separation using state-space model), which follows the mainstream encoder-decoder pipeline. Specifically, the encoder in S4M extracts multiple features with varying resolutions from a flat input mixture, and then feeds them into S4 blocks to capture the representation with global long-range dependencies. Similarly, S4 layer is also employed in the decoder for feature reconstruction. The main strengths of S4M are summarized as follows:

- S4M offers significant advantages over mainstream SS methods, in terms of model complexity and computational cost.
- S4M effectively captures long-range dependencies for high-rate waveforms, which benefits separated feature reconstruction, especially in noisy conditions.

To demonstrate these strengths of S4M, we conducted experiments on clean datasets WSJ0-2Mix and LibriMix, as well as the noisy dataset LRS2-Mix. The experimental results show that S4M achieves comparable performance with other competitive baselines in clean conditions, and achieves state-of-the-art performance on LRS2-Mix, which includes practical noise and reverberation in the mixture. Furthermore, we compared the model complexity of S4M with other models, and results show that S4M has remarkable superiority in terms of computational cost and inference time, making it one potential solution for streaming-based speech separations [18].

2. Background: State-Space Models

Given the input mixture $x \in \mathbb{R}^{1\times T}$, the goal of speech separation is to separate and predict clean speech $y^n \in \mathbb{R}^{1\times T}$ for each speaker, where n is the number of speakers. CNNs and RNNs are the most widely used models for speech separation, each with its own advantages and limitations during training and inference. Specifically, a CNN layer computes a convolution with parameterized kernels

$$K = (k_0, \cdots, k_{w-1}) \qquad y^n = K * x \tag{1}$$

where w is the width of the kernel. The receptive field or context size of a CNN is determined by the sum of kernel widths across all layers. As duration T is usually large for speech signal, this

10.21437/Interspeech.2023-696

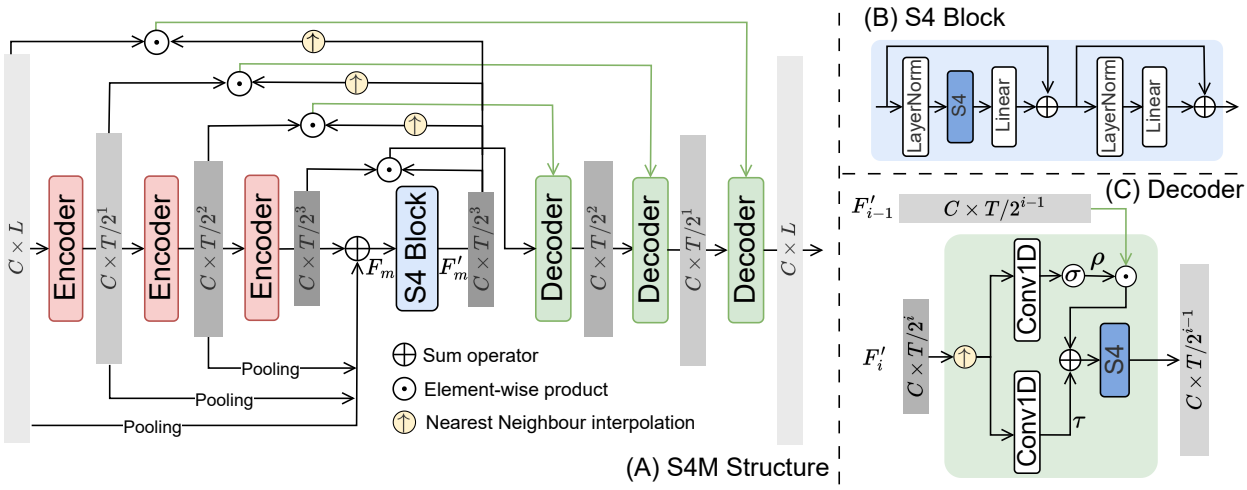

Figure 1: *The block diagram of the (A) S4M model, (B) S4 Block, and (C) Decoder. "σ" denotes the Sigmoid function. The grey chunks denote the hidden features after each layer.*

results in increased computational complexity. To address this, a variant of CNNs called dilated convolution (DCNN) is widely used in SS, where each kernel K is non-zero only at its endpoints [19]. On the other hand, RNNs sequentially compute a hidden state h_t from a previous history state h_{t-1} and current input x. The output y is modeled as:

$$h_t = f(h_{t-1}, x) \qquad y = g(h_t) \tag{2}$$

f is also known as an RNN cell, such as the popular LSTM. The recently proposed deep neural state-space model(SSM) advances speech tasks by combining the properties of both CNNs and RNNs. The SSM [17] is defined in continuous time using the following equations:

$$h'(t) = Ah(t) + Bx(t) \tag{3}$$
$$y(t) = Ch(t) + Dx(t) \tag{4}$$

To operate on discrete-time sequences sampled with a step size of Δ, SSM can be computed with recurrence as follows:

$$h_k = \overline{A}h_{k-1} + \overline{B}x_k \qquad y_k = \overline{C}h_k + \overline{D}x_k \tag{5}$$
$$\overline{A} = (I - \Delta/2 \cdot A)^{-1}(I + \Delta/2 \cdot A) \tag{6}$$

where $\overline{A}, \overline{B}, \overline{C}, \overline{D}$ are the discretized state matrices. According to [15], Eq.(5) can be rewritten as a discrete convolution:

$$y_k = \overline{C}\overline{A}^k\overline{B}x_0 + \overline{C}\overline{A}^{k-1}\overline{B}x_1 + \cdots + \overline{C}\overline{B}x_k \tag{7}$$
$$y = \overline{K} * x \qquad \overline{K} = (\overline{C}\overline{B}, \overline{C}\overline{A}\overline{B}, \overline{C}\overline{A}^2\overline{B}) \tag{8}$$

$\overline{K}$ is the SSM convolution kernel. The Eq.(8) is a single (non-circular) convolution and can be computed very efficiently with Fast Fourier Transformation, provided that $\overline{K}$ is known.

In order to calculate $\overline{K}$, we employ a specific instantiation of SSM, known as **S4 layer** [17], which parameterizes A as a diagonal plus low-rank (DPLR) matrix: $A = \Lambda - pq^*$. This parameterization has three advantages: 1) Faster computation. The kernel $\overline{K}$ in Eq.(8) can be computed very quickly in this setting. 2) Improved capture of long-range dependencies. This parameterization includes HiPPO matrices [20], which theoretically and empirically allow SSM to better capture global correspondence from input. 3) Better stability. SSM involves the spectrum of the state matrix A, which is more easily controlled since $-pp^*$ is always a negative semi-definite matrix [15].

Given any time step Δ, the computation of the SSM convolution kernel $\overline{K}$ requires $\mathcal{O}(S + L)$ operations and $\mathcal{O}(S + L)$ space, where S is the state size and L is the length of input.

3. S4M: State-Space Speech Separation Model

The overview structure of S4M is shown in Fig 1, where an encoder-decoder pipeline with S4 block is employed for speech separation tasks. As a time-domain method, S4M typically converts input waveform $x \in \mathbb{R}^{1\times T}$ to 2D features $F_0 \in \mathbb{R}^{C\times L}$ using a 1-D convolutional layer, where C and L represent the channel number and feature length respectively.

3.1. Encoder and S4 Block

Prior works [21, 22] have demonstrated the advantages of using multi-scale representations with different resolutions for speech tasks. Consequently, we stack three down-sampling encoders (red blocks in Fig. 1) that consists of a 1-D dilation convolution layer, followed by a global normalization layer. The dilation factor is set as 2 to gradually increase the receptive field. In this way, the length dimension L of feature is squeezed layer by layer, shown as the grey chunks in Fig. 1. Subsequently, a set of representations $F = \{F_i \in \mathbb{R}^{C\times \frac{L}{2^{i-1}}} \mid i = 0, 1, \cdots\}$ with same channel C but different length L are extracted from input, where i is set as 4 in this paper.

To integrate the information from the multi-scale representations, we perform average pooling on the features from shallow layer to reshape them, and then add them to obtain feature $F_m \in \mathbb{R}^{C\times \frac{L}{8}}$. To capture global correspondence from F_m, a residual S4 block is employed (shown as blue box in Fig.1). Specifically, it contains a normalization layer, a S4 layer with GELU activation function [23], and a linear layer. We also use additional point-wise linear layers in the style of a feed-forward

network in Transformer, along with a residual connection to avoid the vanishing gradient problem. Notably, the S4 block does not change the shape of feature, therefore, the feature of F'_m with shape of $C \times \frac{L}{8}$ is obtained after the S4 block.

3.2. Decoder

The decoder of S4M progressively reshapes the separated features to maintain symmetry with the encoder. As shown in Fig.1, two decoder inputs F'_{i-1} and F'_i are obtained by the element-wise multiplication between F_{i-1} and F'_m, as well as F_i and F'_m. This mask-based operation is commonly used in speech separation tasks. In addition, up-sampling of nearest neighbour interpolation is required for F'_m due to shape mismatch.

Given F_{i-1} and F_i, the decoder first employs a light local attention mechanism [22] using adaptive parameters ρ and τ, which are respectively denoted as:

$$\tau = f_2(\phi(F'_i)) \qquad \rho = \sigma(f_1(\phi(F'_i))) \tag{9}$$

where f_1 and f_2 are two 1-D convolutional layers followed by normalization layer, ϕ denotes the nearest neighbor interpolation along time dimension L for up-sampling ($C \times \frac{L}{2^i} \rightarrow C \times \frac{L}{2^{i-1}}$), and σ denotes the Sigmoid function. As F'_{i-1}, ρ, and τ have the same shape, the local attention process is formulated by:

$$F'_{i-1} = \rho \odot F'_{i-1} + \tau \tag{10}$$

Then the same S4 block is employed for globally coherent generation after local attention. As shown in Fig. 1, the output of decoder is recursively multiplied by the output of encoder to get F'_{i-2} and then fed it into next decoder layer until the output shape is restored to $C \times L$.

We adopt unfolding scheme for the network as proposed in A-FRCNN [24]. Concretely, the structure shown in Fig. 1 (A) is repeated for B times (weight sharing), such that the input of the current model is also added by each previous model's output.

3.3. Training objective

The objective of training the end-to-end S4M is to maximize the scale-invariant source-to-noise ratio (SI-SNR), which is commonly used as the evaluation metric for source separation. The SI-SNR loss is defined as:

$$\mathcal{L}_{si-snr} = -\sum_{n=1}^{N} 10\log_{10}\left(\frac{\left\|\frac{\hat{y}_n^T y_n}{\|y_n\|^2} y_n\right\|^2}{\left\|\frac{\hat{y}_n^T y_n}{\|y_n\|^2} - \hat{y}_n\right\|^2}\right) \tag{11}$$

where $y^{(n)}$ is the ground-truth signal for speaker n, and $\hat{y}^{(n)}$ is the estimated time-domain speech produced by S4M.

Furthermore, Utterance-level permutation invariant training (uPIT) is applied during training to address the source permutation problem [25].

4. Experiment

4.1. Database

We evaluate S4M and other competitive methods on both clean and noisy datasets, including WSJ0-2Mix [36], LibriMix [37] and LRS2-Mix [22]. To ensure the generality, the mixture in test set are generated by the speakers that are not seen during training.

WSJ0-2Mix is the most common speech separation dataset derived from Wall Street Journal (WSJ0). It consists of a 30 hours of training set (20k utterances), a 8 hours of validation set (5k utterances), and a 5 hours of test set (3k utterances). All utterances are re-sampled to 8 kHz for comparison with other works.

LibriMix. Considering the limited data amount of WSJ0-2mix, we further employ LibriMix dataset to evaluate the performance in clean condition. The target speech in LibriMix is randomly drawn from the train-100 subset of LibriSpeech dataset with 8 kHz sampling rate. Each mixture uniformly samples Loudness Units relative to Full Scale (LUFS) between -25 and -33 dB. The training set contains 13.9k utterances with duration of 58 hours, while the validation set and test set both contain 3k utterances with duration of 11 hours.

LRS2-Mix. The source of LRS2-Mix is LRS2 dataset [38] that includes thousands of video clips from BBC. It contains practical noise and reverberation interference, which is more close to reality. We randomly select utterances of 16 kHz from different scenes and mix them with signal-to-noise ratios sampled between -5 dB and 5dB. In practice, we utilize the same mixing script as WSJ0-2Mix, in which the training set, validation set and test set contain 20k, 5k and 3k utterances respectively.

Model	SI-SDRi (dB)	SDRi (dB)	# Para. (M)
ADANet [26]	9.1	10.4	9.1
WA-MISI-5 [27]	12.6	13.1	32.9
SPN [28]	15.3	15.6	56.6
Conv-TasNet [11]	15.3	15.6	5.1
Deep CASA [29]	17.7	18.0	12.8
FurcaNeXt [30]	-	18.4	51.4
TDANet [22]	18.6	18.9	2.3
DPRNN [12]	18.8	19.0	2.6
SUDO RM-RF [31]	18.9	-	6.4
Gated DPRNN [32]	20.1	20.4	7.5
Sepformer [14]	20.4	20.5	26.0
Wavesplit [33]	21.0	21.2	29.0
SFSRNet [34]	22.0	22.1	59.0
TF-GridNet [35]	**23.4**	**23.5**	14.4
S4M-tiny	19.4	19.7	**1.8**
S4M	20.5	20.7	3.6

Table 1: *SI-SDRi and SDRi results on WSJ0-2Mix. "# Para." denotes the number of trainable parameters for each model. Best results are in bold.*

Model	LibriMix		LRS2-Mix		# Para.
	SI-SDRi	SDRi	SI-SNRi	SDRi	(M)
BLSTM-TasNet	7.9	8.7	6.1	6.8	23.6
Conv-TasNet	12.2	12.7	10.6	11.0	5.6
DPRNN	16.1	16.6	12.7	13.0	2.7
SuDoRM-RF	14.0	14.4	11.3	11.7	6.4
Sepformer	16.5	17.0	13.5	13.8	26.0
WaveSplit	16.6	17.2	13.1	13.4	29.0
A-FRCNN	16.7	17.2	13.0	13.3	6.1
TDANet	**17.4**	**17.9**	14.2	14.5	2.3
S4M-tiny	16.2	16.6	14.2	14.5	**1.8**
S4M	16.9	17.4	**15.3**	**15.5**	3.6

Table 2: *SI-SDRi and SDRi results on LibriMix and LRS2-Mix.*

4.2. S4M Setup

The kernel size of convectional layer to process time domain signal is set as 4ms and the stride size is set as 1ms. The number of channels in the dilation convolution layer of the encoder and the number of hidden units in all linear layers are both set as 512. For S4 layer, we found that the model performs the best when the number of channels is set as 16. Furthermore, we develop a lighter version of S4M called *S4M-tiny*, which removes

the S4 layers (dark blue block in Fig. 1-C) in the decoder. It is worth noting that the S4M-tiny only contains 1.8M trainable parameters.

Both S4M and S4M-tiny are trained for 200 epochs with a learning rate of 0.001. An early-stopping strategy is adopted when validation loss does not decrease for 5 epochs. To avoid gradient explosion, we apply gradient clipping with a maximum L2 norm of 5 during training.

4.3. Evaluation Metric

We asses the clarity of separated audios based on scale-invariant signal-to-distortion ratio improvement (SI-SDRi) and signal-to-distortion ratio improvement (SDRi). To evaluate model efficiency, we measure the processing time consumption per second for all models, indicated by real-time factor (RTF) in the tables. RTF is calculated by processing ten audio tracks of 1 second in length and 16 kHz in sample rate on CPU and GPU (total processing time / 10), represented as "CPU-RTF" and "GPU-RTF" respectively. The numbers are then averaged after running 1000 times. Also, we use the parameter size and the number of multiply-accumulate operations (MACs) to measure the model size. MACs are calculated using the open-source tool PyTorch-OpCounter4 under the MIT license.

5. Result and Analysis

5.1. Main results

We report our main results on the test set of WSJ0-2Mix in Table 1, as well as LirbriMix and LRS3-Mix in Table 2. On WSJ0-2Mix dataset, S4M surpasses CNN-based Conv-TasNet and RNN-based DPRNN, and achieves comparable SI-SDRi performance with Transformer-based Sepformer, with far lower model complexity. In addition, S4M-tiny achieves 19.4 SI-SDRi performance with only 1.8M parameters, which demonstrates the efficiency of state-space model. For LibriMix dataset, we observe that S4M surpasses the Sepformer by 2.4% on SI-SDRi performance when training data doubles (from 30 hours to 58 hours).

We notice that S4M performs particularly well on LRS2-Mix which contains background noise and reverberation in the mixture. S4M-tiny even surpasses Sepformer by 13.3% (13.5 dB $\rightarrow$ 15.3 dB) in terms of SI-SDRi, with only 6.9% parameters. In addition, S4M achieves the best performance on LRS2-Mix in terms of both SI-SDRi and SDRi. This phenomenon indicates that S4M is effective to capture long-range dependencies for specific speaker, resulting in better noise-robustness in a more realistic environment.

5.2. Ablation Study on S4

We conduct ablation study on S4 module which serves as our main contribution. The results are summarized in Table 3. "Mid." denotes whether S4 block exists between the encoder and decoder (Fig. 1-B), and "S" represents the dimension of the state in S4 block. "Dec." denotes whether S4 layer is inserted in the decoder after local attention (Fig. 1-C), where the dimension of state is uniformly set as 16.

We observe that: 1) System 2 outperforms System 1 by a significant margin, highlighting the importance of S4 block which captures global correspondence from the multi-scale representations produced by the encoder and benefits subsequent separation. 2) The system achieves the best performance when the dimension of the state is set as 16. Increasing the value of "S" leads to an increase in the number of parameters and a degradation in speech separation performance. 3) S4 layer is also effective for feature reconstruction in the decoder, but it inevitably increases the number of parameters.

ID	Mid.	S	Dec.	SI-SDRi	SDRi	# Para.
1	✗	-	✗	10.5	10.9	0.23
2	✓	8	✗	13.9	14.3	1.82
3	✓	16	✗	14.2	14.5	1.84
4	✓	32	✗	14.0	14.4	1.88
5	✓	16	✓	15.3	15.5	3.59

Table 3: *Ablation study of S4 on LRS2-Mix dataset. "✓" denotes S4 layer exist in corresponding module. "✗" indicates the opposite.*

Model	GPU-RTF-*f* (ms)	GPU-RTF-*b* (ms)	CPU-RTF-*f* (s)	MACs (G/s)
BLSTM-TasNet	233.85	654.14	5.90	43.0
SuDoRM-RF	64.70	228.57	1.73	10.1
DPRNN	88.79	241.54	8.13	85.3
A-FRCNN	61.16	183.65	5.32	125.3
TDANet	23.77	97.92	1.78	9.1
TDANet (own)	61.25	368.54	5.97	9.1
Sepformer	65.61	184.91	7.55	86.9
TF-GridNet	100.52	285.37	86.4	128.7
S4M-tiny	**18.19**	**73.62**	**1.34**	**8.0**
S4M	40.15	132.83	2.57	38.7

Table 4: *Comparison of inference time and MACs on LRS2-Mix dataset."-f" and "-b" respectively stand for "feed-forward" and "backward" processes. For all metrics, lower is better.*

5.3. Analysis of model complexity

We analyze the model complexity of S4M, which also indicates the separation efficiency. Using RTF and MACs as metrics, we report the performance of S4M and its comparison with other models in Table 4. GPU-RTF-*f* and GPU-RTF-*b* indicate the training time for each model on GPU devices, while CPU-RTF-*f* denotes the inference speed on CPU devices when GPU resources is unavailable in some practical conditions. In addition, the "TDANet (own)" is reproduced without accelerated Transformer by Pytorch.

Table 4 shows that S4M-tiny consistently requires the least training and inference time on both GPU and CPU devices. Moreover, S4M-tiny can achieve better performance than Sepformer on LRS2-Mix dataset, with only 9.2% of MACs of Sepformer. For S4M, its model complexity is 4.8 times higher than S4M-tiny, but still significantly lower than Sepformer.

6. Conclusion

In this paper, we propose a efficient speech separation method (S4M) that achieves competitive performance while maintaining low model complexity. S4M utilizes state-space model to capture long-range dependencies from multi-scale representations, and integrates it into separated feature reconstruction. Experimental results show that S4M achieves comparable separation performance with significantly fewer trainable parameters in comparison with other mainstream methods. Furthermore, we analyze the model complexity using computing time and MACs, which shows that S4M provides a potential solution for streaming-based speech separation on mobile devices or streaming applications [39].

7. References

[1] S. Haykin and Z. Chen, "The cocktail party problem," *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[2] 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.

[3] K. Li, X. Hu, and Y. Luo, "On the use of deep mask estimation module for neural source separation systems," *arXiv preprint arXiv:2206.07347*, 2022.

[4] D. Yu and L. Deng, *Automatic speech recognition*. Springer, 2016, vol. 1.

[5] Y. Hu, C. Chen, R. Li, Q. Zhu, and E. S. Chng, "Gradient remedy for multi-task learning in end-to-end noise-robust speech recognition," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[6] A. E. Rosenberg, "Automatic speaker verification: A review," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 475–487, 1976.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] Y. Hu, C. Chen, H. Zou, X. Zhong, and E. S. Chng, "Unifying speech enhancement and separation with gradient modulation for end-to-end noise-robust speech separation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[9] Z. Zhang, C. Chen, X. Liu, Y. Hu, and E. S. Chng, "Noise-aware speech separation with contrastive learning," *arXiv preprint arXiv:2305.10761*, 2023.

[10] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1562–1566.

[11] 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.

[12] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[13] K. Li and Y. Luo, "On the design and training strategies for rnn-based online neural speech separation systems," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 21–25.

[15] K. Goel, A. Gu, C. Donahue, and C. Ré, "It's raw! audio generation with state-space models," in *International Conference on Machine Learning*. PMLR, 2022, pp. 7616–7633.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv:2111.00396*, 2021.

[18] Q. Wang, I. L. Moreno, M. Saglam, K. Wilson, A. Chiao, R. Liu, Y. He, W. Li, J. Pelecanos, M. Nika *et al.*, "Voicefilter-lite: Streaming targeted voice separation for on-device speech recognition," *arXiv preprint arXiv:2009.04323*, 2020.

[19] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[20] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent memory with optimal polynomial projections," *Advances in neural information processing systems*, 2020.

[21] C. Chen, N. Hou, D. Ma, and E. S. Chng, "Time domain speech enhancement with attentive multi-scale approach," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2021, pp. 679–683.

[22] K. Li, R. Yang, and X. Hu, "An efficient encoder-decoder architecture with top-down attention for speech separation," *arXiv preprint arXiv:2209.15200*, 2022.

[23] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[24] X. Hu, K. Li, W. Zhang, Y. Luo, J.-M. Lemercier, and T. Gerkmann, "Speech separation using an asynchronous fully recurrent convolutional neural network," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 509–22 522, 2021.

[25] 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 Transactions on Audio, Speech, and Language Processing*, vol. 25, 2017.

[26] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, 2018.

[27] Z.-Q. Wang, J. L. Roux, D. Wang, and J. R. Hershey, "End-to-end speech separation with unfolded iterative phase reconstruction," *arXiv preprint arXiv:1804.10204*, 2018.

[28] Z.-Q. Wang, K. Tan, and D. Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[29] Y. Liu and D. Wang, "Divide and conquer: A deep casa approach to talker-independent monaural speaker separation," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2092–2102, 2019.

[30] L. Zhang, Z. Shi, J. Han, A. Shi, and D. Ma, "Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," in *MultiMedia Modeling: 26th International Conference*. Springer, 2020, pp. 653–665.

[31] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo rm-rf: Efficient networks for universal audio source separation," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2020, pp. 1–6.

[32] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7164–7175.

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

[34] J. Rixen and M. Renz, "Sfsrnet: Super-resolution for single-channel audio source separation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 220–11 228.

[35] 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," *arXiv preprint arXiv:2209.03952*, 2022.

[36] 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)*. IEEE, 2016, pp. 31–35.

[37] 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.

[38] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2017.

[39] K. Hu, T. N. Sainath, A. Narayanan, R. Pang, and T. Strohman, "Transducer-based streaming deliberation for cascaded encoders," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8107–8111.