Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

Optimized Real-time Speech Enhancement with Deep SSMs on Raw Audio

Yan Ru Pei[1], *Ritik Shrivastava*[1], *FNU Sidharth*[1]

[1]BrainChip, USA

yanrpei@gmail.com, rshrivastava@brainchip.com, sidharth@brainchip.com

Abstract

We present aTENNuate, a simple deep state-space autoencoder configured for efficient online raw speech enhancement in an end-to-end fashion. The network's performance is primarily evaluated on raw speech denoising, with additional assessments on tasks such as super-resolution and de-quantization. We benchmark aTENNuate on the VoiceBank + DEMAND and the Microsoft DNS1 synthetic test sets. The network outperforms previous real-time denoising models in terms of PESQ score, parameter count, MACs, and latency. Even as a raw waveform processing model, the model maintains high fidelity to the clean signal with minimal audible artifacts. In addition, the model remains performant even when the noisy input is compressed down to 4000Hz and 4 bits, suggesting general speech enhancement capabilities in low-resource environments.

Index Terms: state-space models, autoencoder, denoising, super-resolution, de-quantization

1. Introduction

Speech enhancement (SE) is crucial for improving both human-to-human communication, such as in hearing aids, and human-to-machine communication, as seen in automatic speech recognition (ASR) systems. A challenging task of SE is removing background noises from speech signals, where the complexity of speech and noise patterns poses challenges [1]. Traditional speech enhancement methods such as Wiener filtering [2], spectral subtraction [3], and principal component analysis [4] have shown satisfactory performance in stationary noise environments, but their effectiveness is often limited in non-stationary noise scenarios, resulting in artifacts and substantial degradation in the intelligibility of the enhanced speech [5].

Deep learning audio denoising methods are trained on large datasets of clean and noisy audio pairs, and attempt to capture the nonlinear relationship between the noisy and clean signal features without prior knowledge of the noise statistics, as required by traditional denoising methods [6]. Speech features can be extracted from real or complex spectrograms of the noisy signal in the time-frequency domain, or directly from raw waveforms [7].

Many state-of-the-art deep learning denoising models leverage the feature extraction capability of convolutional networks. The UNet convolutional encoder-decoder is a common network architecture in denoising models [8, 9, 10] exemplified by Deep Complex UNet [11] and the generative adversarial SEGAN [12]. However, CNNs lack the capabilities to model long-range temporal dependencies present in speech signals, without incurring significant memory resources [13]. Another class of models that are more capable in this regard are RNNs, examples of which are DCCRN [14] and FRCRN [15]. However, these models typically show limited robustness to different types of noise and generalization across diverse audio sources [16]. Moreover, their non-linear recurrent nature prevents efficient usage of parallel hardware (e.g. GPUs) for training, which limits their potential for scaling.

Alternative approaches such as PercepNet [17] and RNNoise [18] have attempted to reduce network size and complexity by combining traditional speech enhancement methods with deep learning, resulting in smaller models with fewer parameters capable of running in real-time on general-purpose hardware. Most notably, DeepFilterNet [19] has leveraged speech-specific properties such as short-time speech correlations to achieve comparable results. On the contrary, methods that process raw waveform signals aim to maximize the expressive capabilities of deep networks without resorting to hand-engineered spectral conversions, as demonstrated by Facebook's DEMUCS model's real-time performance [20]. However, the majority of these models cannot perform real-time inference on general-purpose hardware, exemplified by Nvidia's CleanUNet model [21].

Here, we introduce the aTENNuate network, belonging to the class of Temporal Neural Networks [22, 23, 24], or TENNs. It is a deep state-space model (SSM) [25, 26] optimized for real-time denoising of raw speech waveforms on the edge. By virtue of being an SSM, the model is capable of capturing long-range temporal relationships present in speech signals, with stable linear recurrent units. Learning long-range correlations can be useful for capturing global speech patterns or noise profiles, and perhaps implicitly capture semantic contexts to aid speech enhancement performance [27]. During the training of the aTENNuate network, we can use the infinite impulse response (IIR) kernels of the SSM layers as long convolutional kernels over the input features, which can be parallelized using techniques such as FFT convolution [28] or associative scan [29]. During inference, the temporal convolution layers can be converted into equivalent recurrent layers for efficient real-time processing on mobile devices, minimizing latency and reducing the need for excessive buffering of data. The evaluation code is available on PyPI, installable with `pip install attenuate`.

2. Deep State-space Modeling

In this section, we briefly describe what state-space models (SSMs) are, and recall how they can be configured for neural network processing, involving discretization and diagonalization. SSMs are general representations of linear time-invariant (LTI) systems, and they can be uniquely specified by four matrices: $A \in \mathbb{R}^{h\times h}$, $B \in \mathbb{R}^{h\times n}$, $C \in \mathbb{R}^{m\times h}$, and $D \in \mathbb{R}^{m\times n}$. The first-order ODE describing the LTI system is given as

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du, \tag{1}$$

10.21437/Interspeech.2025-19

where $u \in \mathbb{R}^n$ is the input signal, $x \in \mathbb{R}^h$ is the internal state, and $y \in \mathbb{R}^m$ is the output. Here, we are letting $n > 1$, $m > 1$, which yields a multiple-input, multiple-output (MIMO) SSM. For the remainder of this paper, we will ignore the Du term as it effectively serves as a skip connection [28], which is already included explicitly in our network (see Fig. 1).

The SSM in its original form describes a continuous-time system, but in the field of digital signal processing, there are standard recipes for discretizing such a system into a discrete-time SSM. One such method that we use in this work is the zero-order hold (ZOH), which gives us the discrete-time state-space matrices $\overline{A}$ and $\overline{B}$ as follows:

$$\overline{A} = \exp(\Delta A), \qquad \overline{B} = (\Delta A)^{-1} \cdot (\exp(\Delta A) - 1) \cdot \Delta B. \quad (2)$$

The discrete SSM is then given by

$$x[t+1] = \overline{A}\, x[t] + \overline{B}\, u[t], \qquad y[t] = C\, x[t] \quad (3)$$

In other words, this is essentially a linear RNN layer, which allows for efficient online inference and generation (in our case real-time speech enhancement), but at the same time efficient parallelization during training.

It is straightforward to check that the discrete-time impulse response is given as

$$k[\tau] = C\, \overline{A}^{\tau}\, \overline{B} = CK(\tau)\overline{B}, \quad (4)$$

where τ denotes the kernel timestep and K represents the "basis kernels". During training, k can then be considered the "full" long 1D convolutional kernel with shape, in the sense that the output y can be computed via the long convolution $y_j = \sum_i u_i * k_{ij}$. By the convolution theorem, we can perform this operation in the frequency domain, which becomes a point-wise product $\hat{y}_{jf} = \sum_i \hat{u}_i \hat{k}_{ijf}$. The hat symbol denotes the Fourier transform of the signal (with the index f denoting the Fourier modes), which can be efficiently computed via Fast Fourier Transforms (FFTs).

It is a generic property that a diagonal form exists for the SSM, meaning that we can almost always assume $\overline{A}$ to be diagonal [28], at the expense of potentially requiring $\overline{B}$ and C to be complex matrices. Since the original system is a real system, the diagonal $\overline{A}$ matrix can only contain real elements and/or complex elements in conjugate pairs. In this work, we sacrifice a slight loss in expressivity by continuing to restrict $\overline{B}$ and C to be real matrices and letting $\overline{A}$ be a diagonal matrix with all complex elements (but not restricting them to come in conjugate pairs). Since we still want to work with real features[1], we then only take the real part of the impulse response kernel as such:

$$k[\tau] = \Re(C\overline{A}^{\tau}\overline{B}), \quad (5)$$

which equivalently in the state-space equation can be achieved by simply letting $y[t] = C\,\Re(x[t])$. This means that during online inference, we need to maintain the internal states x as complex values, but only need to propagate their real parts to the next layer.

2.1. Optimal Contractions

Similar to previous works in deep SSMs, we allow the parameters $\{A, B, C, \Delta\}$ to be directly learnable, which indirectly trains the kernel k. Unlike previous works in deep SSMs, we do not try to keep the sizes n, h, m homogeneous, to allow for more flexibility in feature extraction at each layer, mirroring the flexibility in selecting channel and kernel sizes in convolutional neural networks[2]. The flexibility of the tensor shapes requires us to carefully choose the optimal order of operations during training to minimize the computational load.

In short, in einsum notation, our SSM operations can be written as $\hat{y}_{jf} = \hat{u}_{if} B_{in} \hat{K}_{nf} C_{jn}$, a series of tensor contractions (matrix multiplications). Based on the tensor shapes involved, we can select an order of operation that is memory- and compute-optimal [24, 23]. In addition, this allows us to freely configure the network as an hourglass network (with the SSM layers in the network having different channel and temporal dimensions), without enforcing the SISO form [26, 30, 31]. This makes our network uniquely capable of handling raw audio waveforms, while retaining strong featural interactions and light computational costs. See supplementary material or our sister work Centaurus [24] for details on optimal tensor contractions.

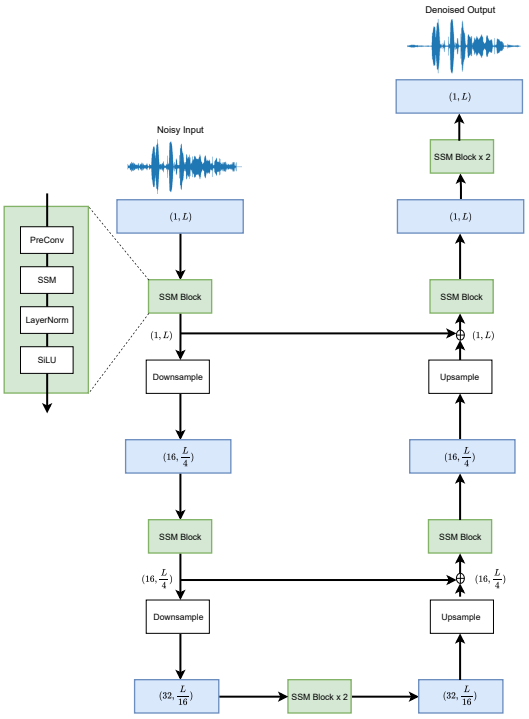

Figure 1: *A schematic drawing of the network architecture, with only 2 encoder and decoder blocks shown for simplicity. The actual model has 6 encoder and decoder blocks. Note that there is no (spectral) processing on the input and output waveforms.*

3. Network Architecture

We use an hourglass network with long-range skip connections, similar in form to the Sashimi network [26] for audio generation. However, unlike previous works using SSMs for audio processing [26, 30, 31, 32, 33], our network directly takes in

[1]This is *a prior* not required, as technically we can configure our network as complex-valued to handle complex features. However, we do not explore this configuration in this work.

[2]The size of the internal state h, can be interpreted as the degree of parametrization of a basis temporal kernel, or some implicit (dilated) "kernel size" in the frequency domain. We explore this in a future work.

raw audio waveforms in the -1 to +1 range and outputs raw waveforms as well, with no one-hot encoding or spectral processing (e.g. STFT or iSTFT). Furthermore, we retain causality as much as possible for the sake of real-time inference, meaning that we eschew any form of bidirectional state-space layers. See Fig. 1 for a schematic drawing.

As with typical autoencoder networks, the audio features are down-sampled in the encoder and then up-sampled in the decoder. For the re-sampling operation, we use a simple operation that squeezes/expands the temporal dimension then projects the channel dimension [26]. More formally, for a reshaping ratio of r, a sequence of features can be down-sampled and up-sampled as:

$$\begin{aligned} \text{Down: } (C_{in}, L) &\xrightarrow{\text{reshape}} (C_{in}r, L/r) \xrightarrow{\text{project}} (C_{out}, L/r) \\ \text{Up: } (C_{in}, L) &\xrightarrow{\text{reshape}} (C_{in}/r, Lr) \xrightarrow{\text{project}} (C_{out}, Lr) \end{aligned} \tag{6}$$

The baseline network uses LayerNorm layers and SiLU activations. In addition, we include a "PreConv" layer which is a depthwise 1D convolution layer with a kernel size of 3, to enable better processing of local temporal features. The PreConv operation is omitted in the 2 SSM blocks in the neck and in any block with only one channel[3]. Using causal convolutions for PreConvs can eliminate additional latencies to the network [22], but we do not explore this implementation here. To better support mobile devices, we also test variants of the network with BatchNorm layers[4], ReLU activations, and omitting PreConv layers in the decoder (and omitting PreConv layers altogether). The results are reported in the next section. The block-wise network architecture is given in Table 1, along with the PreConv latency, if present.

Table 1: *Resampling factor and output channel of each block of the network, which consists of an encoder performing down-samplings, an intermediate bottleneck, a decoder performing up-samplings, and finally an output processor.*

Layers	Resample Factor	Channels	Latency
Encoder			
Block 1	4	16	
Block 2	4	32	0.25ms
Block 3	2	64	1ms
Block 4	2	96	2ms
Block 5	2	128	4ms
Block 6	2	256	8ms
Neck			
Block 1	1	256	
Block 2	1	256	
Decoder			
Block 1	2	128	8ms
Block 2	2	96	4ms
Block 3	2	64	2ms
Block 4	2	32	1ms
Block 5	4	16	0.25ms
Block 6	4	1	
Output			
Block 1	1	1	
Block 2	1	1	

4. Experiments

For experiments, we train on the VCTK and LibroVox training sets downloaded from the Microsoft DNS Challenge, randomly mixed with noise samples from Audioset, Freesound, and DEMAND. We evaluate our denoising performance on the Voicebank + DEMAND (VB-DMD) testset, and the Microsoft DNS1 synthetic test set (with no reverberation) [34]. To guarantee no data leakage between the training and testing sets, we removed the clean and noise samples in the training set that were used to generate the synthetic testing samples. Both the input and output signals of the network are set at 16000 Hz. The loss function is a mix of SmoothL1Loss [35] and spectral loss at the ERB scale [19].

We train the model for 500 epochs, AdamW optimizer with a learning rate of 0.005 and a weight decay of 0.02, augmented with a cosine decay scheduler with a linear warmup of 0.01 of the total training steps. Each epoch contains the full VCTK training set, a random subset of the LibroVox training set (of ratio 0.1). To synthesize random noisy samples on the fly as inputs to the network, the clean samples are mixed randomly with the noise samples, with SNR values uniformly sampled from -5 dB to 15 dB.

The evaluation metric is the average wideband PESQ score between the clean signals and the denoised outputs. The PESQ scores for different variants of the aTENNuate model against other real-time audio-denoising networks are reported in Table. 2. We also report other network inference metrics including parameters, MACs, and latencies. For the latency metric, we focus on the theoretical latency of the network, not accounting for any processing latencies. In simpler terms, it is the maximum time range that the network needs to "wait" or look-forward to produce a denoised data point corresponding to the current input. As seen in Table 2, the PreConv layers (being depthwise) does not make much difference in terms of parameters and MACs, but does add considerably to latency. Unless otherwise denoted, we use the source code and pre-trained weights of each model and run it through a standardized PESQ evaluation pipeline. In addition, we report results for other common speech-enhancement metrics in Table 3.

To further inspect the quality of the denoised samples produced by the network, we listened to the denoised samples from both synthetic data and real recordings, and performed internal A/B tests against other networks. In the supplementary material, we also included the script for generating denoised audios for real recordings from the DNS1 challenge with natural reverberations. In addition, we provide a comparison of the denoised spectrogram and the clean spectrogram in Fig. 2, to ensure that the denoised samples do not contain any unnatural artifacts as common with raw audio processing systems[5], which may not be captured by the PESQ score [36]. In addition, we also tried to test a homogeneous variant of the network working with spectral features (STFT + iSTFT) with a similar number of parameters and MACs, but the PESQ results were significantly worse.

[3]The neck operates on fully-downsampled features, and introducing PreConv layers will incur too much latency in real-time processing. We also omit PreConv layer in single-channel blocks, because it is too lossy.

[4]Since BatchNorm is a static form of normalization during inference, the normalization statistics and the affine parameters can be "folded" into the weights and biases of the previous layer, meaning that the layer does not need to be materialized during inference.

[5]For example, the network may generate low-frequency artificial signals, resulting in a background "humming" noise. It may also attempt to aggressively attenuate high-frequency speech components, resulting in a robotic tone of speech.

Table 2: *Comparing different variants of the aTENNuate network against other real-time audio denoising networks, in terms of performance, memory/computational requirements, and latency.*

Model	PESQ (VB-DMD)	PESQ (DNS1 no-reverb)	Parameters	MACs / sec	Latency
FRCRN [15]	3.21	**3.23**	6.9M	38.1G	30ms
DeepFilterNet3 [19]	3.16	2.58	2.13M	0.344G	40ms
DEMUCS [20]	2.56	2.65	33.53M[a]	7.72G[a]	40ms
PercepNet [17]	2.73[b]	-	8.00M	0.80G	40ms
aTENNuate (base)	**3.27**	2.98	0.84M	0.33G	46.5ms
PreConvs only in encoder	3.21	2.84	0.84M	0.33G	31.25ms
no PreConvs	3.06	2.59	0.84M	0.33G	**16ms**
BatchNorm + ReLU	2.84	2.43	**0.84M**	**0.33G**	**16ms**

[a] These numbers are estimated by passing a one-second segment of data to the model.
[b] This metric is taken directly from the paper as an official implementation of the model does not exist.

Table 3: *Various speech enhancement metrics for the base aTENNuate network.*

Testset	PESQ	CSIG	CBAK	COVL	SISDR
VBDMD	3.27	4.57	2.85	3.96	15.04
DNS1	2.98	4.28	3.55	3.57	15.40

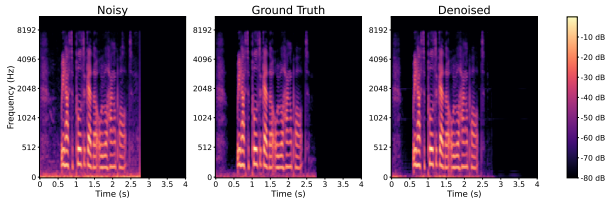

Figure 2: *A comparison of the spectrograms among the noisy signal, the clean ground truth, and the denoised output of a sample in the DNS1 synthetic testset (no reverb). Besides a minor low-frequency artifact in the silent region, the denoised output matches very close to the ground truth signal, despite not using any pre/post-processing in the spectral domain.*

In addition to audio denoising, we also perform studies on the ability of our network to perform super-resolution and de-quantization on highly compressed data. This involves intentionally performing down-sampling and quantization of the input signals, in that order, and re-training the network to handle the degraded/compressed inputs[6]. To use the same network architecture to interface with down-sampled audios, we perform interleaved repeats of the input signals to restore the original sample rate[7]. For quantization, we perform mu-law encoding on the input signals down to the desired bitwidth, then rescale the quantized signal back to the -1 to +1 range. The super-resolution and de-quantization results are reported in Table. 4. Note that the outputs are still evaluated against clean signals at 16000 Hz and full precision.

[6]It is possible to perform parameter-efficient fine-tuning of a baseline model instead, such as using low-rank adaptations on the state-space matrices. We leave this to future work.

[7]An alternative is to simply remove an encoder block with the same downsampling factor as the downsampled input.

Table 4: *The average PESQ scores of the model outputs when the noisy inputs are down-sampled and quantized.*

Input Type	VBDMD	DNS1
8000 Hz & 8 bit	3.19	2.88
4000 Hz & 8 bit	3.04	2.72
8000 Hz & 4 bit	2.90	2.55
4000 Hz & 4 bit	2.72	2.39

5. Future Directions

To make the network even more mobile-friendly, we plan to explore the sparsification and quantization of the network weights and activations. Potentially, we can study a low-rank realization of the state-space matrices, which form the majority of the parameter count. SSMs are also known to be easily adaptable for a spiking implementation on neuromorphic hardware, so it will also be interesting to see whether our model admits an efficient spiking neural network realization, which will further reduce the power required for the solution.

6. Conclusion

We introduced a lightweight deep state-space autoencoder, aTENNuate, that can perform raw audio denoising, super-resolution, and de-quantization. Compared to previous works, the key features of this network are: 1) consisting of SSM layers that can be efficiently trained and configured for inference, 2) allowing for real-time inference with low latency, 3) architecturally simple and light in parameters and MACs, 4) capable of processing raw audio waveforms directly without requiring pre/post-processing, and 5) highly competitive with other speech enhancement solutions.

7. Acknolwedgement

We thank Temi Mohandespour, Keith Johnson, and Nikunj Kotecha for contributing to the early stages of the project. We also thank M. Anthony Lewis, Douglas McLelland, Kristofor Carlson, and Chris Jones for providing useful feedback on the manuscript.

8. References

[1] F. G. Germain, Q. Chen, and V. Koltun, "Speech denoising with deep feature losses," *arXiv preprint arXiv:1806.10522*, 2018.

[2] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction wiener filter," *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1218–1234, 2006.

[3] S. V. Vaseghi, *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[4] V. Srinivasarao and U. Ghanekar, "Speech enhancement-an enhanced principal component analysis (epca) filter approach," *Computers & Electrical Engineering*, vol. 85, p. 106657, 2020.

[5] C. Yu, R. E. Zezario, S.-S. Wang, J. Sherman, Y.-Y. Hsieh, X. Lu, H.-M. Wang, and Y. Tsao, "Speech enhancement based on denoising autoencoder with multi-branched encoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2756–2769, 2020.

[6] A. Azarang and N. Kehtarnavaz, "A review of multi-objective deep learning speech denoising methods," *Speech Communication*, vol. 122, pp. 1–10, 2020.

[7] Z. Zhao, H. Liu, and T. Fingscheidt, "Convolutional neural networks to enhance coded speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 663–678, 2018.

[8] R. Giri, U. Isik, and A. Krishnaswamy, "Attention wave-u-net for speech enhancement," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 249–253.

[9] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," *arXiv preprint arXiv:2006.12847*, 2020.

[10] M. M. Kashyap, A. Tambwekar, K. Manohara, and S. Natarajan, "Speech denoising without clean training data: A noise2noise approach," *arXiv preprint arXiv:2104.03838*, 2021.

[11] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," in *International Conference on Learning Representations*, 2018.

[12] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[13] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5069–5073.

[14] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, "Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement," *arXiv preprint arXiv:2008.00264*, 2020.

[15] S. Zhao, B. Ma, K. N. Watcharasupat, and W.-S. Gan, "Frcrn: Boosting feature representation using frequency recurrence for monaural speech enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9281–9285.

[16] C. Zheng, H. Zhang, W. Liu, X. Luo, A. Li, X. Li, and B. C. Moore, "Sixty years of frequency-domain monaural speech enhancement: From traditional to deep learning methods," *Trends in Hearing*, vol. 27, p. 23312165231209913, 2023.

[17] 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.

[18] J.-M. Valin, "A hybrid dsp/deep learning approach to real-time full-band speech enhancement," in *2018 IEEE 20th international workshop on multimedia signal processing (MMSP)*. IEEE, 2018, pp. 1–5.

[19] H. Schröter, T. Rosenkranz, A. Maier *et al.*, "Deepfilternet: Perceptually motivated real-time speech enhancement," *arXiv preprint arXiv:2305.08227*, 2023.

[20] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," *arXiv preprint arXiv:2006.12847*, 2020.

[21] Z. Kong, W. Ping, A. Dantrey, and B. Catanzaro, "Speech denoising in the waveform domain with self-attention," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7867–7871.

[22] Y. R. Pei, S. Brüers, S. Crouzet, D. McLelland, and O. Coenen, "A Lightweight Spatiotemporal Network for Online Eye Tracking with Event Camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2024.

[23] Y. R. Pei and O. Coenen, "Building temporal kernels with orthogonal polynomials," *arXiv preprint arXiv:2405.12179*, 2024.

[24] Y. R. Pei, "Let ssms be convnets: State-space modeling with optimal tensor contractions," *arXiv preprint arXiv:2501.13230*, 2025.

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

[26] 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.

[27] Z. Wang, X. Zhu, Z. Zhang, Y. Lv, N. Jiang, G. Zhao, and L. Xie, "Selm: Speech enhancement using discrete tokens and language models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 561–11 565.

[28] A. Gu, K. Goel, A. Gupta, and C. Ré, "On the parameterization and initialization of diagonal state space models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 971–35 983, 2022.

[29] J. T. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," *arXiv preprint arXiv:2208.04933*, 2022.

[30] Y. Du, X. Liu, and Y. Chua, "Spiking structured state space model for monaural speech enhancement," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 766–770.

[31] P.-J. Ku, C.-H. H. Yang, S. M. Siniscalchi, and C.-H. Lee, "A multi-dimensional deep structured state space approach to speech enhancement using small-footprint models," *arXiv preprint arXiv:2306.00331*, 2023.

[32] R. Chao, W.-H. Cheng, M. La Quatra, S. M. Siniscalchi, C.-H. H. Yang, S.-W. Fu, and Y. Tsao, "An investigation of incorporating mamba for speech enhancement," *arXiv preprint arXiv:2405.06573*, 2024.

[33] X. Zhang, Q. Zhang, H. Liu, T. Xiao, X. Qian, B. Ahmed, E. Ambikairajah, H. Li, and J. Epps, "Mamba in speech: Towards an alternative to self-attention," *arXiv preprint arXiv:2405.12609*, 2024.

[34] C. K. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun *et al.*, "The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," *arXiv preprint arXiv:2005.13981*, 2020.

[35] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[36] D. de Oliveira, S. Welker, J. Richter, and T. Gerkmann, "The pesqetarian: On the relevance of goodhart's law for speech enhancement," *arXiv preprint arXiv:2406.03460*, 2024.