LEARNING INTERPRETABLE FILTERS IN WAV-UNET FOR SPEECH ENHANCEMENT

Félix Mathieu, Thomas Courtat, Gael Richard, Geoffroy Peeters

LEARNING INTERPRETABLE FILTERS IN WAV-UNET FOR SPEECH ENHANCEMENT

Félix Mathieu[⋆†], *Thomas Courtat*[†], *Gaël Richard*[⋆], *Geoffroy Peeters*[⋆]

[†] Thales SIX, advanced studies AI Lab, [⋆] LTCI, Télécom Paris, IP-Paris

ABSTRACT

Due to their performances, deep neural networks have emerged as a major method in nearly all modern audio processing applications. Deep neural networks can be used to estimate some parameters or hyperparameters of a model, or in some cases the entire model in an end-to-end fashion. Although deep learning can lead to state of the art performances, they also suffer from inherent weaknesses as they usually remain complex and non interpretable to a large extent. For instance, the internal filters used in each layers are chosen in an adhoc manner with only a loose relation with the nature of the processed signal. We propose in this paper an approach to learn interpretable filters within a specific neural architecture which allow to better understand the behaviour of the neural network and to reduce its complexity. We validate the approach on a task of speech enhancement and show that the gain in interpretability does not degrade the performance of the model.

***Index Terms*—** Representation learning, interpretability, speech enhancement

1. INTRODUCTION

Since nearly a decade, deep neural networks have permitted to obtain major improvements in performances for nearly all speech and audio applications, often defining the state-of-the art [1]. This is also visible in speech enhancement where a wide variety of neural architectures have been used with success [2, 3, 4]. However, if deep learning can lead to state of the art performances, they also suffer from inherent weaknesses as they usually remain complex and non-interpretable to a large extent. There is also a specific interest in speech enhancement for algorithms with a low computational cost. In fact, some of the main applications in this domain require real-time processing as illustrated in the Deep Noise Suppression (DNS) challenge [5] where the models have to run on constrained CPU devices.

Typical neural networks architectures exploit internal filters to process the information. However, these filters used in each layer are chosen in an ad hoc manner with only a loose relation with the nature of the processed signal.

In this paper, we propose an approach to learn interpretable filters within a specific neural architecture which allow to better understand the behavior of the neural network and to reduce its complexity. The neural architecture chosen for this study is the Wav-UNet [6]: yet very efficient in signal enhancement applications, its rather simple and flexible structure also facilitates its adaptation towards a more interpretable framework.

Our approach consists in replacing classic convolutions by separable convolutions which express the filtering process as two independent operations [7, 8]. This will allow us to highlight the specific behaviour of the internal filters for each layer and especially in terms of temporal and frequency resolution.

The paper is organised as follows: we discuss the background of our study in part 2 with a focus on speech enhancement and the chosen Wav-UNet architecture. Our proposed methods are described in part 3. We then detail our experiments and the results obtained in part 4 before suggesting some conclusion in part 5.

2. BACKGROUND

The objective of speech enhancement is to recover (enhance) the voice v from a mixture $x(t) = v(t) + n(t)$, where n denote a background noise. Speech enhancement has a long history and a very wide variety of approaches have been proposed in the last decades. Neural-based methods are discussed briefly below.

2.1. Speech enhancement methods

In recent years, speech enhancement has been dominated by neural-based algorithms, where the neural network are used for part or all of the enhancement task. Speech enhancement using deep neural network can be tackled in several ways (see for example [9, 10] for recent overviews).

Very interesting neural methods based on generative modelling (for instance using adversarial networks such as GAN [3, 11]) based on iterative diffusion process [4] have been successfully exploited for speech enhancement.

Another interesting family of efficient neural architectures for speech enhancement are based on auto-encoders including variational autoencoders and extensions (VAE [12], DVAE [13] UNet [14])

Many neural-based approaches recast the speech enhancement task as a mask estimation. In such methods, the aim is to estimate a mask (usually a time-frequency mask) that can be applied to the noisy signal x to obtain the enhanced speech signal [15].

Typically, we can use the Short Time Fourier Transform (STFT) and its inverse (ISTFT) as the first projection layer as in [2, 16, 17]. However, there is a growing interest for methods where the projection (or mask) is automatically learned as in TasNet [18] approaches.

Concurrently, a specific architecture called U-net [14], initially proposed for image segmentation has gained attention and several adaptations for audio source separation and speech enhancement have been proposed ([2, 6]). For instance, the Wav-UNet, which we will further study herein, has obtained state-of-the art results in music source separation at its introduction. It is a slightly different architecture compared to a typical mask approach since the multiplication step is here replaced by a concatenation between x and the learned neural representation followed by a last convolution. Hence, the input signal x is also used in the last layer of the network.

2.2. Wav-UNet: principle and architecture

We are particularly interested here in Wav-UNet type architectures, a UNet Auto-Encoder in the time domain, which processes the wave-

form directly. The benefit in studying this architecture lies in the flexibility of these components, since it is only a succession of 1D convolutions (in time). Moreover, this architecture has proven its efficiency when used alone or when coupled with a second branch as in [19] with a time/frequency representation. Another interesting aspect of this architecture is the possibility to adapt the temporal resolution (via the amount of downsampling used) but also the frequency resolution (via the size of the filters used in the convolutions). We briefly recall below the architecture of the Wav-UNet model.

2.2.1. *Overall Wav-UNet architecture*

Wav-UNet is an end-to-end autoencoder that operates directly on the raw audio waveform. At each layer of the encoder, the temporal resolution is reduced using a cascade of 1D convolutions followed by a downsampling operation (decimation by a factor of 2 in the original paper). On the decoder, the signal is reconstructed by also stacking a cascade of 1D convolutions. Here, the upsampling is obtained by a linear interpolation of the values. The input of each layer l; is the output of the previous decoding layer $l-1$ concatenated with the equivalent layer l of the encoder. This architecture is summarized in Fig. 1 where T represents the duration (in samples) of the input signal, $C_l \in \{1 \dots L\}$ the number of filters of layer l (which is the same in the encoder and decoder).

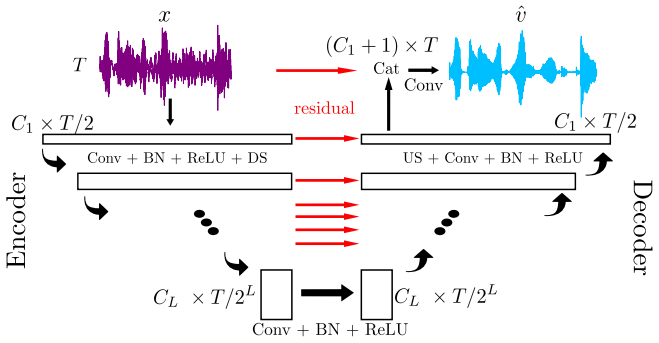

Fig. 1: Wav-UNet framework. Black arrows indicate convolution blocks. Red arrows indicate residual connections (concatenation in this model). $C_l \times T/2$ denotes the time and the features dimension of the signal.

Each convolution block in the encoder consists of a 1D convolution followed by a Down-Sampling (DS) operation, a Batch Normalization (BN) and ReLU:

$$\mathbf{E}_l = \text{DS}(\text{ReLU}(\text{BN}(\text{Conv}_l(\mathbf{E}_{l-1})))), \tag{1}$$

$\mathbf{E}_l \in \mathbf{R}^{C_l \times T/(2^l)}$ denotes the projection of the signal into the l^{th} layer. For the first layer, we identify the mixture x as $\mathbf{E}_0 \in \mathbb{R}^{1\times T}$.

The decoder is the exact opposite of the encoder. The upsampling (US) is performed using linear interpolation between successive values.

$$\mathbf{D}_{l-1} = \text{ReLU}(\text{BN}(\text{Conv}_l(\text{US}([\mathbf{D}_l : \mathbf{E}_l])))), \tag{2}$$

$[\mathbf{D}_l : \mathbf{E}_l] \in \mathbf{R}^{2C_l \times T/(2^l)}$ denotes the concatenation of the previous decoded embedding and the symmetrically encoded embedding.

The output of the decoder $\mathbf{D}_0$ is a feature map of the same temporal resolution as the input audio waveform. A final convolution layer is used on the concatenation of $\mathbf{D}_0$ and the noisy mixture $\mathbf{x}$ to reconstruct the clean signal $\hat{v}$.

2.2.2. *Modifying the time and frequency resolution*

A simple modification in the structure of this architecture allows to modify simultaneously the temporal and frequency resolutions within the network. Firstly, adjusting the depth of the network allows the signal to be considered on different time scales by applying a down-sampling operation and thus to obtain different temporal resolutions. Secondly, the size K of the filters defines the frequency resolution since the model process the signal directly in the time domain, therefore, increasing K can help to improve the frequency resolution.

Note though that using larger filter sizes significantly increases the computational cost. We will propose below a strategy which will allow to exploit larger filter size without increasing the computational cost and still maintaining the expressiveness of the model (i.e., without affecting the number of channels through the network).

3. LEARNING INTERPRETABLE FILTERS

The goal of this part is twofold:

- to introduce interpretable filters in the chosen neural architecture (which will allow a better visualization of the intermediate steps of the network),
- to reduce the computational complexity of the network.

To this aim, we propose to parameterize the convolutional filters such that the hidden representations (features at each layer) remain interpretable in terms of time-frequency content at various resolutions. For the second objective, we propose to factorize the parameterized filters and to exploit the concept of separable convolutions.

3.1. Depthwise Separable Convolution in Wav-UNet

Depthwise separable convolution [7] is a widely used reparametrisation trick for the convolution which allows reducing its overall computational cost [8]. It does so by factorizing the filters as a succession of two independent operations: for layer l

- a depthwise convolution Conv_l^d: which convolves independently each channel of the input $\mathbf{E}_{l-1}$ with a filter of size K (and depth 1); therefore using C_{l-1} independent filters.
- A pointwise convolution Conv_l^p: which corresponds to a linear layer between the channels from C_{l-1} to C_l.

The output of such a convolution is expressed as follows:

$$\text{Conv}_l^{DS}(\mathbf{E}_{l-1}) = \text{Conv}_l^p(\text{Conv}_l^d(\mathbf{E}_{l-1})), \tag{3}$$

The gain in complexity brought by this depthwise/pointwise convolution in the Wav-UNet (see part 4.3 for more details) allows using larger kernels K. Larger kernels allow the filters to have a higher frequency resolution and thus an easier interpretation of their characteristics. While depthwise separable convolution is less expressive than standard convolution, we show in our experimental part that the two perform roughly the same at equivalent FLOPS (i.e. imposing the same computation complexity).

In addition to the use of large filters, and also with the aim of increasing the interpretability of the filters in the frequency domain, we propose in the next part to replace the free-filters of Conv_l^d by parametric filters. As opposed to free-filters where the K values of the filters are learned independently, in parametric filters the K values are defined by a parametric function (for example a $sinc_\theta(\cdot)$ in the case of SincNet [20]) whose θ are the trainable parameters.

This allows us to obtain more sparse and less noisy filters in its frequency decomposition while reducing the number of parameters.

3.2. Gabor filters

Parametric filters (such as Sinc [20] or Gabor [21]) have already been proposed for the front-end of masking networks (for example for source separation in [22, 23]). Our contribution is to propose the use of those filters for all layers of the network (and not only the front-end). For our work, we consider the following parametric filter: the real part[1] of a Gabor function g defined as

$$g(n|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{n^2}{\sigma^2}} \cdot e^{j2\pi\mu n}, \quad (4)$$

where μ and σ are trainable parameters representing the normalized center frequency and the bandwidth of the filter.

4. EXPERIMENTS

In this part, we evaluate our proposal for a task of speech enhancement, analyse the complexity of the system and the properties of the learned filters.

4.1. Training setup

Dataset: We use the dataset from the DNS challenge [5]. The examples are created on the fly from the different voices, backgrounds and impulse responses using SNR in the range -5 to +20 dB. The training, validation and test sets are built such that speakers and backgrounds are different in each set.
Model characteristics: All models are 9-layer Wav-UNet. A layer l has $24 \times l$ channels. The decimation factor-stride is fixed to 2. The last layer of the encoder thus produces a hidden representation of size $216 \times T/2^9$, where T is the initial size of the input signal. All models use a kernel size K=15 for standard convolutions (as proposed by [6]) and K=64 for separable convolutions (in order to have a meaningful frequency representation in all layers).
Experimental protocol: In this study we compare five different architectures:

- *Baseline*: a Wav-UNet model that uses standard convolutions in the Encoder (E) and the Decoder (D),
- *ES*: same as Baseline but using depthwise-separable convolutions in E (Encoder-Separable),
- *FS*: same as Baseline but using depthwise-separable convolutions in E and D (Fully-Separable).

For ES and FS, we compare the use of free filters (ES-Free, FS-Free) and Gabor filters (ES-Gabor, FS-Gabor) for Conv_l^d.

We train the models using 3 s long mixture. Our models have a receptive fields of about 2 seconds at 16.000 Hz, so the choice of 3 s allows full use of the model's capabilities. We train the models by minimizing the Scale-Invariant Source-to-Noise Ratio [24](Si-SNR) using the Adam optimizer with a learning rate of 10^{-3}.

4.2. Speech enhancement results

In Table 1, we report the performances of the models using the SNR, STOI [26] and PESQ [27]. As a reference, the row *Noisy* indicates the average metric before enhancement. For comparison, we added the results obtained with a recent state-of-the-art speech enhancement model, the U-Net architecture CRUSE [25]. CRUSE gets better SiSNR than all our models, but get similar STOI and PESQ.

Table 1: Performance of the different models.

	SNR	STOI	PESQ
Noisy	7.33 ± 3.45	0.84 ± 0.12	2.15 ± 0.70
CRUSE [25]	15.24 ± 2.44	$0.87 \pm .10$	$2.88 \pm .57$
BaseLine	14.41 ± 2.42	0.88 ± 0.10	2.85 ± 0.57
ES-Free	14.36 ± 2.68	0.88 ± 0.10	2.86 ± 0.60
ES-Gabor	14.49 ± 2.54	0.88 ± 0.12	2.83 ± 0.60
FS-Free	14.36 ± 2.67	0.89 ± 0.10	2.88 ± 0.59
FS-Gabor	14.15 ± 2.48	0.89 ± 0.09	2.83 ± 0.57

We now look at our different Wav-UNet models (ES, FS, Free, Gabor). It can be seen that they all perform equally well on all metrics. Two things are worth noting:. First, the use of separable convolution (ES, FS) maintains equivalent performance than the Baselne while reducing the computational cost. Second, the parameterization with Gabor filters yields similar results than Free filters while further reducing the computational cost. In addition, the use of Gabor filters allows to obtain at the output of each layer band pass signals well localized in time and frequency. Finally, the use of Gabor filters in the encoder (ES) or in the whole network (FS) leads to similar performances.

4.3. Complexity analysis

In Table 2, we compare the number of trainable parameters and floating-point operations needed to process 1 s of signal (FLOPS). As seen, the use of ES or FS allows reducing the complexity (in terms of number of parameters and FLOPS) while keeping the same performances (see Table 1). Moreover, it should be noted that a large portion of the parameters and FLOPS are due to the pointwise convolution Conv_l^p. Therefore, one could further increase the size K of Conv_l^s while limiting the overall complexity.

Table 2: Number of parameters and FLOPS of the different models.

Model	# Parameters	FLOPS
BaseLine	3.8M	1.66G
ES-Free	2.70M	1.38G
ES-Gabor	1.43M	1.38G
FS-Free	1.1M	0.42G
FS-Gabor	0.76M	0.42G

4.4. Properties of the learned filters

4.4.1. Fully-Separable Wav-UNet visualisation

We now study the properties of the learned filters (with kernel size $K = 64$) for FS-Free. In Fig 2, we illustrate the amplitude of the Discrete Fourier Transform (DFT) of the 1D-convolution filters learned for layers $l = 1$ and $l = 4$ in the encoder. As it can be seen, the filters are well localized in the frequency domain.

This behavior extends to all layers in the encoder and decoder. The low temporal resolution seems to result in learning a wide range of low-pass filters. Fig. 3 and Fig. 4 further indicate the normalized central frequencies (defined as the $\nu_c = \arg\max_\nu$ of the DFT of the filter values) of all the filters for all layers in the same setting.

[1] Complex Gabor filters were considered at first (since they may improve the phase shift invariance of the network), but were finally not retained for this study, since we use here small stride values (one or two samples). In fact, in the time domain, the use of complex Gabor filters provides redundant information to the network.

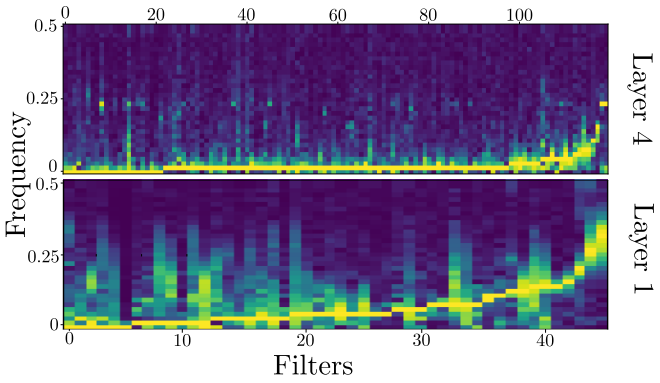

Fig. 2: Amplitude of the DFT of 1D-convolution filters used for layers $l = 1$ and $l = 4$ in the encoder part in a FS-Free.

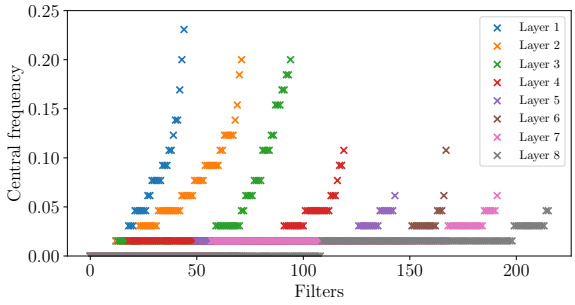

Fig. 3: Normalized central frequencies ν_c of learned filters for each layer of the encoder of FS-Free.

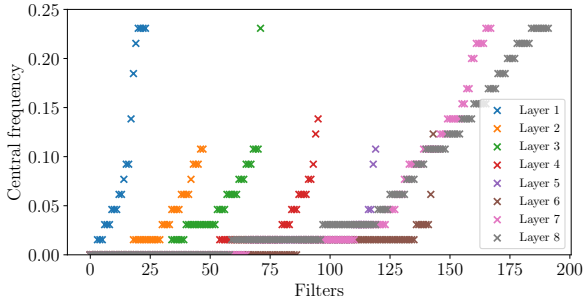

Fig. 4: Normalized central frequencies ν_c of learned filters for each lyer of decoder of FS-Free.

As we see, in almost all cases, the system has learned filters with a central frequency $\nu_c \leq 0.25$. This is a very interesting property because this low-pass behaviour indicates that it is possible to use a stride of 2 in the convolutions without losing information. Then, using a stride of 2 before or after the convolution is strictly equivalent for the encoder in this scenario. This behavior allows to halves the computational cost in the encoder without any change in the final prediction.

4.4.2. Gabor-Wav-UNet visualisation

The previous analyses suggest the use Gabor filters at all layers. Indeed, this parameterization allows us to build filters whose center frequency and bandwidth are controllable. Without any limitation, models learn normalized center frequencies ν_c below 0.25. So, we can limit this center frequency to be less than 0.25 to ensure equivalence with the use of a stride of 2 in the depthwise part of the convolution at each step of the learning process. In Fig. 5, we illustrate the values of ν_c for all Gabor filters (of size K=64) of all layers.

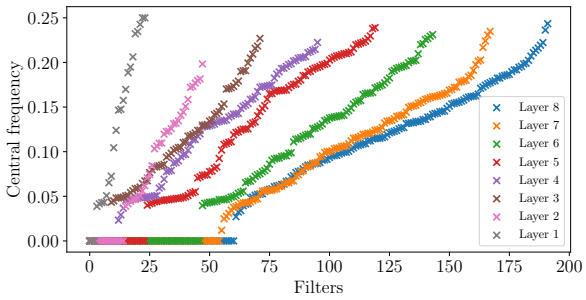

Fig. 5: Central frequencies ν_c of learned filters for each layer in the encoder part in a FS-Gabor.

Fig. 3 and 5 show the behavior for the first layers. The main difference between the Gabor and the free filters is in the center frequencies of the last layers. Although clearly noticeable, we do not have a solid interpretation of this difference for the last layers. We also note, for all layers, a discontinuity near the zero frequencies. To explain this, we investigate the bandwidths associated with these filters. As illustrated in Fig. 6 for l=8, the filters have large bandwidths, which explains that they can cover the lower "missing" frequencies.

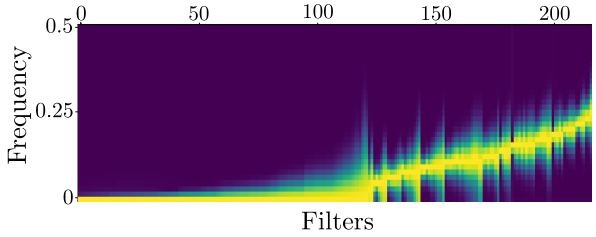

Fig. 6: Amplitude of the DFT of 1D-convolution filters used for layer $l = 8$ in the encoder part in a FS-Gabor.

5. CONCLUSION

In this paper, we have proposed a method for learning interpretable filters for all layers of a dedicated neural architecture (Wav-UNet). We have illustrated how these filters are interpretable and how we have exploited their properties to introduce a significant gain in complexity. We have further demonstrated that the gains in interpretability and complexity have no negative impact on the performance of the model on a task of speech enhancement. Future work will be dedicated to the extension to other audio applications (e.g. music source separation, acoustic scene and event detection) to assess the genericity of the filters obtained and to design, if needed, appropriate adaptation strategies.

6. REFERENCES

[1] Geoffroy Peeters and Gaël Richard, *Multi-faceted Deep Learning: Models and Data*, chapter Deep Learning for Audio and Music, Springer, 2021.

[2] Yx Hu, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie, "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," in *INTERSPEECH*, 2020.

[3] Santiago Pascual, Antonio Bonafonte, and Joan Serrà, "SEGAN: Speech enhancement generative adversarial network," in *INTERSPEECH*, 2017.

[4] Yen-Ju Lu, Zhongqiu Wang, Shinji Watanabe, Alexander Richard, Cheng Yu, and Yu Tsao, "Conditional diffusion probabilistic model for speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[5] Harishchandra Dubey, Vishak Gopal, Ross Cutler, Ashkan Aazami, Sergiy Matusevych, Sebastian Braun, Sefik Emre Eskimez, Manthan Thakker, Takuya Yoshioka, Hannes Gamper, and Robert Aichner, "ICASSP 2022 deep noise suppression challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[6] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *19th International Society for Music Information Retrieval Conference, ISMIR 2018*.

[7] Laurent Sifre, "Rigid-motion scattering for image classification," 2014, PhD Thesis, École Polytechnique.

[8] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017.

[9] DeLiang Wang and Jitong Chen, "Supervised speech separation based on deep learning: An overview," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[10] Emmanuel Vincent, Tuomas Virtanen, and Sharon Gannot, Eds., *Audio Source Separation and Speech Enhancement*, Wiley, 2018.

[11] Huy Phan, Ian V. McLoughlin, Lam Pham, Oliver Y. Chén, Philipp Koch, Maarten De Vos, and Alfred Mertins, "Improving GANs for speech enhancement," in *IEEE Signal Processing Letters*, 2020.

[12] Huajian Fang, Guillaume Carbajal, Stefan Wermter, and Timo Gerkmann, "Variational autoencoder for speech enhancement with a noise-aware encoder," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[13] Xiaoyu Bie, Simon Leglaive, Xavier Alameda-Pineda, and Laurent Girin, "Unsupervised speech enhancement using dynamical variational autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[15] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Deep learning for monaural speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[16] Umut Isik, Ritwik Giri, Neerad Phansalkar, Jean-Marc Valin, Karim Helwani, and Arvindh Krishnaswamy, "Poconet: Better speech enhancement with frequency-positional embeddings, semi-supervised conversational data, and biased loss," in *INTERSPEECH*, 2020.

[17] Xiang Hao, Xiangdong Su, Radu Horaud, and Xiaofei Li, "Fullsubnet: A full-band and sub-band fusion model for real-time single-channel speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[18] Yi Luo and Nima Mesgarani, "Tasnet: Time-domain audio separation network for real-time, single-channel speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[19] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis R. Bach, "Demucs: Deep extractor for music sources with extra unlabeled data remixed," *ArXiv*, vol. abs/1909.01174, 2019.

[20] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018.

[21] Paul-Gauthier Noé, Titouan Parcollet, and Mohamed Morchid, "CGCNN: Complex gabor convolutional neural network on raw speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[22] Félix Mathieu, Thomas Courtat, Gaël Richard, and Geoffroy Peeters, "Phase shifted bedrosian filterbank: An interpretable audio front-end for time-domain audio source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[23] Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent, "Filterbank design for end-to-end speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[24] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

[25] Sebastian Braun, Hannes Gamper, Chandan K. A. Reddy, and Ivan Tashev, "Towards efficient models for real-time deep noise suppression," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[26] Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[27] Antony W. Rix, John G. Beerends, M.P. Hollier, and Andries P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.