PHASE SHIFTED BEDROSIAN FILTERBANK: AN INTERPRETABLE AUDIO FRONT-END FOR TIME-DOMAIN AUDIO SOURCE SEPARATION

Félix Mathieu, Thomas Courtat, Gael Richard, Geoffroy Peeters

PHASE SHIFTED BEDROSIAN FILTERBANK: AN INTERPRETABLE AUDIO FRONT-END FOR TIME-DOMAIN AUDIO SOURCE SEPARATION

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

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

ABSTRACT

The use of a parameterized encoders or audio front-ends has shown promises in improving the interpretability of time domain single-channel source separation models such as Conv-TasNet. This type of filters also allows a potential reduction of the computational cost since larger encoder filters can be used. In this work, we propose to build a new parameterization of such encoder filter-bank which allows gaining interpretability while keeping flexibility. Based on the Hilbert transform and the Bedrosian theorem, we propose to build phase-shifted set of filters by modulating sinusoids through freely learned low pass filters. We show that the use of these filters allows to keep the same performances when using small filters and even improve them when using large filters.

Index Terms— Audio source separation, audio filterbank

1. INTRODUCTION

Over the past years, major improvements in Single-Channel Source Separation (SCSS) has been achieved thanks to Deep Neural Networks (DNN). Among the possible DNN approaches, masking methods have been shown to be the most efficient, either in the spectral domain using the Short Time Fourier Transform (STFT) [1, 2] or in the time domain directly on the audio waveform using learned filters through the training of 1D-convolutions [3, 4].

Given a mixture of C sources, $x(t) = \sum_{i=1}^{C} s_i(t)$, a typical masking network is then made of an encoder $\mathbf{w} = f_e(\mathbf{x})$ applied to the input audio waveform $\mathbf{x}$, a separator $\mathbf{m}_i = f_s(\mathbf{w})$ applied to the output of the encoder — it builds masks $\mathbf{m}_i$ for each source $\mathbf{s}_i$ to be separated — and a decoder $\hat{\mathbf{s}}_i = f_d(\mathbf{w} \odot \mathbf{m}_i)$ applied to the masked (element-wise) output of the encoder.

While much effort has been put in the improvement of the separator f_s [5, 6, 7], the encoder f_e or audio front-ends used in the time domain has been little discussed. However, simple structural modifications of it can significantly change the performances without changing the overall architecture.

$f_e(\mathbf{x})$ is usually a 1D-convolution with free-filters (has in TasNet [3] or Conv-TasNet [3]). It is however possible to constrain the filters by parameterizing their shape such as in SincNet [8] for Automatic Speech Recognition (ASR) which allows to extend the filter banks on the one hand for ASR [9, 10], but also for source separation tasks. Recently, Pariente et al. [11] have shown that a simple trick in the estimation of filters can improve the performance on speech separation in noisy conditions. On the other hand, Ditter et al. [12, 13] have shown that well-chosen Gammatone filters allows to avoid the learning of a front-end without loss of performances.

1.1. Proposal and paper organization

In this paper, we propose a new front-end which achieves high performances for SCSS while using large filter size. The use of large filter size allows to decrease the overall computation (since the number of frames (per second) to process is reduced). For this, we propose to extend the Hilbert transform and apply the Bedrosion theorem to separate and control the amplitude and phase term of the filters. The front-end is parametrized in a regime close to the STFT. A side benefit of our approach is that it enforces interpretability.

The paper is organized as follows. We first review the Time-domain Audio Separation (TAS) architecture of masked network (1.2), review the encoders f_e (audio front-ends) which have been previously proposed (1.3) and discuss the interpretability of the trained filters and their pro and cons. In part 2, we then investigate the construction of new interpretable encoder parameterizations. We propose extended-Hilbert-transform filters (2.1) and Bedrosian filters (modulated sinusoid filterbanks) (2.2). Finally, in part 3, we compare the performances of our two proposals with previous proposals for a task of SCSS. We constrain the systems to have the same complexity (same Floating-Point Operations Per Second (FLOPS)).

1.2. Time-domain Audio Separation Architecture

TAS networks [3, 4] are based on the masking architecture and operates directly on the audio waveform $x(t)$. They have allowed a real breakthrough in SCSS and hence became a central object of study because of its simple structure and flexibility. In those, f_e is a simple 1D-convolution which projects $x(t)$ on a filterbank. It uses N filters of size L. f_s is a Recurrent Neural Network (RNN) in TasNet [4] and a Temporal Convolutional Networks (TCN) in Conv-TasNet [3]. The decoder f_d is also a simple 1D convolution.

1.3. Existing encoders / audio front-ends f_e

1.3.1. STFT front-end

The STFT can be considered as a non-trained f_e: each $\cos, \sin$ basis is then considered as a 1D-filter. In this case, the filters are ordered (by increasing frequency). This ordering allows to apply 2D-convolution directly to the output of the encoder. When f_e is trained, the resulting 1D-filters are not ordered, then it will not be possible to apply 2D-convolution.

1.3.2. Free front-end

In [3, 4], f_e is a simple trained 1D-convolution. In the following we denote by L the size of these filters. In those papers, a small value of L has been chosen (16 samples for the initial article) which seems to be the most efficient for their system. In the Dual Path Transformer network [14], a value even lower (L=4) has been proposed with even higher performances.

However, using such a small filter in f_e leads to a higher cost in the separator f_s. Indeed using smaller L implies more convolutions hence more computations in f_s. Increasing L is therefore an interesting direction to reduce the number of operations. This is the direction we follow here.

Filter analysis. Luo et al. in their Conv-TasNet paper [3] have already noted that f_e seemed to learn waveforms with well-defined frequencies and a large number of phase-shifts for the same waveform. We represent this in Fig. 1 for L=256. We see that many filters have the same frequency pattern but different phase-shifts. In low-frequencies, the filters are very well localized in frequency while in mid-frequencies they are well localized in time. Note that the filters are naturally distributed according to logarithmic frequencies regardless of the size of the filters. However, as already studied in other works [15, 16], the learning process of such large filters presents more and more noise as we increase L. This phenomenon appears clearly in Figure 1 for high-frequencies: the learned filters look like random high pass filters.

1.3.3. Analytic front-end

From the above, we see that many "Free" filters represent similar waveform which only differ by their phase. This leads Pariente et al. [11] to only learn half of the N filters, we denote those by *base-filters* $s_0(t)$, and to deduce the other half by taking their Hilbert transform. The orthogonality of a filter and its Hilbert transform allows to project the signal on a precise waveform whatever its phase. The network can then robustly learn the phase shift of the desired waveform. The whole system remains differentiable then trainable. For any base-filter $s_0(t)$, its Hilbert transform in Fourier domain is:

$$\mathcal{H}(S_0(f)) = \begin{cases} S_0(f).e^{j\pi/2} & \text{if} f > 0, \\ S_0(f).e^{-j\pi/2} & \text{if} f < 0, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Combining $s_0(t)$ with its Hilbert transform $\mathcal{H}(s_0(t))$ as imaginary part leads to the analytical signal $\hat{s}(t)$: $\hat{s}(t) = s_0(t) + j\mathcal{H}(s_0)(t)$. $\hat{s}(t)$ is often used to extract the instantaneous amplitude and phase of $s_0(t)$: $\hat{s}(t) = A(t)e^{j\phi(t)}$, where A corresponds to a strictly positive signal (the *modulating* signal) and ϕ to any real signal (the phase of the *modulated* signal).

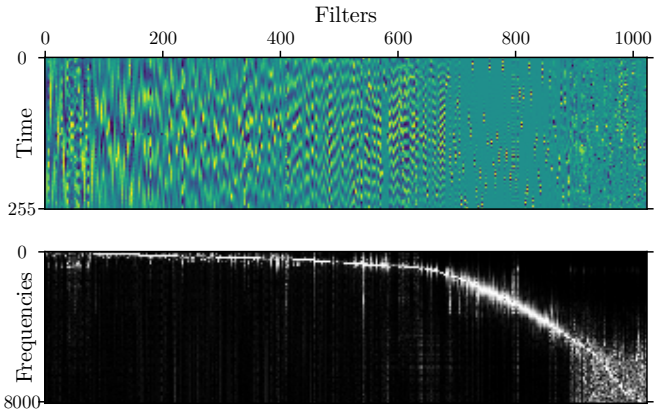

Fig. 1: Example of learned filterbank for Free front-end with L=256. The top and bottom parts represent respectively the learned filters in the time and in the frequency domain.

1.3.4. Gammatone/ over-parametrized front-end

Along the same spirit, Ditter et al. [12] propose to project $x(t)$ on a fixed set of Gammatone filters, each with a predefined frequency f_0 and for each f_0 a predefined set of phase-shifts ψ_i, hence the name Multi-Phase Gammatone Filterbank (MPGTF). A Gammatone filter is expressed as

$$\gamma(t|a, b, n, f_0, \psi_i) = a\ t^{n-1}e^{-2\pi bt} \cdot \cos(2\pi f_0 t + \psi_i) \quad (2)$$

With this, they reached state of the art results with small L. However, one major limitation of their model is that the Gammatone envelopes are always centered in the same area (whatever the values of ψ_i which only concerns the modulated signal). This becomes detrimental as L increases since it therefore provides little or no information on the temporal location. Note that, as for analytic signal, eq. (2) can be considered as a modulating (amplitude A) and modulated (phase ϕ) signal:

$$\gamma(t|f_0, \psi_i) = A(t) \cdot \cos(\phi(t)) = A(t) \cdot \cos(2\pi f_0 t + \psi_i), \quad (3)$$

where ψ_i corresponds to the chosen phase and f_0 to the oscillation frequency of the modulated signal.

While it would be enough theoretically to use only two filters (projection on 0 and $\pi/2$) for a given frequency in order to reconstruct the signal, it is precisely this over-parameterization of the ψ that allows this filter-bank to be so efficient: the phase of the signal is directly encoded in a given filter.

2. PROPOSAL

In order to add interpretability and be able to use large filter sizes L (hence decreasing the computational cost), we pro-

pose two new front-ends which take the benefits of both the analytical filters (1.3.3) and the MPGTF filters (1.3.4).

2.1. Extended Hilbert front-end

We first propose to extend the Hilbert transform to other phase values than $\pi/2$. For this, we simply shift the phase of a base-filter $s_0(t)$ to the desired values ψ:

$$\mathcal{H}_\psi(S_0(f)) = \begin{cases} S_0(f).e^{j\psi} & \text{if} f > 0, \\ S_0(f).e^{-j\psi} & \text{if} f < 0, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Doing so, we therefore re-use (a) the idea of the Hilbert transform (phase shift of a free base-filter $s_0(t)$, eq. (1)) and (b) the idea of the MPGTF (set of phase-shifts ψ_i of eq. (2) applied to the modulated part of Gammatone filters).

For a base-filter $s_0(t)$, we deduce a set of K filters $s_k(t)$ by taking $\psi \in \{k\pi/K\}_{k\in\{0,\dots,K-1\}}$. This corresponds to sampling the upper unit-circle of the complex plane[1]. As Pariente et al. [11] which only learned $N/2$ base-filters, we also reduce here the number of learned base-filters to N/K when using K phase shifts in eq. (4). Again, we can write $s_k(t)$ as a modulating and modulated signal:

$$s_k(t) = A(t) \cdot \cos\left(\phi(t) + \frac{k\pi}{K}\right), k \in \{0, \dots, K-1\}, \quad (5)$$

However, while this front-end is promising (as we will discuss in part 3 [2]), it does not allow to correct the noise which is inserted in $A(t)$. We therefore need to constraint $s(t)$ to avoid the presence of noise. We do this in the Bedrosian front-end by making the modulating signal $A(t)$ a low-pass signal and the modulated one a high-pass one.

2.2. Bedrosian front-end

If $A(t)$ and $\cos(\phi(t))$ are taken randomly, the resulting base-filter $s_0(t)$ is generally not analytic and the filters induced by the phase shifts will lose their interpretation in terms of rotation with respect to each other.

To deal with this, we use results from the Bedrosian theorem [17] which provides conditions to guarantee analyticity. [17] proves that for two complex valued functions f, g in $L^2(\mathcal{R})$, if the support of their Fourier transform are disjoint[3] then the Hilbert transform of their product can be written:

$$\mathcal{H}(fg) = f\ \mathcal{H}(g). \quad (6)$$

By associating f to $A(t)$ and g to $\cos(\phi(t))$ we have a sufficient condition to build analytical filters as before (see eq. (4)). Note that this condition can also be seen as the product of a low pass filter $A(t)$ and a high pass filter $\cos(\phi(t))$.

We now propose a practical implementation of this. Note that this should be considered as a first simple (but efficient) implementation which can be further improved.

Practical implementation. We choose to construct the modulated term $\cos(\phi(t))$ using a simple sinusoid parameterized by its frequency f_0 as in the SincNet architecture [8]. Then the amplitude parameter $A(t)$ is obtained by convolving a learned free filter $a(t)$ by a Gaussian lowpass filter[4]. The whole filters are therefore rewritten:

$$s_k(t|A, f_0) = A(t) \cos\left(2\pi f_0 t + \frac{k\pi}{K}\right), \quad (7)$$

$$\text{such as } A(t) = a(t) \circledast e^{-\left(\frac{t}{\sigma_t}\right)^2}, \quad (8)$$

$$\text{or } \mathcal{F}(A)(f) = \mathcal{F}(a)(f) * e^{-\left(\frac{f}{\sigma_f}\right)^2}, \quad (9)$$

and σ being $\frac{2L}{\pi f_0}$. The positivity constraint of A is obtained by shifting A such that $\min_t(A(t)) = 0$. The parameters to be trained are the f_0 and the free filters $a(t)$.

3. EXPERIMENTS

In the following, we compare the various front-ends (pre-existing and proposed) and their configurations for a task of separating speech from speech.

Dataset. We use the LibriMix [18] dataset. made of mixed items from LibriSpeech [19]. We use *dynamic mixing* to build our examples from the *train-clean-360* training subset with a ratios between 0 and 5 dB between the two speakers.

Experimental protocol. As Conv-TasNet, our separator f_s is a TCN. The different experiments will vary in the choice of the encoder f_e, their parameters and the hyper-parameters of the TCN. The hyper-parameters are chosen in order to have a constant computational cost (number of FLOPS) across experiments. We provide those in Table 1. For computational resource reason and in order to show that satisfactory results can be obtained with low computational costs, we perform our experiments without using an optimal hyperparametrization. Training is performed using 3 s long mixture. Losses and performances measures use the permutation invariant [20] Scale-Invariant Source-to-Noise Ratio [3] (SiSNR). For the implementation of all our networks, we used the Asteroid library [21].

3.1. Effect of the number of phase-shifts K

To understand the impact of base-filter redundancy, we first test the influence of K (the number of phase-shifts) for the extended Hilbert filterbank . We kept the total number of filters N fixed (N=1050) and choose $K \in \{1, 2, 3, 5, 7, 10\}$.

[1] It is in fact not useful to completely decompose the unit-circle since no activation function (such as ReLU) is used at the output of f_e and only a minus sign would appear between a filter and its π phase shift.

[2] Promising since the performances for SCSS remained stable while the number of parameters to be learned is reduced by K

[3] By disjoint we mean: given a real number a, f and g are respectively included in $]-a, a[$ and $]-\infty, -a[\cup]a, \infty[$

[4] The filters is designed to have an attenuation of -20 dB at frequency f_0. This choice is made in order to have smooth modifications of $A(t)$ when adding new frequencies in the free filter $a(t)$ if the value of f_0 increases.

Table 1: N and L refers respectively to the number of filters in the f_e and their size. H, B and S are the hyperparameter of the TCN separator (see [3] for more details on those).

Encoder type	N	L	H	B	S
Small $Sm.$	128	32	128	128	128
Mid $Mi.$	512	128	170	170	170
Large $Lg.$	1024	256	256	256	256
Very Large $Vl.$	2048	1024	512	512	512

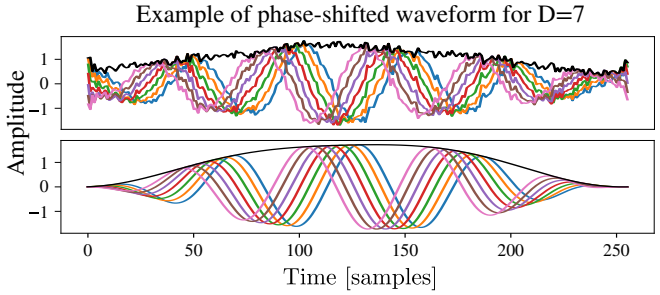

Fig. 2: [Top] Example of a learned base-filter $s_0(t)$ and its associated extended-Hilbert-transforms. We represent in black its temporal envelope. [Bottom] Same with Bedrosian.

The number of base-filter to be learnt varies according to $N_w = N/K$. Note that $K = 1$ corresponds to the free front-end and $K = 2$ to the analytic one (Hilbert-based). We use filters of sizes $L = 256$ (16 ms for a sampling rate of 16 kHz).

Table 2: Performance for different phase-shifts K.

N_w	1050	525	340	210	**150**	105
K	1	2	3	5	**7**	10
SiSNR	10.11	10.66	10.61	10.64	**10.73**	10.45

As can be seen in Table 2, for a fixed N value (1050), it is more important to have less base-filters but with good phase-accuracy (N_w=150, K=7, SiSNR=10.73) than having a large number of base-filters but decorrelated in phase (N_w=1050, K=1, SiSNR=10.11). However, increasing further the number of phases (K=10) leads to a too small number of base-filters (N_w=105) which is not diverse enough to correctly project the signal (SiSNR= 10.45). In Figure 2 [Top], we illustrate the filters for the case K=7. As can be seen, we gain in interpretability: the learned filters are highly correlated and seem to converge towards modulated sinusoids. This further motivates the choice we made for the Bedrosian filters: a set of modulated sinusoids. We illustrate the corresponding Bedrosian filters in Figure 2 [Bottom].

3.2. Comparison between different front-ends

We now compare the various pre-existing front-ends (Free, Gammatone, Analytic) with our proposed Extended Hilbert and Bedrosian filters. We also add to the comparison a fixed random front-end. We test two assumptions.

(1) Our first assumption is that when L increases the performance of the non-structured filters (Random fix, Free) will drop while the ones of structured filters (Analytic, Extended Hilbert and Bedrosian model) will not. We compare the results for L=32 (Sm.), 128 (Mi.) and 256 (Lg.). Table 3 confirms our assumption, it even shows a slight SiSNR increase for structured filters (Analytic, Extended Hilbert and Bedrosian model). As discussed, Gammatone results decrease with L since they do not allow to model the temporal location. For large L, the proposed Bedrosian filters achieved the best SiSNR (10.78). These results show that it is possible to gain in interpretability while maintaining (or even increasing) performances when the filters are larger. To check if we can further extend L, we compare the results obtained with a Very Large ($Vl.$) L=1024. In this case the performances of all front-ends drop (especially those of the Free front-end).

(2) Our second assumption is that structured filters will necessitate less training data. We test this in Table 4 using 0.1%, 1%, 10% or 100% of the data. Our assumption is however not confirmed: the performances of all front-ends drop.

Table 3: Performance (SiSNR) of the different encoder f_e.

Encoder type / L	$Sm.$	$Mi.$	$Lg.$	$Vl.$
Random fix	9.85	9.53	9.08	/
Free	10.40	10.33	10.11	7.76
Gammatone	10.43	8.34	6.52	/
Analytic	10.31	10.41	10.66	8.61
Extended Hilbert	10.37	**10.51**	10.73	8.58
Bedrosian	**10.50**	10.43	**10.78**	8.47

Table 4: Performance (SiSNR) of the different encoder f_e in function of the amount of training data for Lg.

Encoder type / Data	0.1%	1%	10%	100%
Free	6.16	8.66	9.61	10.11
Analytic	6.15	8.72	10.12	10.66
Extended Hilbert	6.21	8.88	10.05	10.73
Bedrosian	6.14	8.61	9.97	10.78

4. CONCLUSION

In this paper, we have studied existing encoders or audio front-ends for SCSS and proposed two new ones: either based on the extended Hilbert transform or on the Bedrosian theorem. This last one allows to gain interpretability of the filters while keeping the performances very high and even slightly improve them in the case of large filters. This parameterization of the front-end allows to bypass the problem of the noise which is usually inserted in the learned filters; it also allows keeping a greater flexibility than the SincNet architecture [8] and could potentially be used for other tasks such as ASR. Further works will concentrate on improving our specific implementation of the Bedrosian filters.

5. REFERENCES

[1] John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. 2016, pp. 31–35, IEEE.

[2] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*. 2017, pp. 246–250, IEEE.

[3] Yi Luo and Nima 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.

[4] Yi Luo and Nima Mesgarani, "Tasnet: Time-domain audio separation network for real-time, single-channel speech separation," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 696–700, 2018.

[5] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. 2020, pp. 46–50, IEEE.

[6] Efthymios Tzinis, Zhepei Wang, and Paris Smaragdis, "Sudo RM -rf: Efficient networks for universal audio source separation," in *30th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2020, Espoo, Finland, September 21-24, 2020*. 2020, pp. 1–6, IEEE.

[7] Liwen Zhang, Ziqiang Shi, Jiqing Han, Anyan Shi, and Ding Ma, "Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," in *MultiMedia Modeling - 26th International Conference, MMM 2020, Daejeon, South Korea, January 5-8, 2020, Proceedings, Part I*. 2020, vol. 11961 of *Lecture Notes in Computer Science*, pp. 653–665, Springer.

[8] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*. 2018, pp. 1021–1028, IEEE.

[9] Paul-Gauthier Noé, Titouan Parcollet, and Mohamed Morchid, "CGCNN: complex gabor convolutional neural network on raw speech," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. 2020, pp. 7724–7728, IEEE.

[10] Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi, "LEAF: A learnable frontend for audio classification," *CoRR*, vol. abs/2101.08596, 2021.

[11] Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent, "Filterbank design for end-to-end speech separation," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. 2020, pp. 6364–6368, IEEE.

[12] David Ditter and Timo Gerkmann, "A multi-phase gammatone filterbank for speech separation via tasnet," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. 2020, pp. 36–40, IEEE.

[13] Wenbo Zhu, Mou Wang, Xiao-Lei Zhang, and Susanto Rahardja, "A comparison of handcrafted, parameterized, and learnable features for speech separation," *CoRR*, vol. abs/2011.14295, 2020.

[14] Jingjing Chen, Qirong Mao, and Dong Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, Helen Meng, Bo Xu, and Thomas Fang Zheng, Eds. 2020, pp. 2642–2646, ISCA.

[15] Neil Zeghidour, Nicolas Usunier, Iasonas Kokkinos, Thomas Schatz, Gabriel Synnaeve, and Emmanuel Dupoux, "Learning filterbanks from raw speech for phone recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. 2018, pp. 5509–5513, IEEE.

[16] Neil Zeghidour, Qiantong Xu, Vitaliy Liptchinsky, Nicolas Usunier, Gabriel Synnaeve, and Ronan Collobert, "Fully convolutional speech recognition," *CoRR*, vol. abs/1812.06864, 2018.

[17] Yuesheng Xu and Dunyan Yan, "The bedrosian identity for the hilbert transform of product functions," *Proceedings of the American Mathematical Society*, vol. 134, pp. 2719–2728, 09 2006.

[18] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent, "Librimix: An open-source dataset for generalizable speech separation," 2020.

[19] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. 2015, pp. 5206–5210, IEEE.

[20] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*. 2017, pp. 241–245, IEEE.

[21] Manuel Pariente, Samuele Cornell, Joris Cosentino, Sunit Sivasankaran, Efthymios Tzinis, Jens Heitkaemper, Michel Olvera, Fabian-Robert Stöter, Mathieu Hu, Juan M. Martín-Doñas, David Ditter, Ariel Frank, Antoine Deleforge, and Emmanuel Vincent, "Asteroid: The pytorch-based audio source separation toolkit for researchers," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, Helen Meng, Bo Xu, and Thomas Fang Zheng, Eds. 2020, pp. 2637–2641, ISCA.